ARTICLE IN PRESS

International Journal of Forecasting xxx (xxxx) xxx

Contents lists available at ScienceDirect

International Journal of Forecasting

journal homepage: www.elsevier.com/locate/ijforecast

# Daily retail demand forecasting using machine learning with emphasis on calendric special days

Jakob Huber *, Heiner Stuckenschmidt

*Data and Web Science Group, University of Mannheim, B6 26, 68159 Mannheim, Germany*

ARTICLE INFO

*Keywords:*
Demand forecasting
Comparative studies
Forecasting practice
Neural networks
Decision trees
Regression
Classification

ABSTRACT

Demand forecasting is an important task for retailers as it is required for various operational decisions. One key challenge is to forecast demand on special days that are subject to vastly different demand patterns than on regular days. We present the case of a bakery chain with an emphasis on special calendar days, for which we address the problem of forecasting the daily demand for different product categories at the store level. Such forecasts are an input for production and ordering decisions. We treat the forecasting problem as a supervised machine learning task and provide an evaluation of different methods, including artificial neural networks and gradient-boosted decision trees. In particular, we outline and discuss the possibility of formulating a classification instead of a regression problem. An empirical comparison with established approaches reveals the superiority of machine learning methods, while classification-based approaches outperform regression-based approaches. We also found that machine learning methods not only provide more accurate forecasts but are also more suitable for applications in a large-scale demand forecasting scenario that often occurs in the retail industry.



## 1. Introduction

Demand forecasting is one of the major challenges for retailers as it is the input for many operational decisions (Van Donselaar, Gaur, Van Woensel, Broekmeulen, & Fransoo, 2010). In particular, for perishable goods with a high rate of deterioration, it is important to provide the correct quantities every day (Van Donselaar, van Woensel, Broekmeulen, & Fransoo, 2006). Such goods have higher average sales and higher order frequencies than non-perishable goods. Their freshness decreases rapidly, which makes daily replenishment inevitable. Unsold items are waste, and discarded goods are a major cost factor that can be reduced by more accurate demand forecasts as these enable the reduction of safety stocks (Ehrenthal & Stölzle, 2013). Retailers typically run a large number of stores and offer a broad assortment of goods. Consequently, numerous daily decisions need to be supported by predictions. A competitive advantage can be gained by automating the prediction process. Moreover, retailers accumulate very large datasets (e.g., sales history) over years that can also be enhanced by external information such as calendar events (Hofmann & Rutschmann, 2018).

Our research is motivated by the requirements of a real-world problem for a large German bakery chain that shares many characteristics with other mid-size bakeries. The company runs over 100 stores in a geographically restricted area. Pastries are produced in a centralized production facility and delivered daily to the stores. Daily delivery is necessary as the freshness of baked goods decreases rapidly, which only allows them to be sold on the day of production. The supply chain is quite agile as all of its important parts are operated by the company; i.e., production and distribution, as well as the stores. Daily demand forecasts are required for many operational decisions, including the determination of

* Corresponding author.
*E-mail addresses:* jakob@informatik.uni-mannheim.de (J. Huber), heiner@informatik.uni-mannheim.de (H. Stuckenschmidt).

https://doi.org/10.1016/j.ijforecast.2020.02.005
0169-2070/

production and delivery quantities, and staffing decisions (Huber, Gossmann, & Stuckenschmidt, 2017; Van Woensel, Van Donselaar, Broekmeulen, & Fransoo, 2007).

Demand in the bakery domain is subject to a strong weekly seasonal pattern. However, this is not entirely true for special days (SDs), which are subject to vastly different demands, making forecasting for such days a key challenge. Our definition of special days includes public holidays, days either side of public holidays, and other calendar events (e.g., Carnival). The daily demand on such days differs from regular days as customers change their daily routines. For instance, public holidays often fall on working days, but most people's typical working schedules (i.e., customers) and the assortment offered in the stores are largely comparable with those on Sundays. Moreover, days close to public holidays are also affected by the holiday as they may fall between the public holiday and the weekend, in which case people typically take an extra day off. Neighboring days of public holidays can also be used to supplement the demand on public holidays if the store is closed. Hence, depending on the location of the store, the demand on special days can be lower or higher compared to that on regular days. An additional challenge is that some public holidays are not always on the same weekday, which makes it difficult to quantify the actual effect of the public holiday.

The application scenario of the bakery chain has several characteristics that make it suitable for exploring the competitiveness of machine learning methods and concepts. Machine learning methods are data-driven pattern recognition algorithms that do not impose strict assumptions on the data generating process (Hastie, Tibshirani, & Friedman, 2009). They are able to learn patterns from data by exploiting large datasets, which makes them suitable for big data applications such as daily demand forecasting for a retailer. First, the stores belong to the same company, which means that they share many characteristics, including branding, pricing, and assortment. Even though the stores are not exactly the same, they belong to common classes depending on their location and facility equipment. Second, the stores are located in a geographically restricted area, which means that the customers as well as the market environments (e.g., comparable competitors) of the stores share similarities, and that external influences (e.g., the weather) are fairly comparable. This can be helpful for special days, for which a small number of observations are available per time series. As a consequence of these assumptions, a very large data pool ("big data") is available for exploitation and for building prediction models.

From the existing literature, it is unclear whether machine learning methods are able to outperform established approaches for retail demand forecasting (see Section 2) given the characteristics of the present case, which play to the strengths of machine learning. With this study, we address the following research questions:

- Are machine learning methods a viable alternative to established approaches for the given forecasting scenario?
- Which method is most suitable for forecasting demand on special days?
- Which machine learning method provides the most accurate predictions? Which modeling approach works best?
- How often do machine learning methods have to be re-trained so that their accuracy does not diminish?

We also extend the existing literature in multiple directions. First, we present a real-world daily demand forecasting application with an emphasis on special calendar days. Second, we elaborate on the possibilities of formulating time series forecasting as a machine learning problem. This includes a rather uncommon transformation of the regression problem to a classification problem. Third, we provide an empirical evaluation of state-of-the-art machine learning methods for large-scale demand forecasting. Based on a comparison with established approaches, we illustrate the viability of using machine learning in a productive setting.

The remainder of this paper is organized as follows: In Section 2, we provide an overview of related work and highlight research gaps that we want to address. We present a framework for modeling time series forecasting as a machine learning task in Section 3. In the subsequent section, we describe how the framework can be applied to our use case (see Section 4). We outline the experimental design of the empirical evaluation in Section 5. The results are presented and discussed in Section 6. In Section 7, we summarize our findings and outline opportunities for further research.

## 2. Related work

Our work addresses the challenge of large-scale demand forecasting with an emphasis on special days using machine learning methods. Thus, we review the literature on demand forecasting for special occasions (see Section 2.1) as well as literature on the application of machine learning methods to time series forecasting (see Section 2.2). We refer to Fildes, Ma, and Kolassa (2019) for a general literature review on retail forecasting.

### 2.1. Forecasting on special occasions

Forecasting in the retail domain primarily focuses on promotions rather than special days. However, the requirements of promotional forecasting are similar, and public holidays, as well as major festivals, are frequently considered in the proposed models. Cooper, Baron, Levy, Swisher, and Gogos (1999) present a promotional forecasting system for weekly retail data based on a regression-style model that incorporates dummy variables for public holidays. Divakar, Ratchford, and Shankar (2005) and van Heerde, Leeflang, and Wittink (2002) discuss the possibility of varying the scope of the model and fitting it to different levels of aggregation. Gür Ali, Sayın, Van Woensel, and Fransoo (2009) present a study of weekly forecasts of perishable goods that have a durability of several days at the store-product level. They state that data pooling improves the results, while only more sophisticated methods (e.g., regression trees) benefit from a more detailed input. According to Van Donselaar, Peters, de Jong, and Broekmeulen (2016), models that are

Please cite this article as: J. Huber and H. Stuckenschmidt, Daily retail demand forecasting using machine learning with emphasis on calendric special days. International Journal of Forecasting (2020), https://doi.org/10.1016/j.ijforecast.2020.02.005.

fitted over multiple categories are more accurate only if the data foundation is sufficient. Huang, Fildes, and Soopramanien (2014) propose a regression model for aggregated retail data, and observe that the accuracies of the evaluated approaches are largely comparable with normal weeks, while improvements are possible during promotional periods. Ma, Fildes, and Huang (2016) and Ma and Fildes (2017) report that integrating more data; i.e., cross-categorical information, leads to more accurate predictions. They also suggest that regular refitting and variable selection are required.

Trapero, Pedregal, Fildes, and Kourentzes (2013) and Trapero, Kourentzes, and Fildes (2015) report that judgmental adjustment for promotions adds value but is not better than statistical models. Kourentzes and Petropoulos (2016) stress the importance of an automatized approach for promotional forecasting. Ramanathan and Muyldermans (2010) present promotional factors affecting demand. These factors include special days (upcoming holidays, festivals such as Easter and Christmas), seasonal factors (e.g., temperature), and promotional factors. An evaluation based on structural equation modeling leads to the conclusion that the relevant factors depend on the product or product family (Ramanathan & Muyldermans, 2011). van Heerde, Leeflang, and Wittink (2000) observe that pre- and post-promotion effects also noticeably influence sales.

The aforementioned studies are based on weekly data as this level of granularity is sufficient for many operational decisions. While some special days are frequently modeled using binary dummy variables, they have not been of specific interest. We suspect that a reason for this is that the effect of special days is mitigated on the weekly level and possibly dominated by promotions. The few studies in the business forecasting literature that are dedicated to daily retail forecasting have also not emphasized the challenges related to special days. Taylor (2007) uses exponential smoothing to compute prediction intervals for daily supermarket sales. Public holidays and periods with unusual demand are explicitly excluded from the evaluation as the considered methods are not designed for those scenarios. Di Pillo, Latorre, Lucidi, and Procacci (2016) employ support vector machines to forecast daily retail data but do not address the challenges related to special days. Arunraj and Ahrens (2015) develop an S-ARIMAX model that incorporates binary dummy variables for holidays for the prediction of daily banana sales in a single store, but are also not concerned with the forecast accuracy on special days. Kolassa (2016) discusses challenges related to the evaluation of intermittent daily retail data and argues that it is important to evaluate predictive distributions instead of specific functionals for such data.

Special days are an explicit subject in the context of intraday load forecasts. Srinivasan, Chang, and Liew (1995) highlight the importance of modeling special days for hourly load forecasting using a fuzzy artificial neural network (ANN). They employ a dedicated model for each of the three day types: weekdays, Saturdays, and Sundays plus public holidays. Similarly, Wang and Ramsay (1998) also train one model per day type and report that the errors for public holidays are the highest, which is justified by the fact that those days fall in different seasons. Kim (2013) incorporates special days in a double seasonal ARIMA model by treating special days that are subject to a similar pattern identically. Cancelo, Espasa, and Grafe (2008) also highlight the importance of treatments for special days and events. Soares and Medeiros (2008) identify a total of 15 day types, including weekdays, days before and after public holidays, and bridge days. Panapakidis (2016) proposes clustering special days according to their load pattern. The cluster information and the average load of reference days are fed into an ANN. Barrow and Kourentzes (2018) model non-calendar special days in the context of call center call arrivals with an ANN. They report the superiority of this model over standard statistical models and provide empirical evidence that it is better to incorporate special days into the model rather than building separate models for special days.

### 2.2. Forecasting using machine learning

Statistical time series methods (e.g., exponential smoothing, ARIMA models) have been successfully applied to many forecasting problems, and there is no definite evidence that they are inferior to machine learning methods (e.g., Ahmed, Atiya, Gayar, & El-Shishiny, 2010; Crone, Hibon, & Nikolopoulos, 2011; Makridakis, Spiliotis, & Assimakopoulos, 2018b). The results of the most recent M4 competition suggest that (combinations of) statistical methods outperform pure machine learning methods, while a hybrid approach performed best at forecasting univariate time series (Makridakis, Spiliotis, & Assimakopoulos, 2018a). Ahmed et al. (2010) compare a variety of machine learning methods, including ANNs and regression trees, on a subset of the monthly time series of the M3 competition. They conclude that machine learning methods, especially ANNs, are contenders with classical statistical models. Makridakis et al. (2018b) concluded in a similar study that machine learning methods are inferior to statistical forecasting methods. However, the findings from Crone et al. (2011) highlight that no approach works best under all circumstances.

The most popular machine learning models with respect to time series forecasting are ANNs. ANNs have been extensively studied in the context of time series forecasting for more than two decades (Adya & Collopy, 1998; Zhang, Patuwo, & Hu, 1998). Alon, Qi, and Sadowski (2001) report that ANNs are superior to ARIMA models and multiple regression when it comes to forecasting monthly aggregate retail sales with a strong trend and seasonal patterns. The study by Chu and Zhang (2003) emphasizes that deseasonalization is preferable to other modeling options if ANNs are applied, while Crone and Kourentzes (2009) were able to model deterministic seasonality with trigonometric functions, which suggests that deseasonalization is not always required. Aburto and Weber (2007) proposed a hybrid demand forecasting approach for retail sales based on ARIMA and ANNs, in which the ANNs are trained on the residuals of the ARIMA model. Doganis, Alexandridis, Patrinos, and Sarimveis (2006) forecast the demand of short-shelf-life products with a radial basis function ANN whose variables are

Please cite this article as: J. Huber and H. Stuckenschmidt, Daily retail demand forecasting using machine learning with emphasis on calendric special days. International Journal of Forecasting (2020), https://doi.org/10.1016/j.ijforecast.2020.02.005.

selected using evolutionary computing techniques. The proposed model produces more accurate predictions than various linear reference methods. In contrast, Carbonneau, Laframboise, and Vahidov (2008) report that recurrent neural networks (RNNs) perform better than support vector machines, but do not outperform traditional approaches such as moving-average or linear regression in the context of monthly demand forecasting of a supply chain.

The vast majority of the comparative studies do not exploit the strength of data-driven machine learning methods since the results are mostly based on univariate time series forecasting. For instance, studies based on the M3 or NN3 dataset only cover monthly time series with 14 to 126 samples. Thus, the derived training dataset is also quite small, which quickly leads to an unfavorable ratio between the number of observations and the parameters of the model.

To summarize, we notice that when it comes to the retail sector, the literature is quite comprehensive but focuses on promotions rather than special days. However, there are still some gaps that need to be addressed. First, existing studies are mostly based on aggregated data with respect to organizational (store level vs. company level) and temporal (weekly data vs. daily data) hierarchies. The effect of special days is mitigated for weekly data and thus has never been the focus of existing studies. Second, the applied promotional forecasting models are mostly multivariate causal linear models or univariate time series models with adjustments in a post-processing step (e.g., the base-times-lift method). This is typically justified by the enhanced interpretability of the models. In our use case, the accuracy of a model is more important than its interpretability as the forecasts are eventually the input for automated decisions. This allows us to investigate data-driven machine learning methods that are not as easy to interpret and are often considered a black box. However, this does not hinder judgmental adjustment, which is common in practice if the demand is expected to deviate from the normal pattern. We argue that there is a need to evaluate machine learning methods on large-scale datasets covering time series that are enriched with explanatory data (e.g., domain knowledge). To address this gap, we aim to outline the modeling possibilities of supervised machine learning for time series forecasting with respect to the learning task (e.g., regression vs. classification) and the scope of the model (e.g., pooled regression). We are not aware of either published work that explores modeling possibilities in this direction or an evaluation of machine learning approaches on a large scale for daily retail data with an emphasis on special days.

## 3. Methodology

Retailers make many operational decisions based on forecasts that are calculated by time series forecasting methods (see Section 2). A univariate time series $Y = (y_1, \dots, y_n)$ with $y_t \in \mathbb{R}$ is a sequence of uniformly spaced time instants. Additionally, each data point $y_t$ of a time series $Y$ can be enriched by explanatory variables $X = (x_1, \dots, x_n)$, $x_t \in \mathbb{R}^p$ comprising information that is not contained in the original time series but can be exploited to understand and model the apparent patterns in $Y$. We consider a scenario that comprises a large set $\mathcal{S} = \{(X_1, Y_1), \dots, (X_N, Y_N)\}$ of time series $Y_s \in \mathbb{R}^{n_s}$ and their respective explanatory variables $X_s \in \mathbb{R}^{n_s \times p}$. We allow a varying length $n_s$ of the time series $Y_s \in \mathcal{S}$, while the structure of the external information has to be identical for all time series. The forecasting task is to predict the next value of $(y_{n_s+1})$ (single-step forecast) or the next $h$ values $(y_{n_s+1}, \dots, y_{n_s+h})$ (multistep forecast) (Ben Taieb, Bontempi, Atiya, & Sorjamaa, 2012; Bontempi, Ben Taieb, & Borgne, 2012) of a time series $Y_s$ based on the available information.

### 3.1. Machine-learning-based demand forecasting framework

Machine learning methods offer features that are well suited for the present forecasting scenario. They are designed to learn patterns from data and are naturally able to process large datasets. Therefore, they do not impose assumptions on the data. This holds for the data generating process as well as the scope of the model. It is even possible to employ $k$-fold cross validation (Barrow & Crone, 2016; Bergmeir, Hyndman, & Koo, 2018) on the full dataset to validate and design models as the ordering of the observations does not have to be preserved. Moreover, the methods do not explicitly distinguish between information that directly stems from the time series and external data. The characteristics and flexibility of machine learning methods make them, in principle, a viable alternative to the established approaches (e.g., multivariate linear regression models and univariate statistical time series models). In the next section, we frame time series forecasting as a machine learning task and outline various modeling options.

We extend the introduced notion of time series data and forecasting and adapt it to machine learning. Time series forecasting can be framed as a supervised machine learning task. The goal is to approximate a function $f(\cdot)$ that models the relation between a vector of quantitative input variables, known as *features*, $X \in \mathbb{R}^p$, and a quantitative output variable, known as the *target*, $Y \in \mathbb{R}$ (Hastie et al., 2009):

$$\hat{Y} = f(X) \tag{1}$$

The predicted value is denoted by $\hat{Y}$, while the observed value is $Y$. To employ a machine learning algorithm, we need to provide a (training) dataset $\mathcal{D} = \{(x_k, y_k)\}_{k=1..n}$ consisting of a set of tuples containing features $x_k \in \mathbb{R}^m$ that describe a target $y_k \in \mathbb{R}$. $\mathcal{D}$ can also be expressed as a pair $(\mathbf{X}, \mathbf{Y})$ of a feature matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ and a target vector $\mathbf{Y} \in \mathbb{R}^n$.

Based on the observed training data, a machine learning algorithm approximates a function $f(\cdot)$ by minimizing a loss function $\mathcal{L}(Y, f(x))$ that assesses the fit of $f(\cdot)$. A loss function $\mathcal{L}$ is typically a globally continuous and differentiable function; e.g., the squared loss (L2-norm) can be used for regression tasks:

$$\mathcal{L}(Y, f(X)) = (Y - f(X))^2 \tag{2}$$

Please cite this article as: J. Huber and H. Stuckenschmidt, Daily retail demand forecasting using machine learning with emphasis on calendric special days. International Journal of Forecasting (2020), https://doi.org/10.1016/j.ijforecast.2020.02.005.

The parameters $\theta$ of $f(\cdot)$ need to be tuned to minimize the loss function $\mathcal{L}$. A common problem is that the approximated function $f(\cdot)$ has a much lower error in the training dataset $\mathcal{D}^{train}$ than in an unseen test dataset $\mathcal{D}^{test}$; i.e., it does not generalize well. This phenomenon is known as overfitting. To control and prevent this, a validation dataset $\mathcal{D}^{valid}$ is usually retained, which can be used to test whether the model does overfit the training dataset. Bergmeir et al. (2018) show that $k$-fold cross-validation is suitable for controlling overfitting when machine learning methods are employed.

*3.1.1. Forecasting as a machine learning task*

Time series data, along with explanatory information, can be transformed to feature-target pairs that can be processed by a machine learning algorithm. For instance, a time series $(y_1, \dots, y_n)$ representing an autoregressive $AR(a)$ process can be formulated as follows (Adya & Collopy, 1998; Zhang et al., 1998):

$$\mathbf{X} = \begin{bmatrix} y_1 & y_2 & \dots & y_a \\ \vdots & \vdots & \vdots & \vdots \\ y_{t-a} & y_{t-a+1} & \dots & y_{t-1} \\ \vdots & \vdots & \vdots & \vdots \\ y_{n-a} & y_{n-a+1} & \dots & y_{n-1} \end{bmatrix} \quad \mathbf{Y} = \begin{bmatrix} y_{a+1} \\ \vdots \\ y_t \\ \vdots \\ y_n \end{bmatrix}. \tag{3}$$

Hence, the lagged time series observations build the feature matrix $\mathbf{X}$. For the present application scenario, each time series is enhanced with explanatory information; i.e., $(X_s, Y_s) \in \mathcal{S}$. Therefore, $\mathbf{X}$ can be extended with the provided explanatory information:

$$\mathbf{X} = \begin{bmatrix} y_1 & y_2 & \dots & y_a & x_{a+1,1} & \dots & x_{a+1,p} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{t-a} & y_{t-a+1} & \dots & y_{t-1} & x_{t,1} & \dots & x_{t,p} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{n-a} & y_{n-a+1} & \dots & y_{n-1} & x_{n,1} & \dots & x_{n,p} \end{bmatrix}. \tag{4}$$

The present model is a multiple regression as a single target variable depends on multiple variables covering autoregressive and external information. A machine learning method (see Section 3.2) can be employed to approximate a functional relation in a data-driven fashion. Therefore, it does not distinguish the origin of the feature variables; i.e., whether they are autoregressive or external because the semantics are hidden from the method.

*3.1.2. Model scope*

We describe in (3) and (4) how a dataset $D_s$ can be obtained from $(X_s, Y_s) \in \mathcal{S}$. As we consider multiple time series, we have a set of datasets $\{D_1, \dots, D_N\}$ that are derived from $\mathcal{S}$ and thus have the same structure. This makes it possible to unify them into one dataset $\mathcal{D} = \cup_{s=1..N} D_s$. Hence, we can vary the scope of the model; i.e., we can train a model $f(\cdot)$ on an arbitrary subset of $\mathcal{D}$. Training a model on a unified dataset can be interpreted as pooled regression. We explicitly do not consider modeling the forecasting problem as a multivariate regression problem by treating each time series as a different dependent variable. Multivariate regression is not well suited for this application as we deal with time series of different lengths and a changing number of time series at a given point in time. For instance, it might be possible that a retailer opens or closes a store.

It is common practice to select and fit models for each time series, which is often the most appropriate approach. However, our application scenario allows us to employ pooled regression. Therefore, the size of the training data that are available to the training algorithm can be significantly increased. This makes it more likely that the machine learning method will separate actual patterns from noise in the data and reduces the likelihood of overfitting the data. Consequently, the trained model should be more robust and ultimately have an improved accuracy. Another advantage is that a globally trained model can be applied to new or short time series (e.g., new stores or articles, or a changing assortment). Reducing the number of models that need to be maintained (e.g., through feature engineering, hyperparameter optimization, or persisting) makes this approach more viable in practice. However, this only makes sense if common patterns are present in multiple time series and can be transferred.

Thus far, we have outlined the advantages of unifying datasets that are derived from different time series. However, we have also pointed out that the pattern of the time series should be comparable. Hence, one might suggest that it makes sense to cluster time series in $\mathcal{S}$ and train one model per cluster. However, clustering is an unsupervised task, and it is not simple to identify features that describe clusters so that the resulting forecast accuracy is minimized. Moreover, clusters are often fuzzy and do not allow a clear distinction between groups. Kang, Hyndman, and Smith-Miles (2017) and Petropoulos, Makridakis, Assimakopoulos, and Nikolopoulos (2014) propose using time series features to infer a model that is most suitable for forecasting a specific time series. We suggest expanding the feature space with time-invariant features (e.g., the location of a store or the features of a product) that allow a machine learning method to implicitly cluster time series while minimizing the loss function. Therefore, this end-to-end training of a model solves the problem of explicitly identifying subsets of $\mathcal{D}$ based on fuzzy clusters. However, machine learning methods are in principle able to distinguish between different time series and adapt the parameters $\theta$ of the model accordingly if relevant features are provided.

In summary, the application of machine learning allows us to vary the scope of a model. By unifying datasets derived from time series, it is possible to fit models to larger datasets, which makes it more likely that noise will be separated from actual demand patterns while reducing the risk of overfitting. To distinguish between different time series during the training process, it is possible to add time-invariant features that allow the model to implicitly cluster time series. A global model is also easier to maintain, which makes the approach viable in practice.

Please cite this article as: J. Huber and H. Stuckenschmidt, Daily retail demand forecasting using machine learning with emphasis on calendric special days. International Journal of Forecasting (2020), https://doi.org/10.1016/j.ijforecast.2020.02.005.

**Table 1**
Transformation of a regression problem into a classification problem. The values of the dependent variable of the regression problem; i.e., the target values of the machine learning task, need to be binned and allocated to classes. Consequently, a numerical class value needs to be assigned to each target class; e.g., the average of the quantitative target values comprised by the class.

| Target | Class | Class value |
|---|---|---|
| 1 | $g_1$ | 4 |
| 2 | $g_2$ | 2.5 |
| 3 | $g_2$ | 2.5 |
| 4 | $g_3$ | 4 |

**Table 2**
*One-hot encoding* allows the model to directly predict the probability for each target class ($g_1 < g_2 < g_3$).

| Class | Encoding | | |
|---|---|---|---|
| $g_1$ | 1 | 0 | 0 |
| $g_2$ | 0 | 1 | 0 |
| $g_3$ | 0 | 0 | 1 |

### 3.1.3. Regression vs. Classification

Supervised machine learning distinguishes between *regression* and *classification*. It is more natural to model time series forecasting as a regression task as the target values are quantitative. However, it is also possible to convert a regression problem into a classification problem as both are essentially function approximation tasks (Hastie et al., 2009). The target values $G$ of a classification task are qualitative. Thus, a surjective mapping $\mathcal{G} : Y \rightarrow G$ between quantitative values $Y$ and qualitative values $G$ needs to be defined. The mapping $\mathcal{G}$ may group multiple target values; i.e., values of the dependent variable of the regression problem, which means that the granularity of the mapping also impacts the accuracy. Moreover, it is also necessary to define an inverse function $\mathcal{G}^{-1} : G \rightarrow Y$ to obtain a numeric value for the predicted class that represents the forecast. An example is provided in Table 1. After binning the values, we have a multiclass classification problem with $K$ levels: $G = \{g_1, \ldots, g_K\}$. The classes $g_k \in G$ are typically represented by one-hot encoded binary $K$-dimensional vectors; i.e., each element of the vector represents a class (see Table 2), but other encodings are also possible. For instance, as the variables $G$ are ordinal (ordered categorically: an ordering exists, but no metric is appropriate), the encoding can be adapted to exploit this information (Cheng, Wang, & Pollastri, 2008; Gutiérrez, Pérez-Ortiz, Sánchez-Monedero, Fernández-Navarro, & Hervás-Martínez, 2016) (see Table 3). One drawback to ordinal encoding is that either a cut-off value needs to be set to choose the predicted class or a subsequent regression model needs to be trained. In contrast, one-hot encoding provides a full probability distribution over the target classes.

The transformation into a classification problem has several advantages. The standard regression approach directly predicts only one value (e.g., mean or quantile) depending on the loss function $\mathcal{L}$. To obtain the probability distribution of the forecast, one has to make distributional assumptions or rely on historical data (e.g., an empirical distribution) (Kolassa, 2016). In contrast, classification

**Table 3**
Ordinal encoding allows the model to predict the probability that the target is greater than or equal to the value of a specific class ($g_1 < g_2 < g_3$).

| Class | Encoding | | |
|---|---|---|---|
| $g_1$ | 1 | 0 | 0 |
| $g_2$ | 1 | 1 | 0 |
| $g_3$ | 1 | 1 | 1 |

models can explicitly predict a complete (discrete) probability distribution over all classes, which is basically a density forecast, and allows us to obtain not only the mean of the distribution but also different quantiles or the mode.

### 3.1.4. Additional remarks

The presented demand forecasting framework already demonstrates flexibility by providing many options to model time series forecasting in a way that enables processing by machine learning methods. However, there are still open questions that need to be considered before applying machine learning.

First, a number of machine learning methods are available; thus, the selection of a method is an important decision. Depending on the method, model-specific parameters need to be set and optimized. This includes the architecture of the model (e.g., the size of the trees, the number of hidden layers and nodes, and activation functions) and hyperparameters (e.g., the learning rate). To optimize the hyperparameters of a machine learning model, grid search, random search (Bergstra & Bengio, 2012), tree of Parzen estimators (Bergstra, Bardenet, Bengio, & Kégl, 2011; Bergstra, Yamins, & Cox, 2013), and Bayesian optimization (Snoek, Larochelle, & Adams, 2012) have been proposed.

Second, the dataset $D$ can be sliced along two dimensions: samples and features. By selecting a subset of samples, it is possible to adjust the scope of the model. While it would be preferable to obtain a global model, it might be beneficial to specify models for different subsets. We also want to point out that the scope of the model can change between the training phase and the application phase. For instance, for some time series, the best option might be to use a global model, while other time series may be more accurately forecast when the model is only trained on its own data, while the data are also used to train the global model. While we assume that identical features are provided for each time series in $\mathcal{S}$, it might be beneficial to only use a subset of the provided features (feature selection). There is already a research stream that focuses on automating the whole machine learning process and partially addresses the first two issues (Bischl et al., 2016; Feurer et al., 2015). It can be expected that at least some time series forecasting applications will benefit from those approaches.

Third, with regard to the application of a model, it must be decided how frequently the model needs to be retrained. This has to be decided carefully because the training process is quite time consuming. If the model is trained on a large data basis and is able to learn a

Please cite this article as: J. Huber and H. Stuckenschmidt, Daily retail demand forecasting using machine learning with emphasis on calendric special days. International Journal of Forecasting (2020), https://doi.org/10.1016/j.ijforecast.2020.02.005.

representation of the existing patterns, regular training is often not required as it does not add much value. During the training phase, the ordering of the observations is not preserved, which means that no special emphasis is placed on the most recent observations. However, it is also possible to update a pretrained model and adjust its parameters $\theta$ by fine-tuning the model with the most recent observations.

Fourth, the current formulation of the forecasting problem only allows one-step predictions. For multistep forecasts, it is possible to iteratively apply the model. However, there are other alternatives that can be considered; for instance, training one model per forecasting step or constructing a model with multiple outputs. Different possibilities have been proposed and evaluated by Ahmed et al. (2010), Ben Taieb et al. (2012), and Bontempi et al. (2012).

### 3.2. Machine learning methods

In this study, we consider feed-forward ANNs (FNNs), RNNs, and gradient-boosted regression trees (GBRTs). All these methods rely on a gradient-based approach to optimize the model parameters $\theta$. However, each method processes data in different ways, as we outline below.

#### 3.2.1. Feed-forward neural networks

FNNs, such as the multilayer perceptron (MLP), have been the most popular neural network architecture for time series forecasting over the past decades (Zhang et al., 1998). In an FNN with $L$ hidden layers ($L \geq 1$), the output of each layer $h^{(k)}(x)$ passes to the next layer ($1 \leq k \leq L+1$):

$$h^{(k)}(x) = \sigma^{(k)}(b^{(k)} + W^{(k)} h^{(k-1)}(x)). \tag{5}$$

The output of the input layer is defined as $h^0(x) = x$, while the output of the last layer represents the prediction of the network; i.e., $f(x) = h^{(L+1)}(x)$. The output of each layer is connected by a fully connected weight matrix $W^{(k)}$ to the next layer. The input of a layer is adjusted with the biases $b^{(k)}$ of each neuron before it passes an activation function $\sigma^{(k)}$. Frequently used activation functions are:

- a logistic function: $\sigma_{sigmoid}(x) = \frac{1}{1+e^{-x}}$
- a hyperbolic function: $\sigma_{tanh}(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$
- rectified linear activation: $\sigma_{relu}(x) = x^{+} = \max(0, x)$
- exponential linear activation: $\sigma_{elu}(x) = \begin{cases} x & \text{if } x \geq 0 \\ e^{x} - 1 & \text{otherwise} \end{cases}$
- a linear function: $\sigma_{linear}(x) = x$
- a softmax function: $\sigma_{softmax}(x) = [\frac{exp(x_1)}{\sum_c exp(x_c)} \cdots \frac{exp(x_C)}{\sum_c exp(x_c)}]$

In this work, we consider $\sigma_{relu}(x)$ and $\sigma_{elu}(x)$ to be the activation functions of the hidden layers. At the output layer, we employ $\sigma_{linear}(x)$ for the regression approach and $\sigma_{softmax}(x)$ for the classification approach. The activation function $\sigma_{softmax}(x)$ provides the probability of each target class.

The performance of an ANN also depends on its initial weights, which are set randomly. Therefore, we employ an ensemble of ANNs with the *median* ensemble operator as this approach is robust to the initial weights and provides reliable results (Barrow, Crone, & Kourentzes, 2010; Kourentzes, Barrow, & Crone, 2014). To optimize the weights of an ANN, we use the stochastic gradient-based algorithm ADAM proposed by Kingma and Ba (2015).

#### 3.2.2. Recurrent neural networks

RNNs process input features in a sequential order and apply the same network to each step in a sequence. RNNs maintain an internal memory that allows them to track dynamic patterns. We use a variant of RNNs known as long short-term memory (LSTM) (Hochreiter & Schmidhuber, 1997), which has a sophisticated memory concept based on input gates $i_t$, output gates $o_t$, forget gates $f_t$, and a cell state $c_t$:

$$f_t = \sigma_{sigmoid}(W_f x_t + U_f h_{t-1} + b_f) \tag{6}$$

$$i_t = \sigma_{sigmoid}(W_i x_t + U_i h_{t-1} + b_i) \tag{7}$$

$$o_t = \sigma_{sigmoid}(W_o x_t + U_o h_{t-1} + b_o) \tag{8}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_{tanh}(W_c x_t + U_c h_{t-1} + b_c) \tag{9}$$

$$h_t = o_t \circ \sigma_{tanh}(c_t). \tag{10}$$

The operator $\circ$ is a Hadamard product; i.e., an elementwise multiplication of matrices and vectors that have the same dimension. The parameters of an LSTM unit are $W$, $U$, and $b$. The parameters can be trained with a stochastic gradient-based algorithm such as ADAM. The output of an LSTM cell $h_t$ can be passed through an arbitrary activation function, which makes LSTM suitable for regression and classification.

#### 3.2.3. Gradient-boosted regression trees

GBRTs have gained much interest in recent years and are an alternative to ANNs for structured data. The most popular implementations are *LightGBM* (Ke et al., 2017) and *xgboost* (Chen & Guestrin, 2016). Like any boosting algorithm, they train a series of simple models $f_k(x)$ (i.e., decision trees) based on the accumulated residuals of the previous model $\mathcal{L}^{(t)}(y, \hat{y}^{(t-1)} + f_t(x))$. Hence, the prediction is the sum of all the trained simple models $f_k(x)$; i.e., $f(x) = \sum_{k=1}^{K} f_k(x)$.

## 4. Application to the retail domain

We apply the outlined framework to a real-world use case of an industrialized bakery running a centralized production facility, from which more than 100 stores are supplied on a daily basis. Thus, daily short-term forecasts are required for many operational decisions concerning not only production and delivery but also other areas such as staffing. Hence, this use case represents a large-scale demand forecasting setting whose requirements are representative of many retailers.

All operational figures are derived from observations at the lowest organizational level, which is the store-article level. Hence, the primary patterns that are present at this level also propagate to aggregated levels. We will focus on sales at the store-category level, which comprises a group of products having comparable characteristics.

Please cite this article as: J. Huber and H. Stuckenschmidt, Daily retail demand forecasting using machine learning with emphasis on calendric special days. International Journal of Forecasting (2020), https://doi.org/10.1016/j.ijforecast.2020.02.005.

**Table 4**
Public holidays in Germany and Baden-Württemberg(*). The weekday changes if a date is stated. Public holidays that are aligned with Easter Sunday occur on a fixed weekday. Easter Sunday is the first Sunday after the first full moon in spring (22.03.–25.04.).

| Date | Description | Comment |
|---|---|---|
| 25.12. | 1. Christmas Day | |
| 26.12. | 2. Christmas Day | |
| 01.01. | New Year | |
| 06.01.* | Epiphany | |
| Friday | Good Friday | Easter Sunday−2 |
| Monday | Easter Monday | Easter Sunday+1 |
| Thursday | Ascension Day | Easter Sunday+39 |
| Monday | Whitmonday | Easter Sunday+50 |
| Thursday* | Corpus Christi | Easter Sunday+60 |
| 01.05. | Labor Day | |
| 03.10. | Day of German Unity | |
| 01.11.* | All Hallows | |

We consider the most common product categories, including buns, breads, viennoiseries, cakes, and snacks. These categories are typically provided by companies in the bakery industry. We evaluate our approach toward aggregated data as it is cleaner; for instance, demand distortion due to stock-outs is reduced because of high substitution rates (Van Woensel et al., 2007), and we do not have to deal with challenges that arise from a changing assortment. However, the data foundation is still sufficient for evaluating the proposed machine learning approach. Moreover, the considered time series allow the application of exponential smoothing models and linear regression models because the history is sufficient and the number of "missing values" is limited. Forecasts at the store-category level are important as they are also an input for operational planning by the staff, who follow a top-down planning approach. A large share of the stores also open on public holidays. However, even stores that are not open on public holidays are affected on the preceding and following days.

### 4.1. Day classification

One particular challenge is to provide demand forecasts on special days (SDs); i.e., days on which the demand vastly differs from the regular patterns due to a calendar event. Special days are primarily triggered by public holidays. Public holidays often fall on working days but have much in common with Sundays. First, the stores are operated as they are on Sundays with respect to the assortment and the opening times. Second, the working schedules of most people is comparable with those on Sundays as they do not have to work. This is due to the German constitution, which states that public holidays are equal to Sundays with respect to the permission to open a store or work. However, other festivities that are not related to official public holidays such as carnivals can be considered special days. In the case of public holidays, not only the actual day but also the surrounding days are affected. Hence, we will use the following special day classification for the evaluation:

- **SD1: special day** (**t**): The day of the public holiday or another significant event.
- **SD2: day before** (**t − 1**): The day before a public holiday is affected as people tend to stockpile. If a public holiday falls on a Monday, the previous Saturday ($t-2$) also belongs to this class.
- **SD3: day after** (**t+1**): The day after a public holiday. This can also be a bridge day; i.e., a day between a public holiday and the weekend.
- **SD4: following week** (**t + 7**): The demand reverts back to the normal pattern in the following week. As demand has a strong weekly seasonality, we use this day type as a sanity check. Autoregressive models might underestimate or overestimate demand if the demand pattern is not appropriately learned by the model.

The types of days are ordered according to their precedence: $t > t-1 > t+1 > t+7$. A date is only assigned to the class that has the highest precedence if multiple classes apply. The neighboring days ($t-1, t+1$) are only of interest for public holidays as no noticeable effects are observed for other special days. We refer to days that are not part of any of the aforementioned classes as SD0 (i.e., regular days).

It is particularly difficult to forecast the demand on special days as the number of historical observations is limited; i.e., there is only one observation per time series and year, and different special days are subject to different changes. For instance, sales at Christmas are not comparable to those on any other special day. Hence, there are also long-term relationships that need to be considered. However, it is also not sufficient to only consider the observed sales of the previous year since the weekday changes for some special days and the demand may be

**Table 5**
Special days that are not public holidays but are also subject to different demand patterns.

| Date | Description | Comment |
|---|---|---|
| 24.12. | Christmas Eve | |
| 31.12. | New Year's Eve | |
| 7 weeks before Easter | (Thu) Women's Carnival Day<br>(Fri) Carnival Friday<br>(Sat) Carnival Saturday<br>(Sun) Carnival Sunday<br>(Mon) Carnival Monday<br>(Tue) Carnival<br>(Wed) Ash Wednesday | Carnival |
| Sunday | Easter Sunday | see Table 4 |
| Sunday | Whitsunday | Easter Sunday+49 |

Please cite this article as: J. Huber and H. Stuckenschmidt, Daily retail demand forecasting using machine learning with emphasis on calendric special days. International Journal of Forecasting (2020), https://doi.org/10.1016/j.ijforecast.2020.02.005.

subject to a general trend or level shift. Moreover, some public holidays are on a fixed date, while others are on a fixed weekday.

In Germany, there are nine state-wide public holidays as well as additional public holidays that are set by each federal state, including three for Baden-Württemberg (see Table 4). The public holidays that are related to Easter are on a fixed weekday but can shift by up to one month across years. The other public holidays have a fixed date but a changing weekday. Therefore, not only the number of observations but also the comparability is limited. The same is true for the special days given in Table 5.

### 4.2. Feature engineering

The challenge in applying machine learning is the engineering of features that allow the prediction of the desired output and often depend on domain knowledge. We outline the features that we use for our concrete use case. However, most features are quite general features that are available for most retailers. For demand forecasting, it seems natural to rely on autoregressive time series data as well as external information. The most important information source is the enterprise resource planning system, which contains master data and transactional data. In general, we rely on the following data sources:

**Master data** comprise information about the available stores and products. Stores have a fixed location (i.e., address) that is required to enhance the data with external information. The data also contain the opening times as they vary among the stores; e.g., not all stores open on Sundays or public holidays. Stores are also assigned to predefined store classes that roughly represent their characteristics; e.g., located in a mall or associated with a supermarket or a coffee house. The products are assigned to specific categories and allow us to obtain the aggregated demand at the category level.

**Transactional data** contain the sales of the provided products, which represent the target variable. We also derive input features based on the target variable. For instance, we consider lagged sales because we expect an autoregressive pattern in the data and rolling features (e.g., a rolling seasonal median). However, they can also be exploited to compute other (time-invariant) derived features such as a general weekday pattern (working day, Saturday, Sunday). We also compute special day-specific features based on historical sales data. The bakery also distributes coupons that are valid for a couple of weeks. The information about active coupon periods (i.e., days when coupons are valid) is available as a binary indicator variable at the company level.

**External data** comprise location-specific data and calendar information. The calendar allows us to obtain the day of the week or the day of the year as we deal with multiple seasonalities. It also contains public holidays, noted special days (see Tables 4 + 5), and school holidays. The calendar information depends on the federal state in which the store is located. In addition to the store classes obtained from the master data, we enhance the dataset with location-specific features that describe the local environment of each store.

A brief overview of the considered features is provided in Table 6. With regard to special days, we can either include them in the model or replace and adjust the prediction in a postprocessing step. As our goal is to build a global model, we introduce specific features based on the approach of domain experts, who typically identify reference days in previous years to predict the expected demand. Hence, we also compute features that are based on the history of each specific special day. The features cover the effect of the special day compared with a regular day. Hence, we measure the absolute and relative change of a special day compared to a rolling median of each weekday. This is necessary as the weekday might change but the total demand might be comparable. To cover level changes, we also include the historical rolling median as a feature. We do not consider including an estimate of the concrete value (e.g., a lagged observation from the previous year) as a feature as this would place too much emphasis on it and cause the model to overfit; i.e., the future forecast accuracy would heavily depend on the feature. More precisely, we create the following features:

**historical demand level** The level is defined as the rolling seasonal median for a specific weekday and is an estimation for the day as if it was not a special day. Public holidays are always compared to Sundays, while the other special days are compared to their actual weekdays.

**absolute change** The absolute change on a special day compared to the level.

**relative change** The relative change on a special day compared to the level.

**relative change (store class)** Additionally, we compute the average relative effect over all stores having the same store class. This effect is more robust as it is based on a larger data basis.

As we distinguish between public holidays and the remaining special days, we define a total of eight special day-specific features. To obtain the values of the features for the forecast horizon, we apply a weighted rolling mean over the history such that the feature values do not only depend on the previous year (see Table 7). If a special day can fall on different weekdays, we consider the historical comparison with the weekday that is relevant within the forecast horizon.

Overall, we consider more than 250 features. The high dimensionality makes it necessary to increase the scope of the model to have more training data available and to reduce the risk of overfitting (e.g., the curse of dimensionality). Moreover, for methods based on ANNs, it is necessary to preprocess some features to make them processable by the models. For instance, it is beneficial if the variables

Please cite this article as: J. Huber and H. Stuckenschmidt, Daily retail demand forecasting using machine learning with emphasis on calendric special days. International Journal of Forecasting (2020), https://doi.org/10.1016/j.ijforecast.2020.02.005.

**Table 6**
Overview of the feature groups considered for the machine learning methods.

| Source | Feature list |
|---|---|
| Master data | Store class, product category, opening times (day, hours/duration) |
| Transactional data | Lagged sales, rolling median of sales, binary promotional information (lags: 2–7, 14); calculated per store: weekly sales pattern, sales distribution over categories, derived specific features of special days |
| External: calendar | Day of year, month, day of month, weekday, public holiday, day type, bridge day, non-working day, indicators for each special day, school holidays |
| External: location | General location (city, suburb, town); in proximity to the store: shops (numbers and types: bakery, butcher, grocery, kiosk, fast-food, car repair), amenities (worship, medical doctors, hospitals), leisure (playground, sport facility, park), education (kindergarten, school, university) |

**Table 7**
For each special day, we compute the level (i.e., rolling seasonal median forecast) as well as the absolute and relative changes between the level and the target (i.e., sales on the special day). In the test period, we compute a weighted rolling mean (weights: 2015: 1; 2016: 2) of historical observations in order to obtain the feature values for the absolute or relative change.

| Year | Target | Level | Abs. change | Rel. change |
|---|---|---|---|---|
| 2015 (train) | 130 | 100 | 30 | 0.3 |
| 2016 (train) | 150 | 125 | 25 | 0.2 |
| 2017 (test) | – | 100 | 26.67 | 0.233 |

are scaled (e.g., min–max scaling) or transformed (e.g., log transformation) to a fixed range (e.g., $[-1, 1]$) and are standardized (i.e., zero mean and unit variance) (LeCun, Bottou, Orr, & Müller, 2012). Moreover, deterministic seasonalities (e.g., weekly seasonality) can be modeled with trigonometric functions (Crone & Kourentzes, 2009).

In this study, we apply a log transformation and then linearly scale the target variable and features directly obtained from it (i.e., autoregressive features) to a range $[-0.5, 0.5]$ as this is beneficial for backpropagation (LeCun et al., 2012). When we transform the regression problem to a classification problem, we create a bin for each percentile, or more frequently if the relative increase between neighboring bins exceeds 10%. In total, we obtain 124 bins as the data are sparser for higher target values. We set the numerical value of a class to the mean of the interval endpoints. For the RNNs, we create short sequences covering the lags as they are provided to the direct approaches, and only vary dynamic features such as sales and weekdays. Hence, the same information is provided to all machine learning methods.

## 5. Empirical evaluation

We conduct an empirical evaluation based on a real-world application as we are using the daily point-of-sales data from a large bakery chain (see Section 4). The evaluation aims to assess the performance of machine learning methods and provides a comparison with state-of-the-art time series methods. In particular, we investigate how well the models predict sales for different types of special days. By reporting the empirical forecast performance, we also stress the importance of considering special days during the model building process. In Section 5.1, we outline the experimental setting, including a brief description of the dataset, the configuration of the machine learning models, and the evaluation criteria. Then, we introduce the reference methods in Section 5.2. The results of the evaluation are reported and discussed in Section 6.

**Table 8**
Mean, standard deviation (sd), and quantiles of the number of observations per time series.

| Mean | Sd | 0.05 | 0.10 | 0.25 | 0.50 | 0.75 | 1.00 |
|---|---|---|---|---|---|---|---|
| 727.21 | 251.38 | 239 | 362 | 510 | 815 | 979 | 992 |

**Table 9**
Number of observations of each special day type (SD); i.e., sales greater than zero, in the training dataset and the test dataset. The training dataset comprises data from 2014-10-01 to 2017-01-31 while the test set comprises data from 2017-02-01 to 2017-06-30.

| SD | Training dataset | | Test dataset | |
|---|---|---|---|---|
| | N | N [%] | N | N [%] |
| 0 | 568,653 | 83.88 | 107,427 | 75.48 |
| 1 | 36,463 | 5.38 | 10,369 | 7.29 |
| 2 | 15,651 | 2.31 | 6,869 | 4.83 |
| 3 | 16,004 | 2.36 | 4,461 | 3.13 |
| 4 | 41,192 | 6.08 | 13,200 | 9.27 |
| $\Sigma$ | 677,963 | 100.00 | 142,326 | 100.00 |

### 5.1. Experimental setup

The dataset contains daily sales at the store-category level for 8 product categories in 141 stores; i.e., 1128 time series. The number of observations per time series varies as some stores are always closed on certain weekdays. Moreover, the available sales history of stores varies due to new openings. For over 90% of the time series, the sales history covers at least one year (see Table 8).

We split the data into a training period and a test period (see Table 9). The training period comprises all observations from October 2014 to January 2017. The test period comprises 150 days between 2017-02-01 and 2017-06-30. This is the most interesting period of the year with respect to our motivation as most special days fall within this time span. In fact, we classify 39 (26%) days in the test period as special days. This includes 15 (10%) days of type SD1, 7 (4.67%) days of type SD2, 5 (3.33%) days of type SD3, and 14 (9.22%) days of type SD4. The number of neighboring days (SD2 + SD3) is smaller than the number of special days (SD1) as they are only considered for public holidays. The distribution of day types is not matched

Please cite this article as: J. Huber and H. Stuckenschmidt, Daily retail demand forecasting using machine learning with emphasis on calendric special days. International Journal of Forecasting (2020), https://doi.org/10.1016/j.ijforecast.2020.02.005.

exactly by the number of observations in the test dataset (see Table 9) as some stores are closed on public holidays.

With regard to the machine learning methods introduced in Section 3.2, we only rely on data from the training dataset to select the parameters and to train the models. Thus, we employ cross-validation within the training period as we also need validation data to select the best (hyper-)parameters and architecture of the models. We also rely on the validation dataset to apply early stopping. For this purpose, we create 10 stratified samples based on the day type and product category, whereby 80% of the full training dataset is used for training and the remaining 20% serves as validation data.

We select the best configuration for each method based on the forecast accuracy on the validation datasets of the 10 samples. In more detail, we compute the sum of the day type-specific RMSEs for each sample; i.e., the ranking criterion is $RMSE_{SD0} + RMSE_{SD1} + RMSE_{SD2} + RMSE_{SD3} + RMSE_{SD4}$. Then, we exclude the two best and the two worst results and compute the average RMSE over the remaining samples per model configuration. We choose the configuration that performs best on the validation datasets and train 40 models on 40 additional samples. This is necessary since neural network approaches require an ensemble of models to produce more reliable results. We employ the median ensemble operator to combine the predictions of the 50 trained models. For the classification approach, we re-scale the sum of the probabilities to 1.0 after computing the median class probability over the samples and before we determine the predicted class and resolve the numeric value of the class. The classification approach is only evaluated for ANNs as preliminary experiments with *LightGBM* did not terminate.

The selected ANN architectures are provided as part of the supplemental material. However, with respect to the ANN architectures, we observe that the capacity of the hidden layers is smaller for the classification approach than the regression approach. One explanation for this is that the output layer has not just one node but 124 nodes; i.e., this is the number of predefined classes. The larger number of output nodes implies more trainable weights between the last hidden layer and the output layer. For all machine learning approaches, we train global models that forecast every time series.

To measure the accuracy of the predictions, we rely on the seasonal mean absolute scaled error (*MASE*) (Hyndman & Koehler, 2006) and the mean absolute error (*MAE*):

$$MASE = \frac{1}{N}\sum_{n=1}^{N}\frac{|y_n - \hat{y}_n|}{\frac{1}{|T_{SD0}|}\sum_{t\in T_{SD0}}|y_t - y_{t-m}|} \tag{11}$$

$$MAE = \frac{1}{N}\sum_{n=1}^{N}|y_n - \hat{y}_n|. \tag{12}$$

The denominator of *MASE* is the scaling factor of the absolute errors and is determined within the training set for each time series by comparing the *seasonal naïve forecast* to the actual values for days of type SD0. Hence, the set $T_{SD0}$ comprises all days of type SD0 and $m = 7$. We only report relative error measures for a better interpretability of performance gains and for reasons of confidentiality; e.g., $MAE_{rel} = \frac{MAE_{M2}}{MAE_{M1}}$ is the relative performance of method M2 compared with method M1 with respect to *MAE*. Moreover, we apply the Wilcoxon signed-rank test to determine if there are significant differences among the evaluated methods at the 0.05 significance level. Most results are shown as graphs, and the result tables are contained in the supplemental materials, where we also provide the results with respect to the symmetric mean absolute percentage error (*SMAPE*) and the root-mean-squared error (*RMSE*).

### 5.2. Baseline & reference methods

We compare the machine learning methods to baseline methods and state-of-the-art time series models to illustrate the competitiveness of the data-driven approaches. The considered baseline methods are the seasonal naïve forecast (*S-Naïve*) and the rolling seasonal median of the previous four weeks (*S-Median*):

$$S\text{-}Naïve : \hat{y}_{t+h} = y_{t+h-m} \tag{13}$$

$$S\text{-}Median : \hat{y}_{t+h} = median(\{y_{t+h-lm} \mid l \in \{1, 2, 3, 4\}\}). \tag{14}$$

The forecast horizon is $h$, and the length of the seasonality is $m = 7$. Both methods are common in the bakery industry because they are easy to understand and cover weekly seasonality, which is a prevalent characteristic of the present time series. We employ a rolling median instead of a rolling average as it is more robust when it comes to outliers. Additionally, we also introduce the method *S-Naïve-Std*, which omits the sales on days of type SD1-SD3 and replaces them with the last observation on a regular day (SD0). Hence, the predictions of *S-Naïve-Std* for the days of types SD0 and SD4 are not distorted by sales on the previous special day.

We also evaluate the performance of the popular exponential smoothing model family. Hyndman, B., D., and Grose (2002), Hyndman, Koehler, Ord, and Snyder (2008) introduce a taxonomy for exponential smoothing models and provide an automatic approach for selecting the optimal model by considering the type of error, trend, and seasonality. For the evaluation, we rely on the training period to select the model per time series using the `ets()` function from the `forecast` package (Hyndman & Khandakar, 2008) of the R statistical software (R Core Team, 2017). An identified model is used for the complete test period, but the model coefficients are updated in a rolling-origin fashion. As exponential smoothing is a univariate forecasting method that does not consider external effects, we replace the sales on special days with the rolling seasonal median (14) as we select and fit the models.

For univariate approaches, it is necessary to make adjustments for special days because sales deviate from regular days. The adjustment strategies of the forecasts are based on the special day features introduced in Section 4.2. In particular, we consider relative adjustments (*pct*, *pct-cl*), which allow a multiplicative effect, as well as absolute adjustments (*abs*), which allow an additive effect. The strategies *pct* and *abs* are calculated

Please cite this article as: J. Huber and H. Stuckenschmidt, Daily retail demand forecasting using machine learning with emphasis on calendric special days. International Journal of Forecasting (2020), https://doi.org/10.1016/j.ijforecast.2020.02.005.

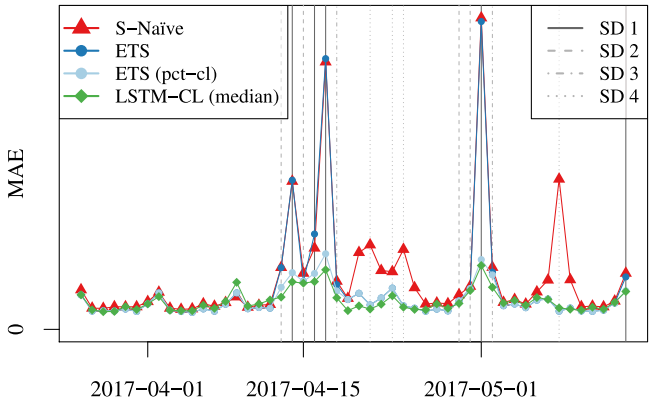


**Fig. 1.** The MAE per day from 2017-03-26 to 2017-05-14 (7 weeks). The different day types are highlighted with vertical lines. The error level is quite constant on regular days, while dramatic error peaks are observable on special days if they are not considered by the model. The scale of the *y*-axis is not provided for reasons of confidentiality.

per time series, while *pct-cl* is the average effect over all stores of the respective store type for each product category. All adjustment values are calculated in the training period by comparing the sales on the special day to the rolling median of the weekday that is to be forecast, or Sunday for public holidays. In addition to the adjustment strategies, we also evaluate the method *ETS [Sun]*, where the forecast on public holidays is replaced by the last prediction for a Sunday.

Additionally, we evaluate a multiple regression model (*LIN-REG*) using LASSO regression (Tibshirani, 1996), which is inspired by the *ADL-own* model proposed by Ma et al. (2016). This model incorporates the same information that is provided to the machine learning models, which includes log-transformed lagged sales and *S-Median*, a coupon period indicator, and binary dummy variables for the different day types and school holidays as well as the special day features. We employ cross-validation within the training period to determine the value for the regularization hyperparameter. The model *LIN-REG* is an alternative to the other baseline approaches because it does not require adjustments in a postprocessing step. The linear regression models are fitted per time series as pooling did not improve the results.

## 6. Results & discussion

In the first part of the evaluation (see Section 6.1), we analyze the single-step predictions that are most important with respect to order decision optimization. The second part is concerned with multistep forecasts (see Section 6.2) as longer planning horizons (e.g., three weeks) are also of interest for operational decisions.

### 6.1. Single-step forecasts

We focus on the performance of the second forecasting step as this is the most crucial prediction with regard to ordering decisions in an agile supply chain that is typical for bakeries. Point-of-sales data from the previous day are often only available after planning and production for the following day starts. Thus, we specify the input features of the machine learning models such that they directly predict the second step by excluding the lagged sales data of the previous day. We highlight the challenge of predicting sales on special days by elaborating on the results of the baseline methods (see Section 6.1.1) before we provide the comparison of the machine learning methods (see Sections 6.1.2 + 6.1.3).

#### 6.1.1. Special day forecasting challenge

The challenge of forecasting demand on special days is depicted in Fig. 1. While the error level is rather stable on regular days, this is not the case for special days (SD1) and their neighboring days (SD2+SD3). Hence, there is a need to design models that provide more accurate forecasts for special days.

Analysis of the results of the baseline and reference methods, presented in Figs. 2–7, leads to three central findings: First, all *ETS*-based approaches clearly outperform the simple baseline approaches for all day types. Hence, there are patterns encoded in the time series that justify the application of sophisticated prediction models. Second, the forecast errors are higher on special days (SD1-SD3) than on regular days, which suggests that special days are more difficult to forecast as the demand patterns differ. Moreover, the severity of the forecast errors on special days is hidden on aggregated key figures. By splitting days into groups, it is possible to identify the origin of the error and improve the prediction model. Third, the adjustment strategies for special days significantly improve the accuracy of all methods, which indicates the existence of underlying demand patterns for such days. This is also supported by the fact that the model *LIN-REG* has fewer errors on special days than the models with unadjusted forecasts.

With respect to the errors on different types of special days, we notice that SD1 (see Fig. 4) has the highest error, which, without adjustments, is more than 150% (300%) higher than on regular days with respect to MASE (MAE). However, the neighboring days SD2 (see Fig. 5) and SD3 (see Fig. 6) also have over 30% (MASE) higher errors than normal days. In contrast, SD0 and SD4 yield rather comparable errors, with the exception of the

Please cite this article as: J. Huber and H. Stuckenschmidt, Daily retail demand forecasting using machine learning with emphasis on calendric special days. International Journal of Forecasting (2020), https://doi.org/10.1016/j.ijforecast.2020.02.005.

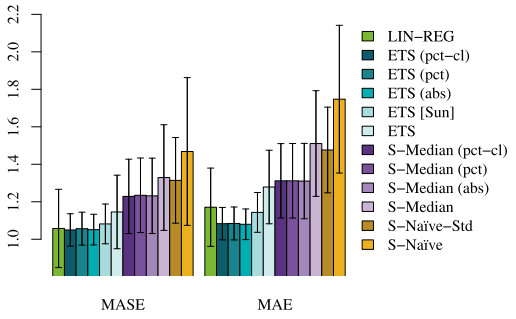


**Fig. 2.** Comparison of baseline approaches: Forecast error over all days relative to ETS on SD0 and its standard deviation.

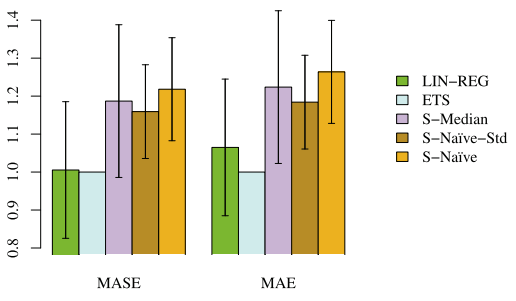


**Fig. 3.** Comparison of baseline approaches: Forecast error on days of type SD0 relative to ETS on SD0 and its standard deviation.

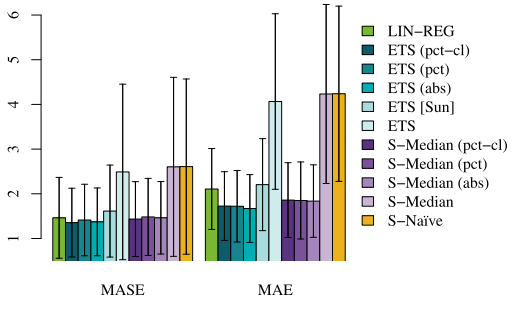


**Fig. 4.** Comparison of baseline approaches: Forecast error on days of type SD1 relative to ETS on SD0 and its standard deviation.

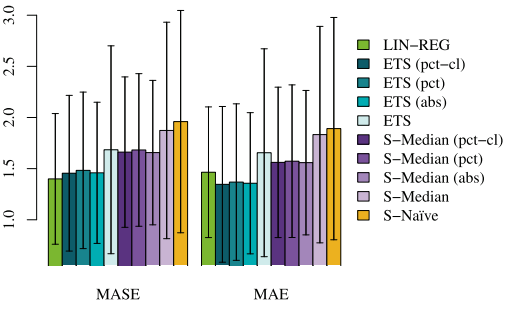


**Fig. 5.** Comparison of baseline approaches: Forecast error on days of type SD2 relative to ETS on SD0 and its standard deviation.

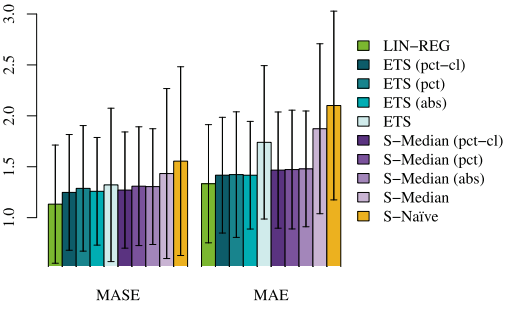


**Fig. 6.** Comparison of baseline approaches: Forecast error on days of type SD3 relative to ETS on SD0 and its standard deviation.

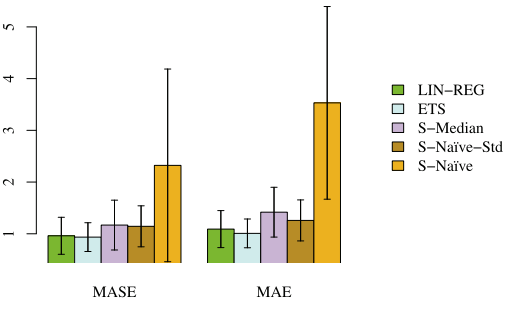


**Fig. 7.** Comparison of baseline approaches: Forecast error on days of type SD4 relative to ETS on SD0 and its standard deviation.

*S-Naïve* method, whose errors for SD4 (see Fig. 7) are closer to the error level of SD1. This supports the assumption that SD4 is suitable as a sanity check to test whether forecasts are affected by previous special days, which is possible for autoregressive models. Hence, for univariate methods, it is important to preprocess the observations on special days and make the required adjustment in a postprocessing step. To some degree, preprocessing is also beneficial for sales on days of type SD2 and SD3 as the errors of *S-Naïve-Std* for SD0 are slightly lower than the error of *S-Naïve*. *S-Median* is more robust to special days as its performance for SD4 is only slightly worse than for SD0, even though special days are not excluded from the sales history.

In general, the adjustment strategies work well for all day types and methods because the forecast error is significantly reduced. The strategies *abs* and *pct* should only yield different forecasts if the demand level changes between the training period and the test period. While *pct* leads to slightly fewer errors, the differences from *abs* are not statistically significant. This observation suggests that the demand level is rather stable for most time series. More surprisingly, the good performance of *pct-cl* seems to make it the most reliable adjustment strategy, but it is also not significantly different from the other strategies. A possible explanation for this is that the special day effect is comparable within a group of stores. Thus, computing the average effect over multiple stores makes *pct-cl* slightly more robust. For SD1, we also note that *ETS [Sun]* is much more accurate than *ETS*, which supports the

Please cite this article as: J. Huber and H. Stuckenschmidt, Daily retail demand forecasting using machine learning with emphasis on calendric special days. International Journal of Forecasting (2020), https://doi.org/10.1016/j.ijforecast.2020.02.005.

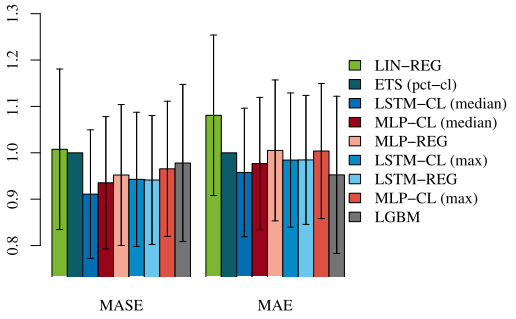


**Fig. 8.** Comparison of machine learning methods: Forecast error over all days relative to ETS (pct-cl) and its standard deviation.

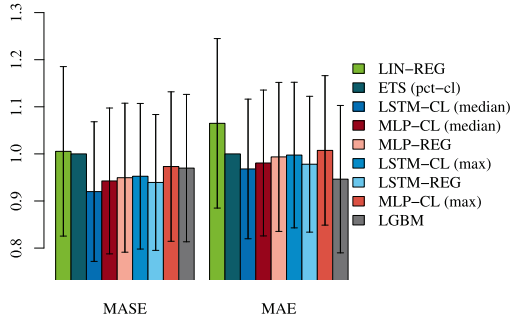


**Fig. 9.** Comparison of machine learning methods: Forecast error on days of type SD0 relative to ETS (pct-cl) and its standard deviation.

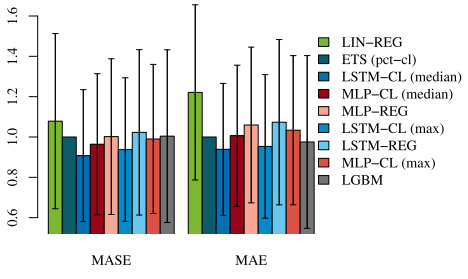


**Fig. 10.** Comparison of machine learning methods: Forecast error on days of type SD1 relative to ETS (pct-cl) and its standard deviation.

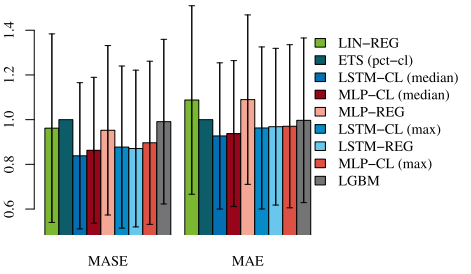


**Fig. 11.** Comparison of machine learning methods: Forecast error on days of type SD2 relative to ETS (pct-cl) and its standard deviation.

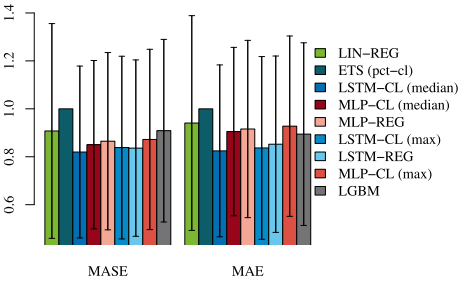


**Fig. 12.** Comparison of machine learning methods: Forecast error on days of type SD3 relative to ETS (pct-cl) and its standard deviation.

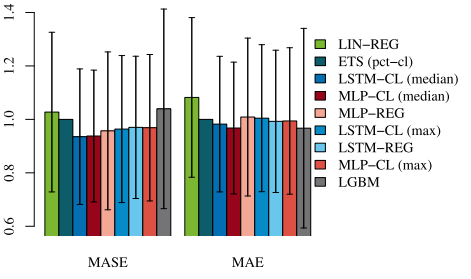


**Fig. 13.** Comparison of machine learning methods: Forecast error on days of type SD4 relative to ETS (pct-cl) and its standard deviation.

assumption that public holidays have much in common with Sundays.

However, while the adjustments significantly reduce the errors for SD1 and SD2, this only holds to a limited degree for SD3. While the error levels of SD3 are comparable with those of SD2, there are no significant differences among all *ETS*-based approaches and *S-Median*-based approaches that rely on adjustments. However, the scale-dependent errors of unadjusted forecasts are still higher than the errors of adjusted forecasts. Thus, adjustment strategies still provide practically relevant improvements for those days, while the relative benefits are reduced compared to those of SD1.

The method *LIN-REG* does not rely on adjustments as the special days are explicitly incorporated, which works well for the neighboring days of public holidays but is worse on the main special days (SD1). On days of types SD0 and SD4, the differences between *LIN-REG* and *ETS*-based models are mostly noticeable from the error measure MAE, while differences with respect to MASE are smaller. Nevertheless, special day patterns can be learned by a linear model, but relying on adjustments in a post-processing step seems to be the better option in some cases. *ETS* is a suitable model for regular demand patterns that follow a very strong weekly seasonality, while other external influences, with the exception of special days, are rather negligible or hard to separate from the noise of daily data. Hence, it is reasonable that *LIN-REG* struggles to outperform *ETS* (with adjustments) even though it incorporates more information.

Overall, the relative advantage of *ETS*-based models compared to baseline methods slightly diminishes for special days as all methods depend heavily on the adjustments, which are identical for all methods. More fine-grained adjustments might be necessary in order to

Please cite this article as: J. Huber and H. Stuckenschmidt, Daily retail demand forecasting using machine learning with emphasis on calendric special days. International Journal of Forecasting (2020), https://doi.org/10.1016/j.ijforecast.2020.02.005.

further reduce forecast errors. As such adjustment rules and values are hard to specify manually, we rely on machine learning methods that are able to learn them from data.

#### 6.1.2. Baseline vs. Machine learning methods

We compare the performance of the machine learning methods to the best reference methods; i.e., *LIN-REG* and *ETS (pct-cl)* (see Figs. 8–13). Overall, the machine learning methods significantly outperform the reference methods by a noticeable error margin. The performance gains are the largest for the neighboring days of public holidays (SD2+SD3), where the error drops by more than 10%. Those are the days where the adjustment strategies did not provide much benefit compared to unadjusted forecasts, and the performance of *LIN-REG* suggests that improvements are possible. One explanation for this is that the neighboring days of public holidays share similarities among the different public holidays. Hence, machine learning methods are able to detect and learn more precise patterns that translate to lower forecast errors. Due to those improvements, the forecast errors for SD2 and SD3 are noticeably closer to the performance on regular days. For SD1, the differences are not as significant. One possible explanation for this is that the demand on special days is highly volatile as the weekday or the time of year varies over the years. This makes it generally hard to estimate the demand for such days. Nevertheless, machine learning approaches are still the preferred option because they provide more accurate predictions.

#### 6.1.3. Regression vs. Classification

A comparison of the machine learning methods (see Fig. 8) reveals that the methods based on ANNs outperform *LGBM* with respect to MASE. However, *LGBM* is, especially for regular days (SD0), the most accurate method with respect to MAE. The reason for this is that *LGBM* provides more accurate predictions for larger values. The preprocessing of the target variables of the ANNs might negatively affect the forecast accuracy for larger target values. For regression-based approaches, we perform a log transformation followed by linearly scaling the target values to the range $[-0.5, 0.5]$. For classification-based approaches, we perform binning, which benefits higher absolute errors as the bins comprise larger intervals as the target values grow. However, preliminary experiments with other transformations led to worse results. In contrast, transforming the target values for the *LGBM* models did not lead to performance gains. *LGBM* implicitly creates data bins during the learning phase while constructing the decision trees, which apparently leads to a better grouping of the target values.

With respect to the ANN-based methods, we notice that recurrent ANNs (LSTM) outperform their feed-forward (MLP) counterparts. Hence, it seems beneficial to process the time series data in a sequential fashion. Moreover, transforming the regression problem to a classification problem also seems to be beneficial, even though the range and the number of possible target values is limited by the number of classes and their underlying values. However, this does not seem to be a problem in the present use case because the classification approaches are at least as good as their regression-based counterparts.

The prediction of a classification model is a probability distribution over the target classes that can be interpreted as a density forecast. A survey on density and probabilistic forecasting is provided by Gneiting and Katzfuss (2014) and Tay and Wallis (2000). To select the predicted values of the classification, we investigate two approaches: By selecting the class with the highest probability (*CL (max)*), we choose the mode of the distribution. However, it is also possible to choose the median (*CL (median)*) at no additional cost, which might be better suited for the performance measures used. Generally, we observe that the accuracy of *CL (max)* is largely comparable to or only slightly better than the accuracy of the standard regression approach. Taking the median of a probability distribution of the target classes leads to significantly better results. We observe the largest performance gain for feed-forward ANNs as the recurrent ANNs are already at a high level of accuracy. Both methods that employ *CL (median)* are generally more accurate than any other method. One explanation for this is that predicting and exploiting the probability distribution over target classes provides an additional level to address the uncertainty associated with a prediction. Gneiting (2011) and Kolassa (2016, 2020) highlight that different functionals optimize different loss functions; e.g., the mean squared error (mean absolute error) is optimized by the mean (median) of the probability distribution, which leads to vastly different outcomes for asymmetric distributions.

In fact, in more than 50% of the cases, the predictions between *CL (median)* and *CL (max)* are different (see Table 10). For nearly 69% of the forecasts, the selected classes are neighboring classes. The relative change of the predicted values is only 0.05% for approximately 50% of the forecasts. The direction of the changes only slightly favors an increase in the forecasts using *CL (median)*. Hence, we conclude that selecting a different class does not serve as bias correction. The comparison also reveals that *CL (median)* is, for roughly 55% of the forecasts, more accurate than *CL (max)*, which translates to a reduction in the forecast error of 6% to 9% (see Table 11). The previous statements hold for *MLP-CL* and *LSTM-CL*.

#### 6.1.4. Effect of retraining

In the forecasting literature, it is common practice to perform a rolling-origin evaluation as this typically leads to better forecasts. Rolling-origin evaluation means that the model coefficients are refitted for every forecast origin. In this study, we only refitted the *ETS* models and *LIN-REG* for each of the 142,326 forecasts (see Table 9). This is already challenging in a productive setting as the time frame for calculating and delivering daily forecasts is limited to only a few hours. For the evaluated machine learning methods, it is certainly not possible to train them on a daily basis. However, we provide empirical support that machine learning models that are not retrained during the test period outperform the reference methods (see Section 6.1.2). Applying the trained models is quite efficient, which makes them suitable for productive usage. Nevertheless, we also investigate whether more frequent

Please cite this article as: J. Huber and H. Stuckenschmidt, Daily retail demand forecasting using machine learning with emphasis on calendric special days. International Journal of Forecasting (2020), https://doi.org/10.1016/j.ijforecast.2020.02.005.

**Table 10**
Comparison of *CL (median)* and *CL (max)*. The table shows the direction of the change and the degree of change with respect to the target class and the values relative to *CL (max)* if the predictions of both approaches do not match. We provide the quantiles for the last two key figures.

| Method | Direction | | | Class steps | | | | | Relative change | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | < | == | > | 0 | 25 | 50 | 75 | 100 | 0 | 25 | 50 | 75 | 100 |
| FNN-CL | 0.24 | 0.46 | 0.31 | −12 | −1 | −1 | 1 | 14 | −0.49 | −0.05 | 0.05 | 0.12 | 10 |
| LSTM-CL | 0.25 | 0.47 | 0.28 | −14 | −1 | 1 | 1 | 17 | −0.42 | −0.05 | 0.05 | 0.12 | 9 |

training leads to improved results since the test period is relatively long and more regular retraining is possible. Having an additional week of data available for training only increases the dataset by 0.98%, which accumulates to approximately 20% over the 150 days of the test period.

Thus, we retrain the machine learning models biweekly from scratch; i.e., with random initial weights and no trees, with the same hyperparameters as before. The only exceptions are the *LSTM* models, whose weights for the previously trained models are only fine-tuned with the extended datasets to save training time. The drawback of this approach is that it is more likely to overfit the "old" training data. The trained models are then used for a limited number of upcoming weeks before they are replaced with refitted models. We report the accuracy of various fitting frequencies in Table 12.

In general, we observe that biweekly retraining leads to the best results, while no retraining - i.e., models being used for 22 weeks - leads to the highest forecast error. Due to biweekly retraining, the error drops by 2%–3% compared with when there is no retraining, depending on the error measure. The relative improvements are larger for MAE than MASE. Retraining frequencies between 10 and 4 weeks produce differences that are not very (statistically) significant, and the additional performance gains due to biweekly retraining are only approximately one percentage point. We also notice that the ANNs tend to benefit more from retraining than *LGBM*, even though they already outperformed *LGBM* with no retraining. The relative error decrease of *LGBM* is not larger than 1.7% for any retraining frequency. Thus, ANNs have the capability to adapt and incorporate additional observations, which helps to reduce the larger absolute forecast error. One reason for this is that the number of available observations decreases as the demand grows, which makes additional data valuable.

To summarize, despite the fact that the models are trained on a very large dataset, forecast accuracy improvements are possible if the models are more frequently retrained. Hence, in a productive setting, how often it is feasible to retrain the models must be evaluated. There are different opportunities to limit computational costs. As we rely on an ensemble of 50 models, it is possible to continuously replace a subset of the models. It is also possible to only fine-tune the weights of the ANNs that are only trained on reasonably old data. However, even if retraining is not workable in a productive setting, the machine learning methods are still highly competitive over a long time period, as shown in Section 6.1.2.

### 6.2. Multistep forecasts

The previous part of the evaluation was concerned with single-step predictions. However, longer planning horizons can also be a requirement for certain operational decisions. For example, public holidays are partially planned in advance. Multistep predictions also allow us to check whether the full demand patterns of special days are actually learned since multiple special days fall within the forecast horizon. We compute forecasts for 21 days (3 weeks) to evaluate whether certain methods perform better for shorter and longer horizons. We exclude the predictions for the first 20 days of the test period to have 21 predictions for each observation, which enables the comparison of the different forecasting steps. The resulting test set comprises 122,730 observations per step. For the machine learning methods, we reuse the trained models from the experiments in Section 6.1.2 and apply them iteratively. The results from the evaluation with respect to MASE are presented in Fig. 14.

**Table 11**
Accuracy of the classification approaches when *CL (max)* and *CL (median)* provide different results. *CL (median)* has a lower forecast error and a lower average rank compared to *CL (max)*. We also provide the average rank and, in brackets, its standard deviation.

| Method | MASE | MAE | Rank |
|---|---|---|---|
| MLP-CL (max) | 1.00 | 1.00 | 1.55 (0.49) |
| MLP-CL (median) | **0.94** | **0.93** | **1.45 (0.49)** |
| LSTM-CL (max) | 1.00 | 1.00 | 1.56 (0.49) |
| LSTM-CL (median) | **0.94** | **0.93** | **1.44 (0.49)** |

We observe that the ordering of the methods is stable for the different forecasting steps. The only exceptions are *ETS*-based approaches, whose errors increase more quickly than the errors of the machine learning methods. *ETS* is partially competitive for the first three steps, but the performance gap with the machine learning approaches increases with the forecast horizon. For the machine learning methods, we also notice that the results are somewhat stable within a season (i.e., a week), which matches the strong weekly seasonality of the time series. A stable forecast error within a season is observable not only for regular days but also for special days. Hence, it can be assumed that the special day patterns are correctly represented by the models, which makes postprocessing obsolete. Finally, the conclusions drawn in Section 6.1.2 are valid not only for single-step predictions but also for the other forecasting steps. This makes it reasonable to initially optimize a model for single-step predictions before options for multistep forecasts are explored.

## 7. Conclusion

We presented a large-scale demand forecasting scenario in the retail domain that requires daily forecasts at the store level. The company operates over one hundred

Please cite this article as: J. Huber and H. Stuckenschmidt, Daily retail demand forecasting using machine learning with emphasis on calendric special days. International Journal of Forecasting (2020), https://doi.org/10.1016/j.ijforecast.2020.02.005.

**Table 12**
The effect of model refitting during the test phase for different frequencies. The relative error compared to no retraining; i.e., refitting every 22 weeks, is provided for each method. For each method, we underline the lowest error and mark frequencies that are not significantly different at the 0.05 significance level according to the Wilcoxon signed-rank test in bold.

| Measure | Model | Model retraining frequency in weeks | | | | | |
|---|---|---|---|---|---|---|---|
| | | 22 | 10 | 8 | 6 | 4 | 2 |
| MAE | LGBM | 1.000 | 0.995 | 0.996 | 0.988 | 0.993 | **0.983** |
| | LSTM-CL (max) | 1.000 | 0.991 | 0.988 | 0.983 | 0.978 | **0.967** |
| | LSTM-CL (median) | 1.000 | 0.991 | 0.987 | 0.986 | 0.981 | **0.969** |
| | LSTM-REG | 1.000 | 0.988 | 0.996 | 0.981 | 0.986 | **0.965** |
| | MLP-CL (max) | 1.000 | 0.986 | 0.992 | 0.991 | 0.992 | **0.971** |
| | MLP-CL (median) | 1.000 | 0.987 | 0.993 | 0.995 | 0.994 | **0.972** |
| | MLP-REG | 1.000 | 0.989 | 1.001 | 0.994 | 1.001 | **0.983** |
| MASE | LGBM | 1.000 | 0.998 | 1.002 | 0.995 | 0.999 | **0.993** |
| | LSTM-CL (max) | 1.000 | 0.993 | 0.993 | 0.991 | 0.987 | **0.980** |
| | LSTM-CL (median) | 1.000 | 0.993 | 0.992 | 0.990 | 0.988 | **0.978** |
| | LSTM-REG | 1.000 | 0.991 | 0.992 | 0.986 | 0.985 | **0.971** |
| | MLP-CL (max) | 1.000 | 0.988 | 0.991 | 0.990 | 0.988 | **0.976** |
| | MLP-CL (median) | 1.000 | 0.986 | 0.990 | 0.991 | 0.988 | **0.973** |
| | MLP-REG | 1.000 | 0.990 | 0.994 | 0.993 | 0.994 | **0.983** |

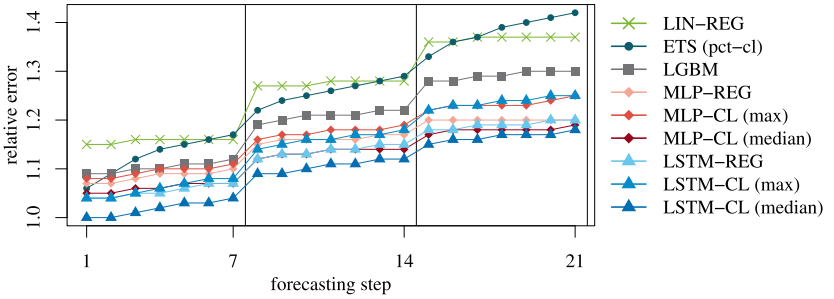


**Fig. 14.** The multistep forecast error with respect to MASE for the next 21 days. The provided error is relative to the best-performing method at step 1.

stores, and sales data for almost three years are available at the store-product level. The sales history can be enhanced with external information such as calendar events or the local environment of each store. This means that a large data pool is available for exploitation by a prediction method. Hence, the characteristics of this application scenario are potentially well suited for the application of machine learning methods for time series forecasting. We discuss the possibilities for applying machine learning methods and perform an extensive empirical evaluation on a real-world dataset. A challenge in this use case is predicting the demand on special days (e.g., public holidays), because such days are subject to vastly different patterns, which lead to high forecast errors using standard approaches. This issue concerns not only public holidays themselves but also their neighboring days.

Our empirical evaluation provides evidence that machine learning methods are indeed a viable alternative to established approaches. First, they achieve a higher forecast accuracy. In particular, for special days, the error can be reduced by more than 10% and up to 20% compared to time series models with adjustments or a regularized linear regression model. The machine learning models incorporate special day-specific features, which makes preprocessing of the time series data and adjustments in a postprocessing step obsolete. Hence, manually defining adjustment rules for special days is also not required. Second, we are able to train global machine learning models that forecast every time series in the dataset and still provide accurate predictions. Therefore, only a limited number of models must be built. Third, while retraining the models during the evaluation improved the accuracy, the performance gains are only between 1% and 3% compared to no retraining during the five-month evaluation period. This suggests that retraining the models is an opportunity to slightly reduce the forecast error rather than a requirement for machine learning methods. Fourth, for longer forecast horizons (up to 21 days), the forecast error is stable within a season (i.e., a week), and the ranking of the evaluated methods is constant over different horizons.

With respect to the evaluated methods, we notice that RNNs (LSTMs) outperform feed-forward ANNs (MLPs) and GBRTs. Moreover, for both types of ANNs, it is beneficial to model the learning problem as a classification task rather than a regression problem. Transforming the forecasting problem into a classification task provides an additional level to address the uncertainty of a prediction, as the prediction - i.e., a probability distribution over the target classes - can be interpreted as a density forecast. Selecting the median of the distribution instead of the class that has the highest probability is particularly recommended; i.e., the mode of the distribution.

In this study, we only evaluated standard versions of state-of-the-art machine learning methods. We also did not employ sophisticated algorithms to optimize the hyperparameters of the methods or select only the most

relevant features. However, the results of our evaluation indicate that a more elaborate model-building and selection process is feasible in a productive setting since the evaluated models are competitive over a long application phase and the ranking is also stable for different forecasting horizons. Thus, it is reasonable to rely on cross-validation with folds and to initially focus on a single prediction step before other horizons are explored.

This research can be extended in multiple directions. With regard to the applied machine learning methods, it would be interesting to investigate whether an automated model-building process leads to better prediction models. We also noticed that various model types provide different forecasts. Thus, the evaluation of ensembles of different model types or stacking should increase accuracy. In our study, the classification approaches had the highest accuracy even though we applied a simple approach to create the target classes. Hence, other approaches for binning the target values should be explored and evaluated. Moreover, it can be discussed whether the classification approach is a suitable method for density forecasts. With respect to the retail domain, it is also necessary to evaluate how machine learning methods perform on the store-article level for daily forecasts. The data are noisier, and new challenges arise due to a changing assortment (e.g., new products) at the store-article level. Another research opportunity is to conduct a feature importance analysis of the trained models to gain domain-specific insights.

## Acknowledgments

This research was supported by *OPAL - Operational Analytics GmbH* (https://www.opal-analytics.com).

## Appendix A. Supplementary data

Supplementary result tables and additional graphs related to this article can be found online at https://doi.org/10.1016/j.ijforecast.2020.02.005.

## References

Aburto, L., & Weber, R. (2007). Improved supply chain management based on hybrid demand forecasts. *Applied Soft Computing*, *7*(1), 136–144.

Adya, M., & Collopy, F. (1998). How effective are neural networks at forecasting and prediction? A review and evaluation. *Journal of Forecasting*, *17*(5–6), 481–495.

Ahmed, N. K., Atiya, A. F., Gayar, N. E., & El-Shishiny, H. (2010). An empirical comparison of machine learning models for time series forecasting. *Econometric Reviews*, *29*(5–6), 594–621.

Alon, I., Qi, M., & Sadowski, R. J. (2001). Forecasting aggregate retail sales: a comparison of artificial neural networks and traditional methods. *Journal of Retailing and Consumer Services*, *8*(3), 147–156.

Arunraj, N. S., & Ahrens, D. (2015). A hybrid seasonal autoregressive integrated moving average and quantile regression for daily food sales forecasting. *International Journal of Production Economics*, *170*, 321–335.

Barrow, D. K., & Crone, S. F. (2016). Cross-validation aggregation for combining autoregressive neural network forecasts. *International Journal of Forecasting*, *32*(4), 1120–1137.

Barrow, D., Crone, S., & Kourentzes, N. (2010). An evaluation of neural network ensembles and model selection for time series prediction. In *The 2010 international joint conference on neural networks* (pp. 1–8).

Barrow, D., & Kourentzes, N. (2018). The impact of special days in call arrivals forecasting: A neural network approach to modelling special days. *European Journal of Operational Research*, *264*, 967–977.

Ben Taieb, S., Bontempi, G., Atiya, A. F., & Sorjamaa, A. (2012). A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. *Expert Systems with Applications*, *39*(8), 7067–7083.

Bergmeir, C., Hyndman, R. J., & Koo, B. (2018). A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics & Data Analysis*, *120*, 70–83.

Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. In *Proceedings of the 24th international conference on neural information processing systems* (pp. 2546–2554). Curran Associates Inc.

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research (JMLR)*, *13*, 281–305.

Bergstra, J., Yamins, D., & Cox, D. D. (2013). Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *Proceedings of the 30th international conference on machine learning - vol. 28* (pp. 115–123). JMLR.org.

Bischl, B., Kerschke, P., Kotthoff, L., Lindauer, M., Malitsky, Y., Fréchette, A., et al. (2016). Aslib: A benchmark library for algorithm selection. *Artificial Intelligence*, *237*, 41–58.

Bontempi, G., Ben Taieb, S., & Borgne, Y.-A. L. (2012). Machine learning strategies for time series forecasting. In *Lecture notes in business information processing*, *Business intelligence* (pp. 62–77). Berlin, Heidelberg: Springer.

Cancelo, J. R., Espasa, A., & Grafe, R. (2008). Forecasting the electricity load from one day to one week ahead for the spanish system operator. *International Journal of Forecasting*, *24*, 588–602.

Carbonneau, R., Laframboise, K., & Vahidov, R. (2008). Application of machine learning techniques for supply chain demand forecasting. *European Journal of Operational Research*, *184*(3), 1140–1154.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. (pp. 785–794). arXiv:1603.02754 [cs].

Cheng, J., Wang, Z., & Pollastri, G. (2008). A neural network approach to ordinal regression. In *IEEE international joint conference on neural networks 2008* (pp. 1279–1284).

Chu, C.-W., & Zhang, G. P. (2003). A comparative study of linear and nonlinear models for aggregate retail sales forecasting. *International Journal of Production Economics*, *86*(3), 217–231.

Cooper, L. G., Baron, P., Levy, W., Swisher, M., & Gogos, P. (1999). PromoCast™: A new forecasting method for promotion planning. *Marketing Science*, *18*(3), 301–316.

Crone, S. F., Hibon, M., & Nikolopoulos, K. (2011). Advances in forecasting with neural networks? Empirical evidence from the NN3 competition on time series prediction. *International Journal of Forecasting*, *27*(3), 635–660.

Crone, S. F., & Kourentzes, N. (2009). Forecasting seasonal time series with multilayer perceptrons-an empirical evaluation of input vector specifications for deterministic seasonality.. In *Proceedings of the 2009 international conference on data mining* (pp. 232–238).

Di Pillo, G., Latorre, V., Lucidi, S., & Procacci, E. (2016). An application of support vector machines to sales forecasting under promotions. *4OR*, *14*(3), 309–325.

Divakar, S., Ratchford, B. T., & Shankar, V. (2005). CHAN4CAST: A multichannel, multiregion sales forecasting model and decision support system for consumer packaged goods. *Marketing Science*, *24*(3), 334–350.

Doganis, P., Alexandridis, A., Patrinos, P., & Sarimveis, H. (2006). Time series sales forecasting for short shelf-life food products based on artificial neural networks and evolutionary computing. *Journal of Food Engineering*, *75*(2), 196–204.

Ehrenthal, J. C., & Stölzle, W. (2013). An examination of the causes for retail stockouts. *International Journal of Physical Distribution and Logistics Management*, *43*(1), 54–69.

Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., & Hutter, F. (2015). Efficient and robust automated machine learning. In *Advances in neural information processing systems: Vol. 28*, (pp. 2962–2970). Curran Associates, Inc.

Fildes, R., Ma, S., & Kolassa, S. (2019). Retail forecasting: Research and practice. *International Journal of Forecasting*.

Gneiting, T. (2011). Making and evaluating point forecasts. *Journal of the American Statistical Association*, *106*(494), 746–762.

Gneiting, T., & Katzfuss, M. (2014). Probabilistic forecasting. *Annual Review of Statistics and its Application*, *1*(1), 125–151.
Gür Ali, Ö., Sayın, S., Van Woensel, T., & Fransoo, J. (2009). SKU Demand forecasting in the presence of promotions. *Expert Systems with Applications*, *36*(10), 12340–12348.
Gutiérrez, P. A., Pérez-Ortiz, M., Sánchez-Monedero, J., Fernández-Navarro, F., & Hervás-Martínez, C. (2016). Ordinal regression methods: Survey and experimental study. *IEEE Transactions on Knowledge and Data Engineering*, *28*(1), 127–146.
Hastie, T., Tibshirani, R., & Friedman, J. (2009). *Springer series in statistics*, *The elements of statistical learning*. New York, NY, USA: Springer.
Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.
Hofmann, E., & Rutschmann, E. (2018). Big data analytics and demand forecasting in supply chains: a conceptual analysis. *The International Journal of Logistics Management*, *29*(2), 739–766.
Huang, T., Fildes, R., & Soopramanien, D. (2014). The value of competitive information in forecasting FMCG retail product sales and the variable selection problem. *European Journal of Operational Research*, *237*(2), 738–748.
Huber, J., Gossmann, A., & Stuckenschmidt, H. (2017). Cluster-based hierarchical demand forecasting for perishable goods. *Expert Systems with Applications*, *76*, 140–151.
Hyndman, R. J., B., K. A., D., S. R., & Grose, S. (2002). A state space framework for automatic forecasting using exponential smoothing methods. *International Journal of Forecasting*, *18*(3), 439–454.
Hyndman, R. J., & Khandakar, Y. (2008). Automatic time series forecasting: the forecast package for R. *Journal of Statistical Software*, *26*(3), 1–22.
Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International journal of forecasting*, *22*(4), 679–688.
Hyndman, R. J., Koehler, A. B., Ord, J. K., & Snyder, R. D. (2008). *Forecasting with exponential smoothing: the state space approach*. Springer Science & Business Media.
Kang, Y., Hyndman, R. J., & Smith-Miles, K. (2017). Visualising forecasting algorithm performance using time series instance spaces. *International Journal of Forecasting*, *33*(2), 345–358.
Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., et al. (2017). LightGBM: A highly efficient gradient boosting decision tree. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), *Advances in neural information processing systems, vol. 30* (pp. 3146–3154). Curran Associates, Inc.
Kim, M. S. (2013). Modeling special-day effects for forecasting intraday electricity demand. *European Journal of Operational Research*, *230*, 170–180.
Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic optimization. In *International conference on learning representations 2015*.
Kolassa, S. (2016). Evaluating predictive count data distributions in retail sales forecasting. *International Journal of Forecasting*, *32*(3), 788–803.
Kolassa, S. (2020). Why the "best" point forecast depends on the error or accuracy measure. In *M4 Competition*: *International Journal of Forecasting*, In *M4 Competition*: *36*(1).208–211,
Kourentzes, N., Barrow, D. K., & Crone, S. F. (2014). Neural network ensemble operators for time series forecasting. *Expert Systems with Applications*, *41*(9), 4235–4244.
Kourentzes, N., & Petropoulos, F. (2016). Forecasting with multivariate temporal aggregation: The case of promotional modelling. *International Journal of Production Economics*, *181*, 145–153.
LeCun, Y. A., Bottou, L., Orr, G. B., & Müller, K.-R. (2012). Efficient BackProp. In *Lecture notes in computer science*, *Neural networks: Tricks of the trade* (pp. 9–48). Springer, Berlin, Heidelberg.
Ma, S., & Fildes, R. (2017). A retail store SKU promotions optimization model for category multi-period profit maximization. *European Journal of Operational Research*, *260*(2), 680–692.
Ma, S., Fildes, R., & Huang, T. (2016). Demand forecasting with high dimensional data: The case of SKU retail sales forecasting with intra- and inter-category promotional information. *European Journal of Operational Research*, *249*(1), 245–257.
Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018a). The M4 competition: Results, findings, conclusion and way forward. *International Journal of Forecasting*, *34*(4), 802–808.
Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018b). Statistical and machine learning forecasting methods: Concerns and ways forward. *PLOS One*, *13*(3), 1–26.
Panapakidis, I. P. (2016). Application of hybrid computational intelligence models in short-term bus load forecasting. *Expert Systems with Applications*, *54*, 105–120.
Petropoulos, F., Makridakis, S., Assimakopoulos, V., & Nikolopoulos, K. (2014). 'Horses for courses' in demand forecasting. *European Journal of Operational Research*, *237*(1), 152–163.
R Core Team (2017). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing, URL https://www.R-project.org/.
Ramanathan, U., & Muyldermans, L. (2010). Identifying demand factors for promotional planning and forecasting: A case of a soft drink company in the UK. In *Supply Chain Forecasting Systems*: *International Journal of Production Economics*, In *Supply Chain Forecasting Systems*: *128*(2).538–545,
Ramanathan, U., & Muyldermans, L. (2011). Identifying the underlying structure of demand during promotions: A structural equation modelling approach. *Expert Systems with Applications*, *38*(5), 5544–5552.
Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*: *Vol. 25*, (pp. 2951–2959). Curran Associates, Inc.
Soares, L. J., & Medeiros, M. C. (2008). Modeling and forecasting short-term electricity load: a comparison of methods with an application to brazilian data. *International Journal of Forecasting*, *24*, 630–644.
Srinivasan, D., Chang, C. S., & Liew, A. C. (1995). Demand forecasting using fuzzy neural computation, with special emphasis on weekend and public holiday forecasting. *IEEE Transactions on Power Systems*, *10*, 1897–1903.
Tay, A. S., & Wallis, K. F. (2000). Density forecasting: a survey. *Journal of Forecasting*, *19*(4), 235–254.
Taylor, J. W. (2007). Forecasting daily supermarket sales using exponentially weighted quantile regression. *European Journal of Operational Research*, *178*(1), 154–167.
Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, 267–288.
Trapero, J. R., Kourentzes, N., & Fildes, R. (2015). On the identification of sales forecasting models in the presence of promotions. *The Journal of the Operational Research Society*, *66*(2), 299–307.
Trapero, J. R., Pedregal, D. J., Fildes, R., & Kourentzes, N. (2013). Analysis of judgmental adjustments in the presence of promotions. *International Journal of Forecasting*, *29*(2), 234–243.
Van Donselaar, K. H., Gaur, V., Van Woensel, T., Broekmeulen, R. A., & Fransoo, J. C. (2010). Ordering behavior in retail stores and implications for automated replenishment. *Management Science*, *56*(5), 766–784.
Van Donselaar, K. H., Peters, J., de Jong, A., & Broekmeulen, R. A. C. M. (2016). Analysis and forecasting of demand during promotions for perishable items. *International Journal of Production Economics*, *172*, 65–75.
Van Donselaar, K., van Woensel, T., Broekmeulen, R., & Fransoo, J. (2006). Inventory control of perishables in supermarkets. *International Journal of Production Economics*, *104*(2), 462–472.
van Heerde, H. J., Leeflang, P. S., & Wittink, D. R. (2000). The estimation of pre- and postpromotion dips with store-level scanner data. *Journal of Marketing Research*, *37*(3), 383–395.
van Heerde, H. J., Leeflang, P. S. H., & Wittink, D. R. (2002). How promotions work: SCAN*PRO-based evolutionary model building. *Schmalenbach Business Review*, *54*(3), 198–220.
Van Woensel, T., Van Donselaar, K., Broekmeulen, R., & Fransoo, J. (2007). Consumer responses to shelf out-of-stocks of perishable products. *International Journal of Physical Distribution and Logistics Management*, *37*(9), 704–718.
Wang, A. J., & Ramsay, B. (1998). A neural network based estimator for electricity spot-pricing with particular reference to weekend and public holidays. *Neurocomputing*, *23*(1).
Zhang, G., Patuwo, B. E., & Hu, M. Y. (1998). Forecasting with artificial neural networks:: The state of the art. *International Journal of Forecasting*, *14*(1), 35–62.

Please cite this article as: J. Huber and H. Stuckenschmidt, Daily retail demand forecasting using machine learning with emphasis on calendric special days. International Journal of Forecasting (2020), https://doi.org/10.1016/j.ijforecast.2020.02.005.