IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 17, 2020, accepted July 21, 2020, date of publication July 27, 2020, date of current version August 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3012032

Cold-Start Promotional Sales Forecasting Through Gradient Boosted-Based Contrastive Explanations

CARLOS AGUILAR-PALACIOS [1], SERGIO MUÑOZ-ROMERO [1,2], AND JOSÉ LUIS ROJO-ÁLVAREZ [1,2], (Senior Member, IEEE)

[1]Department of Signal Theory and Communications, Telematics, and Computing Systems, Universidad Rey Juan Carlos, 28943 Madrid, Spain
[2]Center for Computational Simulation, Universidad Politécnica de Madrid, 28040 Madrid, Spain

Corresponding author: Carlos Aguilar-Palacios (carlos.aguilar.palacios@gmail.com)

This work was supported in part by the Spanish Government through the Research under Grant FINALE TEC2016-75161-C2-1-R and Grant KERMES TEC2016-81900-REDT/AEI.

ABSTRACT Multiple Machine Learning solutions in Industry exist where interpretability is required. In retail, this is especially important when dealing with cold-start forecasting of promotional sales. In the planning phase of these promotions, retailers produce sales predictions that are scrutinised by both forecasting experts and managers. In this paper, we combine the predictive benefits of Gradient Boosted Decision Trees regressors and the interpretability of contrastive explanations. These are implicitly generated by the manner we shape data. Our method presents the cold-start forecasts in relation to the observed promotional sales of other products, which we call *neighbours*. They are selected based on a measure of closeness to the predicted promotion, which is derived from the variable importance calculated during the training of the regressors. With this information, the expert reviewer adjusts the cold-start prediction by simply varying the contribution of the neighbours. To validate our results we test our method on a surrogate model, as well as on real-market data. The results on the surrogate model demonstrate that our method is able to accurately identify the features that contribute to sales and then select the closest neighbours to produce a contrastive explanation. The results on real-market data also show that the proposed method performs at a similar level to widespread methods such as conventional *CatBoost*, *NGBoost* or *AutoGluon*, and has the advantage of providing interpretability.

INDEX TERMS Cold-start forecasting, contrastive explanations, interpretable ML (iML), retail promotions, supply chain.

I. INTRODUCTION

Sales promotions are strategies used by retailers to incentivise the market demand of products. The most common form of promotions include price discounts, special displays, freebies, deals (two for the price of one), vouchers and rebates [1]. Apart from being profitable, promotions also help to reduce stockpiling, attract customers, build store traffic, introduce new products, and counteract competitors [2]–[4]. The process of planning promotions is complex and challenging [5] and the estimation of sales can benefit the entire supply chain, including both manufacturers and suppliers [6], as well as it can help reducing waste.

Promotional forecasting is a difficult task and the lack of historical sales in the cold-start problem makes it even more challenging. Machine Learning approaches have been used to tackle this problem, especially in fields such as recommender systems, energy forecast, and online retailers [7], [8]. With an increasing number of automated decisions originating from Machine Learning (ML) systems, certain applications exist where trusting the decisions is challenging. Predictions alone and metrics calculated on these predictions do not suffice to characterise a model [9]. Understandably, a participant in the supply chain of a retailer might feel uneasy placing an order on a new product with very little information about it. This situation gets more complicated when the decision on the number of units originates from an automated ML system with no supporting feedback on the figures, the so called ''black-box'' models. This situation of mistrust can be remediated by adding information on how the decision has been made. Ideally some level of cooperation should exist between the human and the ML system where the user can control and modify the results.

The associate editor coordinating the review of this manuscript and approving it for publication was Tu Ngoc Nguyen.

Amongst all types of promotions, there is a particular subset known as cold-start promotions where the stock keeping unit (SKU) has not been on promotion before. The main reasons are typically the inclusion of a new product in the catalog, a new variety of an existing product, or a regular product that has just never been on promotion.

This paper focuses on adding *interpretability* to the predictions produced by a ML system in grocery retail. The goal of interpretability is to describe a system in a way that is understandable to humans [10], and fulfilling three pillars: trust, interaction and transparency [11]. As such, we propose a new method able to forecast cold-start promotions while providing interpretability through *contrastive explanations*. For a philosophical approach to the definitions of contrastive explanations we recommend the works by Lipton [12], [13]. In the context of ML, Miller [14] distinguishes contrastive explanations according to the type of question that they answer: "Why P rather than Q?" or "Why P but Q". The former is the *alternative* question and the latter is the *congruent* one, which is the one that we focus on in this paper. A congruent question asks about the fact P occurs in the current situation, while some surrogate Q actually occurred in some other situation. *Interpretability* should not be confused with *explainability*, which focuses on the explanation of the internals of a model to a human. *Explainability* has been the subject of many recent works in the research community [15]–[19] aiming to clarify and facilitate the understanding of ML and Deep Learning models.

The novelty of this paper relies on adapting contrastive explanations in the context of promotional forecast, the focus being interpretability and not accuracy. A cold-start item P is likely to have a market demand that is similar to the one of product Q, which was on promotion in the past. We forecast cold-start promotions leveraging Gradient Boosted Decision Tree (GBDT) methods in an interpretable manner. As the data for new promotional items is non-existent, the GBDT methods learn the linkages between new and established items [20] and produce an output relating them to the new product.

The remainder of this paper is organised as follows. Section II presents the framework and describes the proposed method. Section III conducts a series of experiments to evaluate the model, both in artificial and real market data. Section IV discusses the method and its suitability to support managers and forecasters in grocery retailers.

II. CONTRASTIVE EXPLANATIONS WITH GBDT

In this section we present our algorithm for the estimation of the cold-start promotional sales with interpretability. We first frame contrastive explanations in a promotional context and then proceed to describe the input data and how to generate interpretability using conventional GBDT models.

Our method is founded on the idea that the performance of a promotion in terms of sales is largely determined by the promotional parameters. For example, a promotion on ice cream at a great discount during summertime is expected to have larger sales than the same ice cream at an expensive price during wintertime. This is also equivalent to expecting that the sales will be similar to the ones during the last summer, which constitutes a *congruent contrastive explanation*. To present predictions as a cause and effect mechanism, we suggest a paradigm shift from the conventional GBDT application where traditionally, the GBDT regressor is trained to directly predict the response variable. In this paper we aim to forecast the difference in sales between pairs of promotions, so we arrange the input data in a particular shape that allows us to produce forecasts as comparisons between promotions.

The promotional data collected by retailers can typically be divided into four categories, namely, numerical, binary, time-date, and categorical variables. Amongst the numerical variables, is it common to find information about the number of stores running the promotion, the prices before the promotion is launched and during the promotion, as well as the discount. Binary variables, which are a particular case of categorical variables, are used to inform about features such as a product being seasonal or the placement of a product on a featured display, amongst others. The date and date variables normally mark starting and end dates. The categorical variables describe the type of store, shelf, or the type of promotion [21]. Clearly, there are confounding variables that affect the performance of a promotion although the present work focuses on the ones that retailers can directly measure.

A. METHOD

To explain the arrangement of this input data, let us introduce the terms and notation followed along this paper. Given a set of m observed sales promotions, we represent the explanatory variables composed of n_n numerical, n_c categorical and n_d date variables as $\mathbf{X} \in \mathbb{R}^{m\times n}$, where $n = n_n + n_c + n_d$. We use $\mathbf{y} \in \mathbb{R}^m$ to represent the response variable of the m promotions, which typically is the amount of sales in units, or sometimes a compound metric of the sales that factors the sales and other market indicators. This is the set $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$, whose matrices are

$$\mathbf{X} = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & \dots & x_n^{(1)} \\ x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & \dots & x_n^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_1^{(m)} & x_2^{(m)} & x_3^{(m)} & \dots & x_n^{(m)} \end{bmatrix} \text{ and } \mathbf{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix}, \quad (1)$$

where superscript $^{(i)}$ denotes the i-th row, a promotion, of a matrix and subscript $_j$ denotes the j-th column or variable of a matrix.

With this notation, we describe next the process of building a contrastive training set, which we refer to as extended matrix, $\mathbf{X^{ext}}$. Note that we use bold superscript or subscript to identify different types of elements, vectors or matrices. This dataset comes from repeating the following process iteratively. From $\mathcal{D}$, we randomly select a first *reference* observation $(\mathbf{x^{ref_1}}, y^{\mathbf{ref_1}})$, which is compared to another k observations that we call *neighbours*. These k promotions are also randomly drawn from $\mathcal{D}$ with the restriction that they must have occurred at a time prior to the reference promotion, formally $(\mathbf{X}^{\mathbf{neig}}, \mathbf{y}^{\mathbf{neig}})$, where $\mathbf{X}^{\mathbf{neig}} = \left[\mathbf{x}^{\mathbf{neig}(1)^\top}, \ldots, \mathbf{x}^{\mathbf{neig}(k)^\top}\right]^\top$ and $\mathbf{y}^{\mathbf{neig}} = \left[y^{\mathbf{neig}(1)}, \ldots, y^{\mathbf{neig}(k)}\right]^\top$. This is done to preserve the time structure of the forecasting problem and also to provide diversity during the training stage.

We now arrange the submatrices with $p = n_n + n_c$ numerical and categorical columns. The reference promotion part is $\tilde{\mathbf{x}}^{\mathbf{ref_1}} = \left[\tilde{x}_1^{\mathbf{ref_1}}, \ldots, \tilde{x}_p^{\mathbf{ref_1}}\right]$. This selection is applied for each of the i-th neighbours too, as $\tilde{\mathbf{x}}^{\mathbf{neig}(i)} = \left[\tilde{x}_1^{\mathbf{neig}(i)}, \ldots, \tilde{x}_p^{\mathbf{neig}(i)}\right]$, being $i = 1, \ldots, k$.

To encode the n_c categorical variables as numerical we use either the James-Stein encoder [22], or the target additive encoding [23], or the *CatBoost* approach [24]. The date and time variables are also manipulated so we can operate on them. Typically both month and season are derived from the launching dates, mainly to account for seasonal effects and trends. The differences in days between the reference promotion and the neighbours are also calculated, adding up to a total of q date-derived variables, which are represented as the following matrix $\mathbf{Q}^{\mathbf{ref}} \in \mathbb{R}^{k \times q}$, as per

$$\mathbf{Q}^{\mathbf{ref}} = \begin{bmatrix} d_1^{\mathbf{ref_1}\text{-}\mathbf{neig}(1)} & \ldots & d_q^{\mathbf{ref_1}\text{-}\mathbf{neig}(1)} \\ \vdots & \ddots & \vdots \\ d_1^{\mathbf{ref_1}\text{-}\mathbf{neig}(k)} & \ldots & d_q^{\mathbf{ref_1}\text{-}\mathbf{neig}(k)} \end{bmatrix}. \tag{2}$$

We then form the k rows of $\mathbf{X}^{\mathbf{ext}}_{\mathbf{ref_1}}$ by concatenating the neighbours and the reference per row as follows,

$$\mathbf{X}^{\mathbf{ext}}_{\mathbf{ref_1}} = \begin{bmatrix} \tilde{\mathbf{x}}^{\mathbf{neig}(1)} & \tilde{\mathbf{x}}^{\mathbf{ref_1}} & \\ \vdots & \vdots & \mathbf{Q}^{\mathbf{ref_1}} \\ \tilde{\mathbf{x}}^{\mathbf{neig}(k)} & \tilde{\mathbf{x}}^{\mathbf{ref_1}} & \end{bmatrix}, \tag{3}$$

with dimensions according to $\mathbf{X}^{\mathbf{ext}}_{\mathbf{ref_1}} \in \mathbb{R}^{k \times (2p+q)}$.

The target variable to train the model on is the difference of the response variable between the reference and the neighbours, as per

$$\mathbf{y}^{\mathbf{ext}}_{\mathbf{ref_1}} = \begin{bmatrix} y^{\mathbf{ref_1}} - y^{\mathbf{neig}(1)} \\ y^{\mathbf{ref_1}} - y^{\mathbf{neig}(2)} \\ \vdots \\ y^{\mathbf{ref_1}} - y^{\mathbf{neig}(k)} \end{bmatrix} = \begin{bmatrix} y_\Delta^{(1)} \\ y_\Delta^{(2)} \\ \vdots \\ y_\Delta^{(k)} \end{bmatrix}. \tag{4}$$

The process detailed above is repeated N times to produce the training set $\mathcal{D}^{\mathbf{ext}} = (\mathbf{X}^{\mathbf{ext}}, \mathbf{y}^{\mathbf{ext}})$ where $\mathbf{X}^{\mathbf{ext}} = \left[\mathbf{X}^{\mathbf{ext}}_{\mathbf{ref_1}}, \ldots, \mathbf{X}^{\mathbf{ext}}_{\mathbf{ref_N}}\right]^\top$ and $\mathbf{y}^{\mathbf{ext}} = \left[\mathbf{y}^{\mathbf{ext}}_{\mathbf{ref_1}}, \ldots, \mathbf{y}^{\mathbf{ext}}_{\mathbf{ref_N}}\right]^\top$.

Figure 1 depicts the construction of $\mathcal{D}^{\mathbf{ext}}$ in an example where there are two reference promotions, and the steps involved in the algorithm are summarised in Algorithm 1. Please note that the input features to our model are directly interpretable as they have not been engineered. This is referred as the transparency property of interpretable models, particularised as decomposability [9] or intelligibility.

Algorithm 1 Pseudo-Code for Building $\mathcal{D}^{\mathbf{ext}}$

Input: $\mathbf{X} \in \mathbb{R}^{m \times n}$, $\mathbf{y} \in \mathbb{R}^{m \times 1}$, N (number of iterations), K (number of neighbours).
Output: $\mathbf{X}^{\mathbf{ext}}$: Extended $\mathbf{X}$ for training. $\mathbf{y}^{\mathbf{ext}}$: Extended $\mathbf{y}$ for training.

```
1: for $n = 1 \to N$ do
2:     Select a random row from $\mathbf{X}$ and $\mathbf{y}$ and name it reference $\mathbf{x}^{\mathbf{ref_n}}$ and $y^{\mathbf{ref_n}}$.
3:     for $k = 1 \to K$ do
4:         Select a random row from $\mathbf{X}$ and $\mathbf{y}$ from the subset of promotions that occurred before the reference one. Name it $\mathbf{x}^{\mathbf{neig}}$ and $y^{\mathbf{neig}}$.
5:         Concatenate the numerical and categorical variables of $\mathbf{x}^{\mathbf{neig}}$ and $\mathbf{x}^{\mathbf{ref_n}}$. Operate on the time-date variables. Append all of them to $\mathbf{X}^{\mathbf{ext}}$.
6:         Append to $\mathbf{y}^{\mathbf{ext}}$ the difference between the values of the target variable $y^{\mathbf{ref}} - y^{\mathbf{neig}}$.
7:     end for
8: end for
```

The set $\mathcal{D}^{\mathbf{ext}}$ is used to train one of the following GBDT methods: *XGBoost* [25], *CatBoost* [26] and *LightGBM* [27] for regression, or *NGBoost* [28] for probabilistic regression. The reasons for using GBDT methods, apart from their accuracy and speed, is that they natively provide with methods to calculate the importance of the input features. The regressors are trained using Bayesian hyper-parameter optimisation [29]. Once the regressor is trained and meets the minimum requirements for accuracy and error, we calculate the feature importance vector $\mathbf{v} \in \mathbb{R}^{(2p+q)}$, which can be broken down into the features that it represents as $\mathbf{v} = [\mathbf{v}^{\mathbf{neig}}, \mathbf{v}^{\mathbf{ref}}, \mathbf{v}^{\mathbf{Q}}]$. This vector plays a fundamental role at forecasting time as it is used to select the closest neighbours to each instance of test set. It is worth noting that to facilitate interpretability, the feature importance vector is presented to the user as the addition of neighbour and reference, which can be written as $\mathbf{v}' = [\mathbf{v}^{\mathbf{neig}} + \mathbf{v}^{\mathbf{ref}}, \mathbf{v}^{\mathbf{Q}}]$.

B. DISTANCES, NEIGHBOURS, AND FORECAST

To illustrate the steps involved in the prediction stage, let us recall the ice cream example mentioned at the beginning of the section. Given that we want to predict the sales of a promotion happening during summertime, in order to form a contrastive explanation we are interested in promotions that have occurred under similar circumstances, such as summertime or hot days during springtime, and at a similar price. In this context, very different promotions do not add a large explanatory value. This concept is translated into our framework by leveraging the components of vector $\mathbf{v}$. Given a new promotion $\mathbf{x}^{\mathbf{test}} \in \mathbb{R}^n$ to forecast for, to create $\mathbf{x}^{\mathbf{ext}}_{\mathbf{test}}$ we need k-neighbours. Instead of selecting them randomly as we do during training, we choose them as the closest promotions to $\mathbf{x}^{\mathbf{test}}$ according to a weighted Euclidean distance. This process involves to greedily calculate all the distances between the

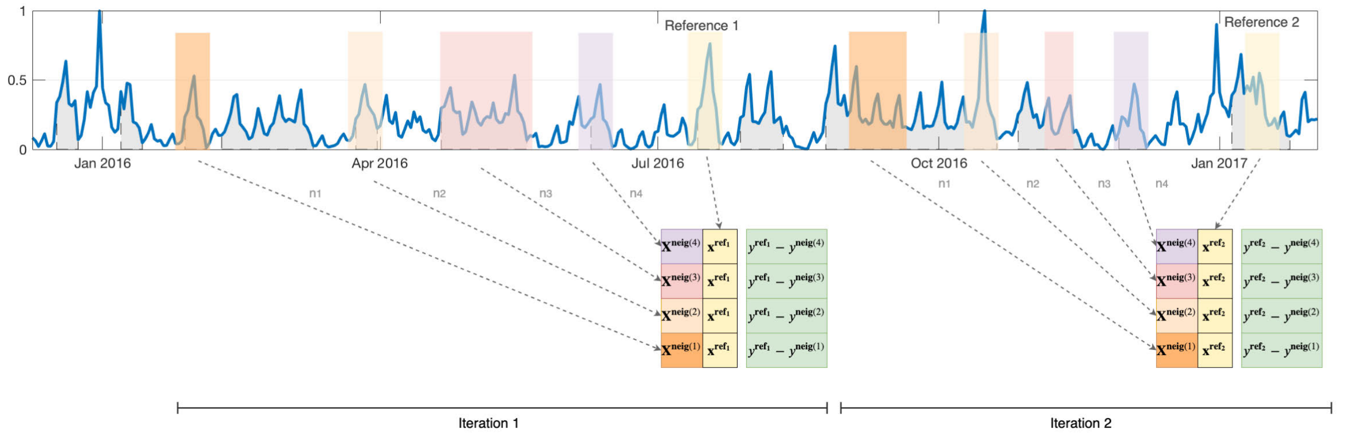

FIGURE 1. Arrangement of the training dataset $\mathbf{X}^{\text{ext}}$. During each iteration, we randomly withdraw a reference row and a set of neighbours (4 in the image) from the input dataset X and stack them.

standardised test promotion and the standardised instances in the training set, denoted as $d(\mathbf{x}^{\mathbf{neig}}, \mathbf{x}^{\mathbf{test}}, \mathbf{v})$. Please refer to the Appendix B for a discussion on the calculation of this weighted Euclidean distance.

The distances are then used to forecast each test promotion using the trained regressor, obtaining k delta values, $\mathbf{y}_\Delta = [y_\Delta^{(1)}, \ldots, y_\Delta^{(k)}]^\top$, that represent the difference between the promotion of interest and the neighbouring promotion. This increment can be subsequently decomposed into the actual forecast by adding the sales of the neighbouring promotions,

$$\begin{bmatrix} y_\Delta^{(1)} \\ y_\Delta^{(2)} \\ \vdots \\ y_\Delta^{(k)} \end{bmatrix} \Rightarrow \begin{bmatrix} y_\Delta^{(1)} + y^{\mathbf{neig}(1)} \\ y_\Delta^{(2)} + y^{\mathbf{neig}(2)} \\ \vdots \\ y_\Delta^{(k)} + y^{\mathbf{neig}(k)} \end{bmatrix} = \begin{bmatrix} \tilde{y}^{\mathbf{test}(1)} \\ \tilde{y}^{\mathbf{test}(2)} \\ \vdots \\ \tilde{y}^{\mathbf{test}(k)} \end{bmatrix} = \tilde{\mathbf{y}}. \quad (5)$$

The idea in this equation is to present the forecast as a contrast to the neighbouring promotions, so the user can compare with the features and performance of the k neighbours. Returning to the ice cream example, if the reference promotion is planned for fall season, the method is expected to present a negative figure when comparing to summertime promotions but most likely a positive one when comparing to wintertime offers.

The final forecast can be calculated with either the standard mean $\hat{y}^{\mathbf{test}} = E[\tilde{\mathbf{y}}]$, or as a weighted mean leveraging the Euclidean distances calculated in Eq. (10), as follows,

$$\hat{y}^{\mathbf{test}} = \frac{\mathbf{w}^\top \tilde{\mathbf{y}}}{\mathbf{w}^\top \mathbf{1}}, \quad (6)$$

where $\mathbf{w} = \mathbf{1} \oslash \mathbf{d}$, $\mathbf{1}$ is a vector with k ones, $\mathbf{d} = [d(\mathbf{x}^{\mathbf{neig}(1)}, \mathbf{x}^{\mathbf{test}}, \mathbf{v}), \ldots, d(\mathbf{x}^{\mathbf{neig}(k)}, \mathbf{x}^{\mathbf{test}}, \mathbf{v})]$ and $\oslash$ denotes the element-wise division. These two approaches for averaging the final forecast are later discussed on the experiments in Section III-C1.

Note that in our setup, the distance between promotions can not be zero because the date time variables are always different. Nevertheless, to grant numerical stability in Eq. (6), the distances are limited to a minimum value of 10^{-3}.

C. FORECAST GOVERNANCE

It is important to verify that the cold-start forecast $\hat{y}^{\mathbf{test}}$ is *reasonable* given the historical promotions. For this purpose, we add a layer of governance. As the forecast is calculated using the closest k-neighbours, we leverage the actual sales of these k-neighbours, $\mathbf{y}^{\mathbf{ext}}_{\mathbf{ref_k}}$, to detect if the cold-start forecast is an outlier. We do so by means of a modified z-score [30]:

$$\frac{0.6745\ |\hat{y}^{\mathbf{test}} - median(\mathbf{y}^{\mathbf{ext}}_{\mathbf{ref_k}})|}{median(|\mathbf{y}^{\mathbf{ext}}_{\mathbf{ref_k}} - median(\mathbf{y}^{\mathbf{ext}}_{\mathbf{ref_k}})|)} \leq \eta. \quad (7)$$

Therefore, when the forecast exceeds η, it is flagged as an outlier and it should be reviewed by the forecasting team. Note that, although the proposed value in [30] for η is 3.5, we recommend setting the threshold η to about 2.5 or 3.0 in order to be conservative, due to the risk assumed in making decisions for promotional sales.

D. ONLINE MODEL TRAINING

Typically, the items catalogue of grocery retailers follows a well-defined hierarchical structure. Different retailers use different naming conventions but generally, items are organised into categories, families and classes. For example, the ice cream mentioned earlier on might belong to the "frozen food" category, to the "ice cream and frozen desserts" family and to the "ice cream tubs" class.

The training set used by method is always at the SKU class level, which in our real-market dataset usually contains between 30 and 200 products. It is stored-aggregated and composed by the past two years historical promotions. This is typically less than 1000 records, allowing us to calculate the forecast in an online fashion, meaning that we train and forecast on-the-fly per cold-start product. This is done for several reasons. Promotional environments are typically non-stationary. By training the model online we make sure the

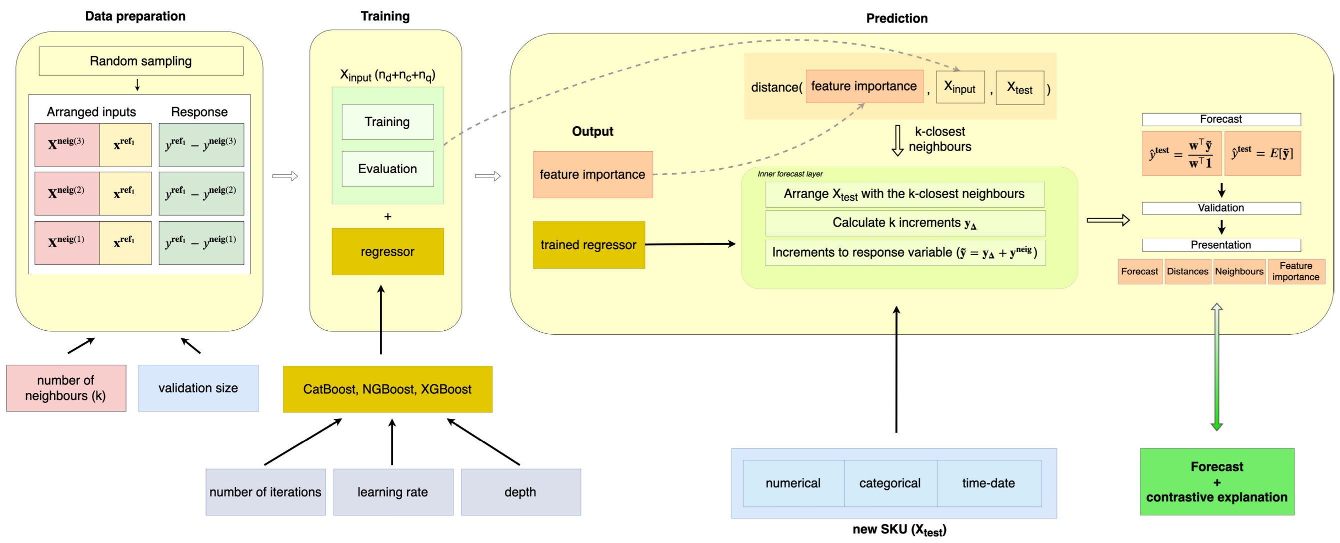
Data preparation
Random sampling
Arranged inputs
Response
$\mathbf{X}^{neig(3)}$ $\mathbf{x}^{ref_1}$ $y^{ref_1} - y^{neig(3)}$
$\mathbf{X}^{neig(2)}$ $\mathbf{x}^{ref_1}$ $y^{ref_1} - y^{neig(2)}$
$\mathbf{X}^{neig(1)}$ $\mathbf{x}^{ref_1}$ $y^{ref_1} - y^{neig(1)}$
Training
X_{input} $(n_d+n_c+n_q)$
Training
Evaluation
+
regressor
Prediction
Output
feature importance
trained regressor
distance(feature importance , X_{input} , X_{test})
k-closest neighbours
Inner forecast layer
Arrange X_{test} with the k-closest neighbours
Calculate k increments $\mathbf{y}_\Delta$
Increments to response variable ($\tilde{\mathbf{y}} = \mathbf{y}_\Delta + \mathbf{y}^{neig}$)
Forecast
$\hat{y}^{test} = \frac{\mathbf{w}^\top \tilde{\mathbf{y}}}{\mathbf{w}^\top \mathbf{1}}$
$\hat{y}^{test} = E[\tilde{\mathbf{y}}]$
Validation
Presentation
Forecast
Distances
Neighbours
Feature importance
number of neighbours (k)
validation size
CatBoost, NGBoost, XGBoost
number of iterations
learning rate
depth
numerical
categorical
time-date
new SKU (X_{test})
Forecast + contrastive explanation

FIGURE 2. Schema representing the contrastive regressor broken into functional blocks. The data preparation block illustrates how the input data is arranged given the number of neighbours. The training block illustrates the the selection of a base regressor and the hyperparameters (they can be optimised through Bayesian hyper-optimisation). The prediction block aims to represent the role of both number of neighbours and feature importance in the method. To calculate the forecast of a new SKU, the feature importance is decisive in selecting the k-neighbours from the input data that are the closest to the new SKU. The neighbours are then used to calculate k predictions of the response variable that are aggregated to calculate the forecast. The forecast and the constrastive explanations are presented to the user, that has the option of modifying the forecast by varying the distances to the neighbours or overriding the feature importances.

latest records available are always used. Another reason is reducing the overhead of model management. This can be done by adding a forecast governance layer (see Section II-C) and managing the lifecycle with open source platforms such as *MLflow*^{TM} [31].

E. MODEL TRANSPARENCY AND USER INTERACTION

One of the properties of interpretable models is transparency, which connotes some sense of understanding of the mechanism by which the model works [9].

In our method, there are two main mechanisms involved in producing a prediction: inner and outer layers. The inner layer is the underlying GBDT model that generates the $\mathbf{y}_\Delta$ values. Aiming to explain the internals of these models is referred as *explainability* and is not the purpose of this paper. SHAP [32] provides local explanations for GBDT in this context. On the other hand, the outer layer is the one that provides transparency and intuition about the prediction produced by our method. The forecast is presented to the user as a comparison with k past promotions (neighbours), showing the importance of the input variables, which explains why the neighbours were selected. It also presents the expected change in sales of the cold-start promotion regarding the neighbour, as well as the distance between the neighbour and the test promotion. This presentation is shown alongside the experiments in Section III, see Table 1.

The user can appropriately contest the algorithmic decision in this outer layer. The most immediate action is to modify the neighbour distances. For example, the method presents a recent and similar promotion that coincided with a sports event that massively increased the sales. The user can also discount the promotion by setting large values for its distance, so it is disregarded in Eq. 6.

The second form of interacting with the forecast is to override the variable importance and re-forecast. For example, this is often done in order to combat demand volatility. In abnormal situations such as crisis, the historical demand of products has little correlation with the actual demand during the crisis. To ensure the neighbours are the most recent records, a user can assign a large weight to the variable that represents the difference in days.

Figure 2 summarises the components described in Section II. The data preparation block illustrates how the input data is arranged given the number of neighbours. The training block illustrates the selection of a base regressor and the hyperparameters (they can be optimised through Bayesian hyper-optimisation). The prediction block aims to represent the role of both number of neighbours and feature importance in the method. To calculate the forecast of a new SKU, the feature importance is decisive in selecting the k neighbours from the input data that are the closest to the new SKU. The neighbours are then used to calculate k predictions of the response variable that are aggregated to calculate the forecast. The forecast and the constrastive explanations are presented to the user, having the option of modifying the forecast by varying the distances to the neighbours, or overriding the feature importances.

III. EVALUATION

In this section, we focus on evaluating the method through a series of experiments. The first experiment analyses the role of the variable importance in the forecast and the explanation.

The second experiment examines the method focusing on the selection of the neighbours in a toy dataset. The third experiment tests the method with real market data, and it is divided into two parts, namely, analysis of the number of neighbours, and comparison with *Extremely Randomized Trees* [33], *CatBoost* [26], *NGBoost* [28] and *AutoGluon* [34], which are widespread state-of-the-art methods used both in academia and industry.

A. ANALYSIS OF THE FEATURE IMPORTANCE

The purpose of this first experiment is to demonstrate that our method is able to recognise the importance of the variables that contribute to the promotional sales, and consequently to select the closest neighbours to a cold-start instance. We also discuss the results that the model produces and their interpretation.

A simple approach to evaluate if the model finds the relevant features that drive the sales is to use a linear model where the contribution of the independent variables to the response variable is known. Let us generate 500 samples of a model with 5 independent variables $(x_1, \ldots, x_5)$ which are drawn from the uniform distribution $U(0, 1)$, being $\mathbf{X} \in \mathbb{R}^{500\times5}$. To define the impact of the independent variables in the response variable, we use a vector of weights $\mathbf{w} = [42, 34, 16, 0, 8]$ being $\sum_{i=1}^{5} w_i = 100$. The response variable $\mathbf{y} \in \mathbb{R}^{500}$ is the linear combination of $\mathbf{w}$ with $\mathbf{X}$ as per $\mathbf{y} = \mathbf{w}^\top\mathbf{X}$. We then train our model on $\mathbf{X}$ and $\mathbf{y}$ using as the base regressor *CatBoost*, a learning rate of 0.08, a validation set of 20%, the tree depth set to 12, 300 iterations and 5 neighbours.

Once trained, the variable importance vector for the neighbours is $\mathbf{v}^{\mathbf{neig}} = [18.84, 17.88, 6.10, 1.75, 2.00]$ and the reference counterpart is $\mathbf{v}^{\mathbf{ref}} = [24.14, 17.75, 6.65, 2.52, 2.36]$. As mentioned in Section II-E, this information is presented to the user as the addition $\mathbf{v}' = \mathbf{v}^{\mathbf{neig}} + \mathbf{v}^{\mathbf{ref}}$, which in our experiment results in $\mathbf{v}' = [42.97, 35.63, 12.75, 4.27, 4.36]$. Despite the simplicity of the linear model, it is remarkable that the feature importances calculated by the method are very close to the actual weights.

The second part of this experiment shows how to interpret the contrastive explanations produced by the method. Let us predict one test instance $\mathbf{x}^{\mathbf{test}} = [0.1, 0.5, 0.5, 0.5, 0.5]$ and $y^{\mathbf{test}} = 33.2$ with the model and present the explanation in Table 1. The table shows in its second row the variable importance colouring the highest value as light yellow and the lowest as dark green. As known from the variable importance calculation, x_1 and x_2 are the variables that have the larger effect on the promotional sales, and therefore, the neighbours of the test promotion have very similar x_1 and x_2 values. Column $\mathbf{y}_\Delta$ contains the raw forecast from the regressor, which is the expected difference between the sales of the neighbour and the cold-start instance. Column $\mathbf{y}$ (sales) contains the historical sales of the neighbours whereas column $\mathbf{y} + \mathbf{y}_\Delta$ represents the addition of the two aforementioned.

Let us interpret the predictions starting with the contrastive explanation given for neighbour 2. The values x_1 and x_2 are both larger in the cold-start instance than in neighbour 2. The model predicts $y_\Delta^{(2)} = 5.62$, meaning that the cold-start promotion is expected to sell 5.62 units more than neighbour 2 did. The forecast with respect to neighbour 2 is $\tilde{y}^{\mathbf{test}(2)} = y_\Delta^{(2)} + y_{\mathbf{neig}}^{(2)} = 33.33$. When comparing to neighbour 3, the cold-start promotion is expected to sell less, as $y_\Delta^{(3)} = -2.88$ units. The forecast $\tilde{y}^{\mathbf{test}(3)} = y_\Delta^{(3)} + y^{\mathbf{neig}(3)} = 34.95$, an error of 1.75 units. The closest neighbour, number 1, is a mix of the two aforementioned cases as x_1 is larger but x_2 is smaller. The actual sales of neighbour 1 are 32.61 units, so the method is right in expecting more sales from the cold-start promotion but it over-forecasts by 1.76 units.

In this experiment we have discussed the explanations generated by our method and proved that they are insightful, valuable, and accurate.

B. PROMOTIONAL PLANNING

In this experiment we focus on how to use our method in the planning phase of a promotion. Let us illustrate the following situation. A retailer is interested in launching a promotion on a cold-start product. During the planning of the promotion the retailer chooses parameters such as the price of the product the type of display. In this experiment we explore five possible combinations of price and type of shelf and discuss how our method produces the forecast and selects the neighbours on a surrogate model.

This data model simulates the sales of 3 SKUs at a country level (constant number of stores) allowing to control the number of times that the product has been on promotion, the baseline sales, the price and discount range, and also the effects on the promotional sales of both the discount and the type of feature display.

We model the baseline sales of each product as a normal distribution $x_b \sim \mathcal{N}(\mu_b, \sigma_b^2)$. We simplify the type of promotions in our model to price cut deals, which allows us to model the discount as a half-normal distribution $x_d \sim \left|\mathcal{N}(0, \sigma_b^2)\right| \in [0, \infty)$. The deal effect curve [35], which connects discount and sales, is modelled as a shifted Gompertz distribution. In this setup, the sales increase with the discount reaching a point where they experience a decay. This saturation is typically observed in stock limited offers. The shifted Gompertz distribution at a time t, with scale parameter b and shape parameter η that is used in our examples is defined as:

$$f(t; b, \eta) = 1 + \left(2.2be^{-bt}e^{-\eta e^{-bt}}[1 + \eta(1 - e^{-bt})]\right) \quad (8)$$

The deal effect curves are shown in Figure 3, where the blue line represents product P1, a product where discounts do not make a large difference as the product is normally included in the regular basket, for example, produce products. Product P2 is represented by the orange line, and in this case, the discount rapidly increases the sales up to the point of running out of stock, affecting sales negatively. Finally, in green colour line there is product P3, which represents a real case scenario that we have particularly observed in Asian markets, where there are certain products for which

TABLE 1. Constrastive explanation of a Cold-Start forecast from Experiment 1. The variable importance is presented in the second row. Column $\mathbf{y}_\Delta$ contains the raw forecast from the regressor which is the expected difference between the sales of the neighbour and the cold-start instance. For example, the cold-start promotion is expected to sell 6.19 units less than neighbour 4, which has larger x_1 and x_2 values. Column y (sales) shows the actual sales amount for the neighbour. Column $\mathbf{y} + \mathbf{y}_\Delta$ is the addition of the change in sales and the neighbour sales. The last column shows the weighted Euclidean distance between the neighbours and the cold-start promotion.

	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$				
feature importance	42.97	35.63	12.75	4.27	4.36				
cold-start	0.10	0.50	0.50	0.50	0.50	$\mathbf{y}_\Delta$	$\mathbf{y}$ (sales)	$\mathbf{y} + \mathbf{y}_\Delta$	distances
neighbour 1	0.13	0.42	0.51	0.26	0.57	2.35	32.61	34.96	0.72
neighbour 2	0.07	0.40	0.41	0.36	0.58	5.62	27.72	33.33	0.79
neighbour 3	0.13	0.61	0.48	0.75	0.52	-2.88	37.83	34.95	0.84
neighbour 4	0.12	0.59	0.63	0.75	0.58	-6.19	39.61	33.42	0.89
neighbour 5	0.23	0.48	0.44	0.48	0.49	-4.22	36.98	32.76	0.90

TABLE 2. Parameters of the surrogate model in Experiment 2 for the 3 simulated products.

Product	Observations	baseline sales	σ discount	feature display (c)	display impact (δ)	Gompertz (b, η)	μ price
P1	110	80 ± 6	8.2	[12, 4, 8, 6]	2.5	(0.3,8)	80
P2	80	40 ± 4.5	6	[5, 4, 3, 6]	1.5	(0.5,9)	100
P3	95	140 ± 8	1.75	[8, 4, 14, 6]	1.85	(0.8,4)	62

a small price cut can rapidly increase sales. As a byproduct the shelves are empty very quickly resulting in a negative customer experience.

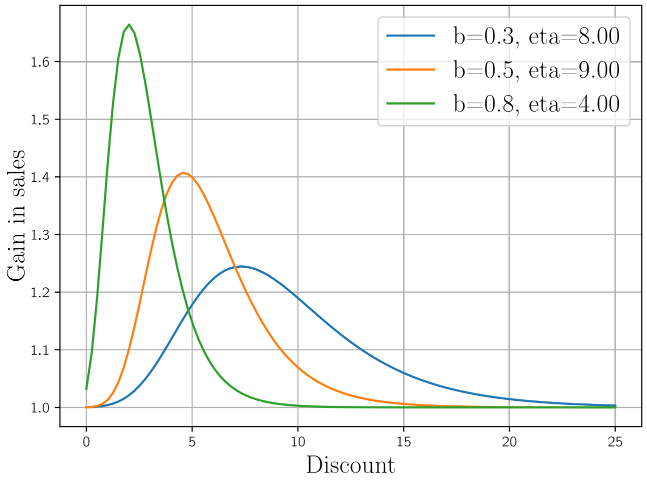

FIGURE 3. Deal effect curve modelled as a shifted Gompertz function of the 3 simulated products in Experiment 2.

The effect of shelf or feature capacity is simply modelled as a direct increment to the sales as per $4.8 \cdot 10^{-3} \cdot \delta c$, where δ models the impact of the shelf in that product and c is the capacity of the shelf. The independent variables of the model are the market price (includes the discount) and the feature type, which is encoded as a categorical variable. The response variable is modelled as the baseline sales scaled by the Gompertz function of the discount and also dependant on the shelf capacity, calculated as $y = x_b \cdot f(x_d; b, \eta) + 4.8 \cdot 10^{-3} \cdot \delta c$. The parameters of the toy model for the 3 products are summarised in Table 2.

Our model is trained on the variables discount and feature type, which is mapped to numerical as mentioned in Section II. The number of neighbours is to 5, the training split to 25% and the validation test size to 20%. We use *CatBoost*

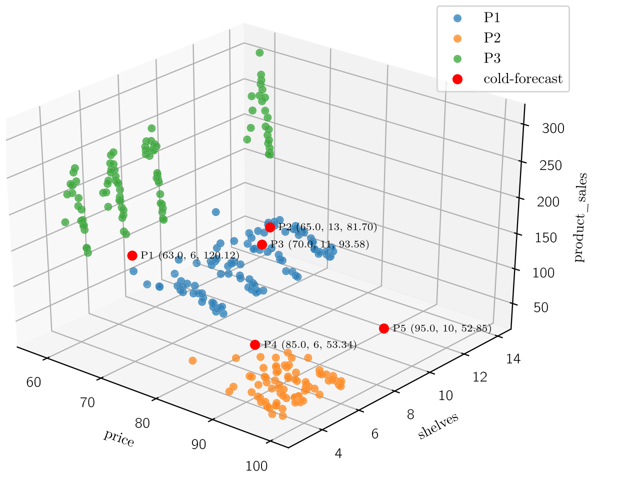

FIGURE 4. Plot of the promotional planning problem and its predictions for different scenarios. The x and y axis are the input variables price and feature capacity, respectively. The z-axis represents the historical sales per product with the exception of the red dots that are the predicted sales.

as the underlying regressor, where the number of iterations is 543, the learning rate 0.09, the tree depth 10, and the cost function is the root mean squared error (RMSE). From the training process we retrieve the variable importance vector, corresponding to 42.95 for the price and 6.53 for the type of feature display.

During the planning phase, the retailer wants to see the sales predictions of the 5 following (*price*, *shelf*) combinations: [(63, 6), (65, 13), (70, 11), (85, 6), (95, 10)]. The forecast and contrastive explanations are shown in Table 3 and a spatial interpretation of the training set, along with the cold-start forecast offered in Figure 4. Let us analyse the predictions. For example, in the first scenario, the product will debut in the market with a price of 63 units and a shelf type 6. The historical promotions in that region of the space are mainly from P3 and only one from P1. Given that the

TABLE 3. Results for the cold-start forecast of a simulated product in 5 different promotional planning scenarios. The reference product column shows the 5 closest neighbours. The value of features price and shelves (along with their importance) are presented in columns 2 and 3. Column y_Δ shows the difference between the cold-forecast and the historical promotion. This parameter is part of the contrastive explanation. It shows that the promotion to be forecast for is expected to perform differently from the neighbouring one in that number of units. Column $y + y_\Delta$ is the new forecast according to that past promotion. Column distances shows the scaled Euclidean distance between reference promotion and cold start promotion. It is used to calculate the prediction presented in each forecast row.

ref product	**price** (vi: 42.95)	**shelves** (vi: 6.53)	y (sales)	y_Δ	$y + y_\Delta$	**Distance**
P1	63.14	6.00	98.24	11.38	109.62	0.14
P3	61.99	6.00	147.81	-1.57	146.24	1.02
P3	61.99	4.00	145.85	0.58	146.42	1.06
P3	61.83	4.00	151.94	-17.33	134.61	1.21
P3	61.77	4.00	152.17	-17.58	134.59	1.27
1-new product	**63**	**6**	forecast: **120.12**			
P1	65.61	6.00	86.10	-5.02	81.08	0.62
P1	64.96	10.00	87.59	-7.39	80.20	0.87
P1	64.64	8.00	94.19	-28.03	66.16	1.02
P1	65.63	10.00	96.15	0.72	96.87	1.08
P1	64.35	12.00	83.61	2.33	85.94	1.11
2-new product	**65**	**13**	forecast: **81.70**			
P1	70.72	6.00	115.03	-17.83	97.19	0.74
P1	70.01	12.00	117.32	-15.41	101.91	0.89
P1	70.63	10.00	108.61	-16.11	92.50	1.08
P1	70.51	8.00	107.56	-19.28	88.27	1.08
P1	70.69	8.00	102.63	-18.93	83.71	1.18
3-new product	**70**	**11**	forecast: **93.58**			
P2	84.90	5.00	47.30	1.84	49.14	3.14
P2	88.66	6.00	45.51	-0.15	45.36	3.70
P2	80.52	4.00	44.25	-3.43	40.82	4.53
P1	79.97	8.00	91.11	-15.93	75.18	5.16
P2	89.22	5.00	61.35	2.73	64.07	5.29
4-new product	**85**	**6**	forecast: **53.34**			
P2	94.98	4.00	58.35	2.33	60.68	0.67
P2	94.43	4.00	45.27	0.56	45.83	0.88
P2	95.61	4.00	49.16	0.96	50.11	0.91
P2	95.68	4.00	54.15	0.94	55.09	0.96
P2	95.93	4.00	43.01	6.36	49.36	1.15
5-new product	**95**	**10**	forecast: **52.85**			

TABLE 4. Comparing different regression methods on 17 cold-start products sorted by MAPE values. In bold letters the method presented in this paper.

Method	MAPE	MAE	frc_bias	frc_acc	RMSE	mean Error	R^2	frc_error
Extremely Randomized Trees (ERT)	36.87	469.45	-0.06	1.06	739.33	-208.41	0.99	0.14
contrastive weighted CatBoost	**54.35**	**940.18**	**-0.04**	**1.04**	**1991.29**	**-154.88**	**0.92**	**0.27**
contrastive weighted NGBoost	**62.42**	**795.80**	**-0.14**	**1.14**	**1410.37**	**-582.37**	**0.96**	**0.23**
AutoGluon weighted ensemble k0 l1	65.15	830.76	-0.18	1.18	1384.44	-741.63	0.96	0.24
AutoGluon ERT MSE	67.45	842.41	-0.15	1.15	1304.69	-617.60	0.97	0.25
AutoGluon Catboost	71.87	966.34	-0.11	1.11	1354.80	-441.59	0.96	0.28
contrastive CatBoost	**74.06**	**880.42**	**-0.05**	**1.05**	**1259.02**	**-185.63**	**0.97**	**0.26**
AutoGluon Random Forest MSE	76.25	731.03	-0.16	1.16	1349.77	-677.91	0.96	0.21
AutoGluon KNN Dist	92.13	4660.38	-0.55	1.55	15413.26	-4162.16	-3.76	1.36
original CatBoost	95.29	1418.08	-0.27	1.27	2672.18	-1243.45	0.86	0.41
AutoGluon KNN Unif	103.83	4709.99	-0.58	1.58	16054.59	-4689.35	-4.16	1.37
AutoGluon LightGBM Custom	115.44	1035.31	-0.23	1.23	1819.01	-1021.35	0.93	0.30
AutoGluon LightGBM	116.92	870.88	-0.13	1.13	1381.21	-525.97	0.96	0.25
contrastive NGBoost	**119.78**	**1223.26**	**-0.23**	**1.23**	**2346.70**	**-1026.09**	**0.89**	**0.36**
original NGBoost	486.21	3568.07	-0.42	1.42	6493.72	-2489.26	0.16	1.04
AutoGluon NeuralNet	1409.70	22072.39	-0.87	1.87	38627.58	-22058.87	-28.88	6.42

weight of the variable price is considerably larger that the shelf variable, the neighbours are selected mainly by their price proximity. Our method selects a point from P1 as the closest, and it is interesting to see the interpretation. The selling price of P1 is on average 73.46, considerably larger than 63. Our model infers that the new product at that price

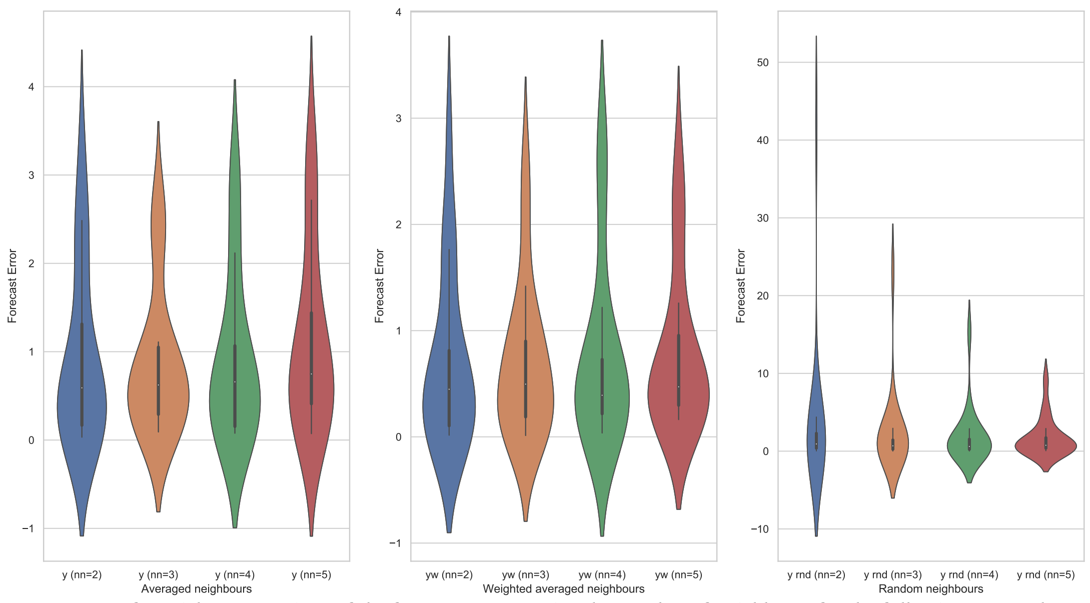

FIGURE 5. **(Left to right) Comparison of the forecast error varying the number of neighbours for the following approaches: averaged neighbours, weighted average based on the distance and random neighbours. The random neighbours option produces forecast errors 10 times larger than the averages.**

should sell more that P1, and it therefore sets the $\mathbf{y}_{\Delta}$ value to 11.38 units more than P1. With regards to that P1 promotion, our new product is expected to sell $\mathbf{y} + \mathbf{y}_{\Delta} = 109.62$ units. The second neighbour is one promotion from product P2, which normally trades at a lower price. Our model infers that even if the product is on the same type of shelf, because the price of the cold-start product is larger, it is expected to sell -1.57 units less. The same reasoning applies to the remaining instances. Finally, these forecast are combined by means of the distances to predict that the new product will sell around 120.12 units.

When reviewing the planing predictions, a forecast analyst can change the contribution of a neighbour, by varying either the quantity in $\mathbf{y} + \mathbf{y}_{\Delta}$ or the distance, which can be set to a high value to disregard a particular neighbour. It is also possible to override the variable importance to return a different set of neighbours.

C. REAL MARKET DATA

In this third experiment, we apply our method to forecasting real market products from a European retailer and evaluate the results. It is a common belief that interpretability means sacrificing accuracy. Our findings imply quite the contrary. In this section we demonstrate that our explainable method is as accurate as the state-of-the-art methods that we benchmark against.

The data used in the experiment was collected circa 2016 and contains approximately a year of promotional sales for different retail categories. The dataset features the number and type of stores, the dates when the promotions occurred, the regular and promotional prices and the category that each product belongs to. Some factors are worth considering about the data. It only features the promotional sales, so assumptions about the regular demand or popularity of the products (including the cold-start ones) can not be made. The cold-start products can range from a completely new product, to a new flavour of a existing SKU, to a product with a new presentation or packaging. Some of these factors positively contribute to the high accuracy of the cold-start forecasts tested along this section. This dataset contains 17 products that have been on promotion only one time and they conform our cold-start test set and they are salty snacks, jelly beans, pre-cooked rice, chocolate, Malvasia wine, lassi, and shampoo.

1) NUMBER OF NEIGHBOURS AND DISTANCES

In this section we discuss how the number of neighbours affect the forecast and we also analyse the method used to select the distance amongst them. One may think that the effort in selecting the closest neighbours must be justified. At the very least, it has to be better than just selecting random neighbours. We also evaluate if predicting with the weighted average (derived from the distances) is more accurate than using the average of the sales of the neighbours. We set to demonstrate that this is the case by forecasting the 17 products and comparing the forecast error, which is defined in the Appendix table 5.

We sweep the number of neighbours from 2 to 5, both inclusive. As we can see in Figure 5, the first 4 plots from the left side correspond to the average, followed by the 4 weighted average forecasts, and finally the 4 when random neighbours are selected. The forecast error from random neighbours is just out of scale.

2) ACCURACY

Given that our method is built on top of GBDT methods, we compare it directly to the GBDT methods to evaluate the differences in accuracy. specifically, we benchmark against

CatBoost and *NGBoost*. To ensure meaningful pairwise comparisons, we use the same data and same parameters. We also benchmark against another type of trees known as *Extremely Randomized Trees (ERT)* [33] using the Scikit-learn [36] implementation. The fixed parameters for all the tree-based models are 600 iterations, learning rate set to 0.08, validation set of 20% (except ERT), and depth to 12. We use our method with 3 neighbours. We also want to compare our method to the auto-ML framework *AutoGluon* [34] using the same datasets. We use *AutoGluon's* tabular prediction module and apply the fit function without hyper-parameter tuning. For each one of the cold-start predictions, we train all the methods on a dataset that contains all the historical promotions of the category of the product to forecast for. The response variable is the total number of units sold during the promotion.

Ranking forecasting methods using just one metric is difficult and it can be misleading as the metric might not reflect some particularities that are important to the business. Owing to this, we have decided to evaluate the cold-start forecast on the following metrics: MAPE, MAE, forecast bias, forecast accuracy, RMSE, mean error, forecast error and R^2, all defined in Table 5. For a discussion on time-series forecasting error metrics and their implications, we refer to [37]. The results are presented in Table 4, which summarises these metrics per method calculated for the 17 cold-start promotional products and sorted by MAPE values. ERT achieves the best scores overall. Both contrastive weighted implementations of *CatBoost* and *NGBoost* produce smaller error figures than the original implementations. The central part of the table is dominated by the *AutoGluon* models. The largest overall errors are produced by the *AutoGluon* neural network regressor.

IV. CONCLUSION

Cold-start forecasting is a difficult but common problem in a variety of disciplines as no historical information exists on the subject of the forecast. This problem is particularly challenging in retail promotions, as the complexity of promotional forecasting is added to the cold-start difficulty. In this paper, we proposed to solve this problem through interpretable ML. We align with authors such as Rudin [38] in the statement that we should move away from black-box models where the decisions made by automated systems require interpretability. Furthermore, in scenarios as such, the lack of interpretability directly affects the usability of a prediction as it inherently produces mistrust. Given the nature of the grocery retail business, forecasting affects many elements of the supply chain, from producers, suppliers, distributors, and customers. The implications are such that retailers commonly employ teams of analysts that validate and modify promotional forecasts generated by automated systems. Our method has been designed for these analysts. Apart from automatically estimating the sales of a cold-start promotion, the method produces explanations on how the prediction has been made, and also provides appropriate mechanisms to interact with the forecast as well as to detect outlier forecast that should be flagged for review.

In this work, we achieved cold-start promotional forecasting and interpretability altogether, which to the best of our knowledge has not yet been addressed in the literature. Our method is divided into two layers: an inner layer that calculates the regression using GBDT (*CatBoost*, *XGBoost*, *LightGBM*, and *NGBoost* have been tested), and an outer layer that both calculates the forecast, and informs about it using congruent contrastive explanations. One of the particularities of our method is that the GBDT regressors are trained to forecast the difference in sales between pairs of promotions, instead of a direct estimation of the sales. In order to produce a cold-start forecast, the k historical promotions, or neighbours, are selected from the training set according to the closest values in the most relevant features. Their search is done through a weighted Euclidean distance, where the weights are the importance of the features calculated by the GBDT regressor during the training stage. To forecast one promotion, the inner layer produces k predictions which are subsequently aggregated by the outer layer using the inverse of the weighted Euclidean distance. Moreover, the method allows user interaction to manipulate the predictions by simply varying the distances or by overriding the variable importance to select different neighbours.

We recommend to tune the hyperparameters of the model (number of iterations, depth and learning rate) using Bayesian hyper-parameter optimisation although the parameters can also be fixed.

Most retailers group their products into relevant classes. We train our method on all the historical promotions pertaining to the class of the product that we forecast for. This poses a limitation as there might be cases when the cold-start product sells in a different manner to the SKUs within the class. Equally, if the class does not contain many products or the promotional observations are very scarce, the forecast probably will not be very accurate.

Another potential limitation of the method are outlier promotions. The method is able to remove from the training set promotions that sold very little or featured a small number of stores, but it is not able to detect promotions that sold a atypical number of units. These promotions can lead the method to over-forecasting.

Evaluating interpretability is a non-trivial task and is it not clear how it should be measured [39]. We demonstrated using a linear model that our contrastive explanations are consistent with the model. Particularly, we demonstrated that the method correctly identifies the variables that influence the sales the most, and accordingly finds their importance. We also showed how the contrastive explanations are agreeable with the intuition of a forecast analyst.

We have also addressed the effect of the number of neighbours and the influence of the distance when selecting the neighbours using real market data. Also, we have addressed the myth of interpretability at the cost of accuracy, proving that our method performs at the same level as *CatBoost*, *NGBoost* or *AutoGluon*, which are not interpretable by themselves.

TABLE 5. Summary of the forecast measures used along the paper.

Metric	Short name	Formula
Forecast Error	F_{err}	$\frac{\sum_{i=1}^{N} \lvert y_i - \hat{y}_i \rvert}{\sum_{i=1}^{N} y_i}$
Forecast Bias	F_{bias}	$\sum_{i=1}^{N}(y_i - \hat{y}_i) / \sum_{i=1}^{N} \hat{y}_i$
Forecast Accuracy	F_{acc}	$1 - F_{bias}$
Mean Error	ME	$\frac{1}{N} \sum_{i=1}^{N} y_i - \hat{y}_i$
Mean Absolute Error	MAE	$\frac{1}{N} \sum_{i=1}^{N} \lvert y_i - \hat{y}_i \rvert$
Mean Absolute Percentage Error	MAPE	$\frac{100}{N} \sum_{i=1}^{N} \left\lvert \frac{y_i - \hat{y}_i}{y_i} \right\rvert$
Root Mean Squared Error	RMSE	$\sqrt{\mathbf{E}\left[(\mathbf{y} - \hat{\mathbf{y}})^2\right]}$

Future work should consider filtering outlier promotions and a more strategic sampling of the historical promotions. Also a solution is required for the edge cases when the data in the class is not representative of the product.

We hope that this paper will contribute towards the adoption of interpretable ML in cold-start promotional forecast systems. Aversion to black-boxes can be overcome with contrastive explanations and human-system interaction, where a human understands the consequences of modifying the parameters of the method, in our case, neighbour distances and the importance of the variables, to get a different forecast.

APPENDIX A
NEIGHBOUR DISTANCES

The distance between a promotion to forecast for and the instances in the training set, $d(\mathbf{x}^{\mathbf{neig}}, \mathbf{x}^{\mathbf{test}}, \mathbf{v})$, determines the so-called neighbours, and subsequently the forecast itself as per Eq. (5). It also indicates how far the cold-start instance is from the observed data. This information is valuable to the forecast reviewer in the sense that a promotion that lies far from the ones observed in the training set is more unconventional.

We train the regressor on random neighbours so it learns the differences in the input features related to the difference in the response variable. Therefore, the model should be able to produce a decent forecast with random neighbours. However, the main goal of this paper is interpretability and to achieve that, we present the forecast as the combination of promotions with similar features. The role of the distance is to select these meaningful instances to compare to.

We leverage the feature importance vector for this task in two different manners: symmetrical and asymmetrical weights matrix.

A common approach that follows the axioms of distance measures is to have a symmetric positive definite weights matrix $\mathbf{W}$, that scales both column vectors $\mathbf{x}^{\mathbf{neig}}$ and $\mathbf{x}^{\mathbf{test}}$. This can be achieved by adding the variable importance of the neighbour and reference as $\mathbf{W} = diag(\mathbf{v}) \in \mathbb{R}^{n \times n}$, where $diag(\cdot)$ denotes a diagonal matrix with its main diagonal formed by the elements of the vector given by $(\cdot)$. Let us follow [40] to factorise the weights matrix using Cholesky decomposition as $\mathbf{W} = \mathbf{L}^{\top}\mathbf{L}$ and use it in the distance calculation as follows,

$$\begin{aligned} d(\mathbf{x}^{\mathbf{neig}}, \mathbf{x}^{\mathbf{test}}, \mathbf{v}) &= (\mathbf{x}^{\mathbf{neig}} - \mathbf{x}^{\mathbf{test}})^{\top}\mathbf{W}(\mathbf{x}^{\mathbf{neig}} - \mathbf{x}^{\mathbf{test}}) \\ &= (\mathbf{L}\mathbf{x}^{\mathbf{neig}} - \mathbf{L}\mathbf{x}^{\mathbf{test}})^{\top}(\mathbf{L}\mathbf{x}^{\mathbf{neig}} - \mathbf{L}\mathbf{x}^{\mathbf{test}}). \quad (9) \end{aligned}$$

To avoid recalculations, we recommend to store the standardised and scaled training set $\mathbf{L}\mathbf{x}^{\mathbf{neig}}$ as a property of the predictor class. The caveat of this approach is that in some cases, matrix $\mathbf{W}$ is positive semi-definite so we then default to the direct calculation of $d(\mathbf{x}^{\mathbf{neig}}, \mathbf{x}^{\mathbf{test}}, \mathbf{W})$ skipping the decomposition.

On the other hand, the asymmetrical approach scales the training set instances by $\mathbf{v}^{\mathbf{neig}}$, the calculated importance on the neighbours. The test set is scaled by $\mathbf{v}^{\mathbf{test}}$. The formulation is as follows,

$$d(\mathbf{x}^{\mathbf{neig}}, \mathbf{x}^{\mathbf{test}}, \mathbf{v}) = \sum_{i=1}^{p} \left(\sqrt{v_i^{\mathbf{neig}}} \cdot x_i^{\mathbf{neig}} - \sqrt{v_i^{\mathbf{test}}} \cdot x_i^{\mathbf{test}} \right)^2, \quad (10)$$

where $\mathbf{x}^{\mathbf{neig}}$ represents one of the rows from the training set. With the k-closest neighbours we then arrange the extended test set $\mathbf{X}_{\mathbf{test}}^{\mathbf{ext}}$.

APPENDIX B
FORECAST EVALUATION METRICS

Table 5 summarizes the calculation of the forecast metrics used in the paper.

CODE AND DATA AVAILABILITY

The Python implementation of our method and the surrogate model data are available at https://github.com/CarlitosDev/contrastiveExplanation/.

ACKNOWLEDGMENT

C. Aguilar-Palacios would like to thank Anneke Ely, as in 2012, they worked together on using decision trees to calculate ratios and differences on promotional data, the idea of which led to the inception of this article.

REFERENCES

[1] R. C. Blattberg and S. A. Neslin, *Sales Promotion: Concepts, Methods, and Strategies*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1990.

[2] M. I. Gómez, V. R. Rao, and E. W. Mclaughlin, ''Empirical analysis of budget and allocation of trade promotions in the U.S. supermarket industry,'' *J. Marketing Res.*, vol. 44, no. 3, pp. 410–424, Aug. 2007.

[3] K. L. Ailawadi, B. A. Harlam, J. César, and D. Trounce, ''Promotion profitability for a retailer: The role of promotion, brand, category, and store characteristics,'' *J. Marketing Res.*, vol. 43, no. 4, pp. 518–535, Nov. 2006.

[4] M. Natter, T. Reutterer, A. Mild, and A. Taudes, ''Practice prize report—An assortmentwide decision-support system for dynamic pricing and promotion planning in DIY retailing,'' *Marketing Sci.*, vol. 26, no. 4, pp. 576–583, Jul. 2007.

[5] S. Ma and R. Fildes, ''A retail store SKU promotions optimization model for category multi-period profit maximization,'' *Eur. J. Oper. Res.*, vol. 260, no. 2, pp. 680–692, Jul. 2017.

[6] S. F. Alamdar, M. Rabbani, and J. Heydari, ''Pricing, collection, and effort decisions with coordination contracts in a fuzzy, three-level closed-loop supply chain,'' *Expert Syst. Appl.*, vol. 104, pp. 261–276, Aug. 2018.

[7] C. Xie, A. Tank, A. Greaves-Tunnell, and E. Fox, ''A unified framework for long range and cold start forecasting of seasonal profiles in time series,'' 2017, *arXiv:1710.08473*. [Online]. Available: http://arxiv.org/abs/1710.08473

[8] R. Wen, K. Torkkola, B. Narayanaswamy, and D. Madeka, ''A multi-horizon quantile recurrent forecaster,'' 2017, *arXiv:1711.11053*. [Online]. Available: http://arxiv.org/abs/1711.11053

[9] Z. C. Lipton, ''The mythos of model interpretability,'' *Queue*, vol. 16, no. 3, pp. 31–57, 2018.

[10] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, ''Explaining explanations: An overview of interpretability of machine learning,'' in *Proc. IEEE 5th Int. Conf. Data Sci. Adv. Analytics (DSAA)*, Oct. 2018, pp. 80–89.

[11] M. Fox, D. Long, and D. Magazzeni, ''Explainable planning,'' 2017, *arXiv:1709.10256*. [Online]. Available: http://arxiv.org/abs/1709.10256

[12] P. Lipton, ''Contrastive explanation,'' *Roy. Inst. Philosophy Suppl.*, vol. 27, pp. 247–266, 1990.

[13] P. Lipton, ''Contrastive explanation and causal triangulation,'' *Philosophy Sci.*, vol. 58, no. 4, pp. 687–697, Dec. 1991.

[14] T. Miller, ''Contrastive explanation: A structural-model approach,'' 2018,*arXiv:1811.03163*. [Online]. Available: https://arxiv.org/abs/1811.03163

[15] D. V. Carvalho, E. M. Pereira, and J. S. Cardoso, ''Machine learning interpretability: A survey on methods and metrics,'' *Electronics*, vol. 8, no. 8, p. 832, Jul. 2019.

[16] G. Montavon, W. Samek, and K.-R. Müller, ''Methods for interpreting and understanding deep neural networks,'' *Digit. Signal Process.*, vol. 73, pp. 1–15, Feb. 2018.

[17] F. Doshi-Velez and B. Kim, ''Towards a rigorous science of interpretable machine learning,'' 2017, *arXiv:1702.08608*. [Online]. Available: http://arxiv.org/abs/1702.08608

[18] W. James Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu, ''Interpretable machine learning: Definitions, methods, and applications,'' 2019, *arXiv:1901.04592*. [Online]. Available: http://arxiv.org/abs/1901.04592

[19] F. Doshi-Velez and B. Kim, ''Considerations for evaluation and generalization in interpretable machine learning,'' in *Explainable and Interpretable Models in Computer Vision and Machine Learning*. Cham, Switzerland: Springer, 2018, pp. 3–17, doi: 10.1007/978-3-319-98131-4_1.

[20] J.-H. Böse, V. Flunkert, J. Gasthaus, T. Januschowski, D. Lange, D. Salinas, S. Schelter, M. Seeger, and Y. Wang, ''Probabilistic demand forecasting at scale,'' *Proc. VLDB Endowment*, vol. 10, no. 12, pp. 1694–1705, Aug. 2017.

[21] C. Aguilar-Palacios, S. Munoz-Romero, and J. L. Rojo-Alvarez, ''Forecasting promotional sales within the neighbourhood,'' *IEEE Access*, vol. 7, pp. 74759–74775, 2019.

[22] C. N. Morris, ''Parametric empirical Bayes inference: Theory and applications,'' *J. Amer. Stat. Assoc.*, vol. 78, no. 381, pp. 47–55, Mar. 1983.

[23] D. Micci-Barreca, ''A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems,'' *ACM SIGKDD Explor. Newslett.*, vol. 3, no. 1, pp. 27–32, 2001.

[24] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, ''CatBoost: Unbiased boosting with categorical features,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 6638–6648.

[25] T. Chen and C. Guestrin, ''XGBoost: A scalable tree boosting system,'' 2016, *arXiv:1603.02754*. [Online]. Available: http://arxiv.org/abs/1603.02754

[26] A. Veronika Dorogush, V. Ershov, and A. Gulin, ''CatBoost: Gradient boosting with categorical features support,'' 2018, *arXiv:1810.11363*. [Online]. Available: http://arxiv.org/abs/1810.11363

[27] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, ''LightGBM: A highly efficient gradient boosting decision tree,'' in *Proc. NIPS*, 2017, pp. 1–9.

[28] T. Duan, A. Avati, D. Y. Ding, S. Basu, A. Y. Ng, and A. Schuler, ''NGBoost: Natural gradient boosting for probabilistic prediction,'' 2019, *arXiv:1910.03225*. [Online]. Available: https://arxiv.org/abs/1910.03225

[29] J. Snoek, H. Larochelle, and R. P. Adams, ''Practical Bayesian optimization of machine learning algorithms,'' 2012, *arXiv:1206.2944*. [Online]. Available: https://arxiv.org/abs/1206.2944

[30] B. Iglewicz and D. C. Hoaglin, *How to Detect and Handle Outliers*, vol. 16. Milwaukee, WI, USA: American Society for Quality Press, 1993.

[31] M. Zaharia, A. Chen, A. Davidson, A. Ghodsi, S. A. Hong, A. Konwinski, S. Murching, T. Nykodym, P. Ogilvie, M. Parkhe, and F. Xie, ''Accelerating the machine learning lifecycle with MLflow,'' *IEEE Data Eng. Bull.*, vol. 41, no. 4, pp. 39–45, Jun. 2018.

[32] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee, ''Explainable AI for trees: From local explanations to global understanding,'' 2019, *arXiv:1905.04610*. [Online]. Available: http://arxiv.org/abs/1905.04610

[33] P. Geurts, D. Ernst, and L. Wehenkel, ''Extremely randomized trees,'' *Mach. Learn.*, vol. 63, no. 1, pp. 3–42, Apr. 2006.

[34] N. Erickson, J. Mueller, A. Shirkov, H. Zhang, P. Larroy, M. Li, and A. Smola, ''AutoGluon-tabular: Robust and accurate AutoML for structured data,'' 2020, *arXiv:2003.06505*. [Online]. Available: https://arxiv.org/abs/2003.06505

[35] H. J. Van Heerde, P. S. H. Leeflang, and D. R. Wittink, ''Semiparametric analysis to estimate the deal effect curve,'' *J. Marketing Res.*, vol. 38, no. 2, pp. 197–215, May 2001.

[36] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, ''Scikit-learn: Machine learning in Python,'' *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.

[37] A. Davydenko and R. Fildes, ''Forecast error measures: Critical review and practical recommendations,'' in *Business Forecasting: Practical Problems and Solutions*, vol. 34. Hoboken, NJ, USA: Wiley, 2016.

[38] C. Rudin, ''Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead,'' *Nature Mach. Intell.*, vol. 1, no. 5, pp. 206–215, 2019.

[39] C. Molnar, *Interpretable Machine Learning*. Abu Dhabi, United Arab Emirates: Lulu.Com, 2019.

[40] R. Kurniawati, J. S. Jin, and J. A. Shepherd, ''Efficient nearest-neighbour searches using weighted euclidean metrics,'' in *Advances in Databases*, S. M. Embury, N. J. Fiddian, W. A. Gray, and A. C. Jones, Eds. Berlin, Germany: Springer, 1998, pp. 64–76. [Online]. Available: https://link.springer.com/chapter/10.1007/BFb0053472

CARLOS AGUILAR-PALACIOS received the B.Sc. degree in engineering from Universidad de Castilla-La Mancha, the M.Sc. degree in engineering from Universidad Rey Juan Carlos, and the M.Sc. degree in physics and mathematics from Universidad de Castilla-La Mancha, in 2011. He is currently pursuing the Ph.D. degree in machine learning with Universidad Rey Juan Carlos.

SERGIO MUÑOZ-ROMERO received the degree in engineering and the Ph.D. degree in machine learning from Universidad Carlos III, Spain, in 2009 and 2015, respectively. His current research interests include machine learning algorithms and statistical learning theory, mainly dimensionality reduction and feature selection methods.

JOSÉ LUIS ROJO-ÁLVAREZ (Senior Member, IEEE) received the B.Sc. degree in telecommunication engineering from the University of Vigo, in 1996, and the Ph.D. degree in telecommunication engineering from the University Politécnica de Madrid, in 2000. He is currently a Professor with the Department of Signal Theory and Communications, University Rey Carlos, Spain. His research interests include statistical learning methods for signal and image processing, arrhythmia mechanisms, robust signal processing methods for cardiac repolarization, and Doppler image post-processing. He has (co)authored more than 120 international articles. He has contributed to more than 160 conference proceedings.