TRUST-AI

D6.2 Initial validation of the explainable AI models from business experts

Use case 2 - Online Retail

LTPlabs

Reviewing partner: Apintech

January 25, 2023

D6.2 Initial validation of the explainable AI models from business experts

Use case 2 - Online Retail

LTPlabs

January 25, 2023

Document Control Page

DOCUMENT	D6.2 Initial validation of the explainable AI models from business experts
TYPE	Report
DISTRIBUTION LEVEL	Public
DUE DELIVERY DATE	25/01/2023
DATE OF DELIVERY	25/01/2023
VERSION	0.4
DELIVERABLE RESPONSIBLE	LTPlabs
AUTHOR (S)	Francisco Amorim (LTPlabs), André Morim (LTPlabs), Fábio Moreira (INESC TEC), Daniela Fernandes (INESC TEC)
OFFICIAL REVIEWER/s	Nikos Sakkas (Apintech)

Document History

VERSION	AUTHORS	DATE	CONTENT AND CHANGES
0.1	André Morim (LTPlabs)	16/09/2022	First draft
0.2	André Morim (LTPlabs)	25/09/2022	Second draft considering partners contributions (Version submitted to reviewers).
0.3	André Morim (LTPlabs)	30/09/2022	Final revision
0.4	André Morim (LTPlabs)	25/01/2023	Revision of rejection letter feedback

Acknowledgments

Name	Partner
Nikos Sakkas	Apintech

Contents

List of Figures

List of Tables

Acronyms

ABT Advanced-Booking Time. 8, 26, 27, 41, 47, 48

AHD Attended Home Delivery. 1, 5, 7, 9, 32, 39

AI Artificial Intelligence. 2

BBM Black-Box Model. 2, 17

CTS Cost To Serve. 22, 23, 27, 31, 39, 40, 47–50

DTSPP Dynamic Time Slot Pricing Problem. 2, 5, 7, 8, 13, 14, 16, 17, 20, 25–28, 32, 36, 37, 41, 43, 45, 47–50

GBM Gradient Boosting Machine. v, 10, 15, 17, 22, 31, 33, 36–39, 47

GP Genetic Programming. 2, 10, 11, 47–49

MDP Markov Decision Process. 5, 8

ML Machine Learning. 2

VRPTW Vehicle Routing Problem With Time Windows. 2, 6, 9, 22, 39

WTP Willingness To Pay. 2, 10, 11, 14, 16, 17, 20, 21, 27, 29, 31–33, 35, 36, 39, 43, 47–50

XAI explainable artificial intelligence. 2

Chapter 1

Introduction

Online retail has become an important channel across different industries, including fashion, electronics, and grocery. In 2020, the total volume of retail sales contracted due to the imposed COVID-19 quarantines and business closures. Despite this phenomenon, customers turned to online channels to acquire goods, resulting in a total volume of retail e-commerce sales of $3.915B, a 16.5% increase comparing to 2019's figures [8]. It is expected that the momentum introduced by the pandemic will last as the majority (76%) of the customers that increased online spending admit they will not return to the pre-outbreak buying patterns [12].

In grocery retail, although in-store pickups are available, companies often perform home deliveries. Due to the perishability of the products sold in this business context, customers must attend home deliveries, and thus, e-grocers resort to Attended Home Delivery (AHD). In such delivery systems, the retailer delivers goods to customers within previously agreed delivery time slots. Typically, customers' requests proceed as follows: customers add the set of products to be purchased to their shopping baskets and then proceed to the checkout; at this point, if they have not logged in, they will be asked to do so. At the checkout, the customer is shown a specific price panel containing all the available time slots. The customer then either chooses one of the available slots to receive his shopping basket or walks away without finalizing his purchase.

While AHD avoids costly delivery failures, as customers have compromised on being home at the moment of delivery, managing attended home deliveries entails a trade-off between operational efficiency and customer satisfaction. On the one hand, the company will want to address its customer base at the least cost possible. Suppose the company is incapable of managing the demand and purely attends to customer preferences. In that case, there will be an unbalanced logistic load through time as customers will tend to choose similar time slots, e.g., late afternoon time slots after work hours. Consequently, either the company's operation scales up to match peak demand, or the promised delivery dates are in jeopardy. Moreover, geographically dispersed customers may place orders over the same time slot, requiring much longer routes than if demand were geographically clustered for each time window. On the other hand, if delivery efficiency is preferred, customers will have fewer time slots to choose from or at unfair prices, leading to decreased customer satisfaction and potential dropouts.

Therefore, there is an interest on the part of retailers in obtaining a more profitable operation by applying demand management techniques. The key idea is to steer customers' choice to delivery time slots with lower operating costs, thus maximizing profits, while maintaining an adequate level of customer service. To accomplish so, retailers might resort to different demand management methods, e.g., restricting the number assortment of time slots they present to particular customers (slotting), offering discounts or shopping points for less attractive time slots, etc.

From the many demand management tools retailers have at their disposal, pricing, particularly dynamic pricing, stands out as a mighty lever to manage demand in real-time. With dynamic pricing, the e-grocer is able to best adapt the price panel for each incoming customer by taking into account not only already accepted orders but expected future requests, as well as incoming customer information, i.e., their location, profit resulting from their shopping basket, and their Willingness To Pay (WTP) for different time slots.

The underlying optimization problem of the time slot prices is known as the Dynamic Time Slot Pricing Problem (DTSPP). Its standard formulation consists of a stochastic dynamic programming model used as a framework by several authors in academic literature [23],[9],[10]. The DTSPP model, however, cannot be solved to optimality for several reasons. First, it suffers from the curses of dimensionality [16]. Second, deriving the opportunity costs, which encompass delivery costs, requires solving a Vehicle Routing Problem With Time Windows (VRPTW) for each incoming customer and time slot. Third, the customer's choices ultimately result from the decision, i.e., the time slot prices, and must be anticipated via a customer's choice model.

Two more sources of complexity arise from the application in a fast-paced real e-commerce environment. As it is an online problem, the decision concerning which price panel to show to the customer must be made in fractions of a second. Additionally, the DTSPP involves the cooperation of stakeholders from different company divisions, namely, marketing and operations. Given the distinct goals of the involved parties, a transparent and reliable tool is required to assist the dialogue between them.

To tackle the challenges posed by employing the DTSPP in a real environment, we propose a problem decomposition and approximation methods based on Artificial Intelligence (AI). We estimate the delivery costs of inserting a customer in each time slot through a Machine Learning (ML) regression model. For the customer's choice model, we take a significant leap forward in the state-of-the-art and model customer behavior at a granular level, i.e., for each customer. With the intent of shedding light on customer preferences, we propose two explainable artificial intelligence (XAI) approaches to model customer behavior: a Black-Box Model (BBM) followed by a feature contribution study; and an interpretable symbolic expression learned by Genetic Programming (GP). Additionally, we present a heuristic that combines the customer choice model with both the estimations of the delivery cost and occupation targets to prescribe a price panel for each incoming customer.

The remainder of this document is organized as follows. In Chapter 2 we describe the business context and the dynamic time slot pricing problem that will be faced by the considered retailer in the future. Chapter 3 details the main components of the solution approach that is proposed

to solve the dynamic problem faced by the retailer. This approach is composed of two predictive models and a prescriptive heuristic to decide the best time slot pricing panels to be shown. In Chapter 4, the results regarding the three components of the solution approach are presented. The explainability of the different models and algorithms is also discussed. In Chapter 5, we present the main conclusions derived from the research undertaken until this moment and provide an overview on future improvements and research ideas to be included in the current models and solution approach.

Chapter 2

Problem Description

In this chapter, we first present the business context and the challenge faced by the online retail partner considered in Section 2.1. Section 2.2 presents an illustrative example of the steps followed by a customer in the online platform, and the steps followed by the online retailer to prescribe time slot prices. In Section 2.3, we formally describe the problem tackled by the online retailer as a DTSPP and model it as a sequential decision problem, more specifically, a Markov Decision Process (MDP). Finally, in Section 2.4 we introduce the explainability challenges associated with the DTSPP we aim to solve.

2.1 Business context

The retail partner considered in this use case operates a set of physical retail stores geographically spread across the country. These physical stores are served by large distribution centers where products are prepared in pallets to be sent to the physical stores. At the same time, some of these retail stores and distribution centers are responsible for preparing online orders, placed in the website of the retailer. These online orders are later delivered to the customers by providing AHD services. To provide these services, the retailer manages logistic resources, such as a fleet of delivery vehicles and drivers, and defines the pricing decisions appearing in the website. In this business context, AHD services are still in their infancy. Therefore, retailers are still searching for better planning processes to further improve resource utilization and increase profits.

In the current approach adopted by the considered retailer, pricing decisions are subject to periodic reviews, where new pricing policies to guide time slot prices are formulated and tested directly in a real-world context. If the experiment yields satisfactory results from the perspective of the decision maker, the new pricing policies are implemented in the website. The ruling of the decision-maker is mostly empirical and fails to grasp the fine balance between customer satisfaction and the impact on fulfillment operations. Additionally, there is no risk nor sensitivity analysis as the monolithic process architecture requires a real-world experiment for the assessment of each scenario. An additional pitfall of the current approach is that it relies on a tactical perspective, disregarding the operational factor that is inherent to the online retail business. Hence, the time slot

offerings are not tailored to the profile of each customer, his/her buying behavior, or his/her current shopping basket, nor do they take into consideration the current or predicted load on the fulfillment operations or the cost to serve such customers. The aggregated effect of empirical decision-making allied with the non-existent operational perspective manifests in a lack of control over the selection of time slots by the customer pool, leading to upstream operational inefficiencies. Orders tend to be clustered around the preferred times of day, causing oscillating loads on warehouse and delivery operations. As the fulfillment capacity is mostly non-elastic, peak periods of demand tend to generate order backlog and inefficient delivery routes.

In the approach to be implemented in the future, time slot prices are dynamic. This means that they are computed every time a new customer enters the web site. To compute these dynamic time slot prices, the considered online retailer intends to analyze customer-related data, such as (1) the value of the shopping basket to be purchased; (2) the shopping profile of the customer according to past online purchases; (3) the preferred time slots; (4) the delivery fees charged in the past; and (5) the current state of the fulfillment capacity according to the orders that were already accepted. These data will then be used to decide upon the best time slot pricing panel to be shown to the customer. By tackling the problem from an operational dynamic pricing perspective, the trade-off between customer service / satisfaction and delivery efficiency can be accurately modeled.

Instead of pursuing simplistic tactical rules that fail to cater the needs of each customer, the proposed dynamic time slot pricing policies are aimed at steering customer choices. The objective is to nudge customers into selecting time slots that are beneficial both in terms of delivery operations (ex: inducing vehicle routes with good time window slacks) and in terms of profit (ex: by charging higher prices in preferred time slots), bearing in mind that delivery fees influence the probability of each customer to reject the proposed offers and walk-away from the website.

2.2 Illustrative example

The dynamic time slot pricing approach to be implemented maps into a new problem to be solved by the considered online retailer. Before formalizing this new problem as a sequential decision problem, we present Figure 2.1 to illustrate the perspectives of the online retailer and customers.

Retailer The online retailer receives customer arrivals at its website during a booking horizon (ex: several days). At the checkout of each customer, the retailer finds a decision point where a price panel needs to presented. This price panel contains the prices for the delivery time slots that are open at that particular moment. This process continues until the booking horizon closes at a predefined cut-off time. This cut-off defines the moment at which a set of time slots related to a particular time period is closed (i.e., does not accept additional customer) and a delivery schedule needs to be defined by solving a VRPTW.

Customer Each customer first decides upon a set of products to be purchased, the shopping basket. At the checkout, the customer is shown a specific price panel containing all the time slots

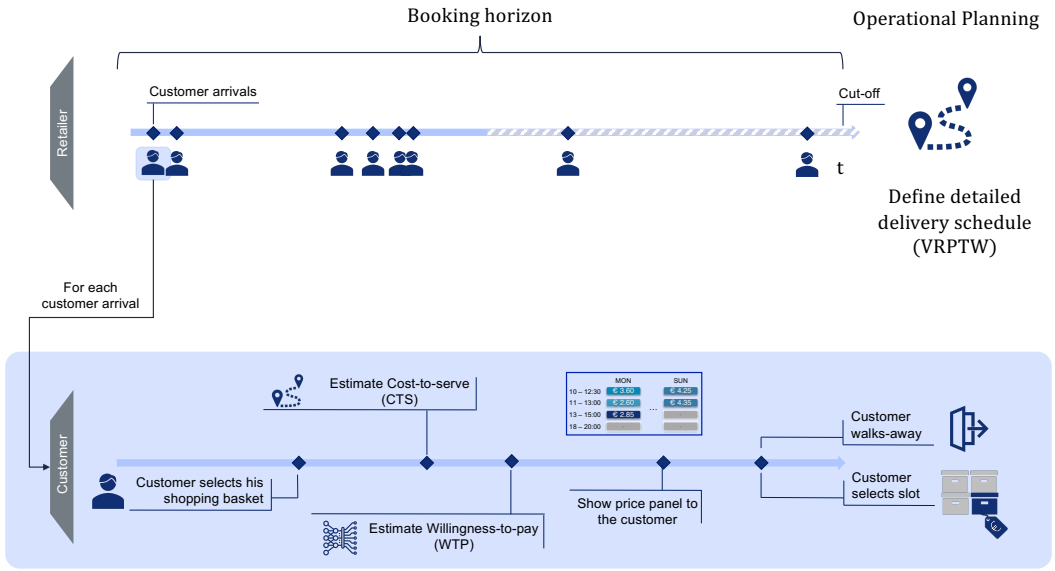

Figure 2.1: Overview of decision points encountered by the retailer and the customer during an online purchase.

available at the moment. The customer needs then to decide if he books an AHD to receive its shopping basket or if he walks-away without finalizing its purchase.

Note that, in the static approach, time slot prices were defined at a tactical level and remained unchanged for longer periods. In the new dynamic pricing paradigm, the online retailer is responsible for a decision point where it needs to compute a price panel to be shown. This process needs to be executed in fractions of a second, making it impossible to use complex mathematical solvers to solve the problem without any compromise or approximation. For that reason, the retailer needs to adopt solution methods that are capable of extracting pricing policies that can be executed in an online retail environment, within a very tight computational budget.

2.3 Formalization

The challenge faced by the online retailer considered in this use case is associated with a DTSPP. In this problem, an online retailer serves customers within a predefined delivery region, using a single depot. A unique delivery location is associated with each customer $i \in \mathscr{C} = \{1,...,C\}$, thus we use these terms interchangeably. During a certain booking horizon, customers place online orders dynamically and stochastically and, for each order, they book an AHD by choosing a specific time slot $w \in \Delta = \{1,...,W\}$ at a certain period $d \in \mathscr{D} = \{1,...,D\}$. This booking horizon closes at a so-called cut-off time and is followed by a service period in which the accepted customers will be visited by vehicle routes. These visits are to be performed within the time slot w and period d that was chosen by the customer, meaning that the vehicle must arrive at the customer between an earliest time a_{wd} and a latest time b_{wd}.

Let us now specify the components of the MDP that models the DTSPP as a sequential decision problem. We start by defining a decision epoch and the state variable. Then, we define the decision variables, exogenous information, transition function, and objective function (see [16]).

2.3.1 Decision epochs

A new decision epoch k begins whenever a new customer i arrives at the online platform of the retailer. Therefore, the number of decision epochs comprised in the considered MDP is equal to the number of customers C.

2.3.2 States

A state S_k describes all the relevant information at the beginning of decision epoch k, just before a new customer arrival. This information is used to compute state transitions, decisions, and contributions. Therefore, in the considered DTSPP, a state comprises information about the customers that have been revealed until decision epoch k, and their choices regarding time slots booked or walk-away events. Moreover, the state comprises additional information regarding the state of the system, namely, the current situation in terms of Advanced-Booking Time (ABT) curves for each time slot. These curves indicate the desired occupation level (a percentage of slot capacity) at each time instant between the opening and cutoff times of each time slot. For the sake of simplicity, we formally define these curves later in the document. Formally, we define the state as a vector $S_k = (S_{ik})_{C\times 1}$, with an entry S_{ik} dedicated to each customer. If the customer has not arrived yet or has arrived but did not book a time slot, $S_{ik} = (0,0)$. If the customer has already arrived and booked a time slot w on a certain period d, $S_{ik} = (w,d)$. At the initial state S_0, corresponding to the beginning of the booking horizon, $S_{ik} = (0,0)$ for every customer. Upon the arrival of a new customer, new information is revealed regarding the value of its shopping basket, geographical location, willingness-to-pay, and the price point $p \in \mathscr{P} = \{1,...,P\}$ charged for the chosen time slot.

2.3.3 Decisions

The online retailer makes decisions on the time slot price panels to be shown to each customer. A price panel characterized by a set of available time slots and their corresponding delivery fees. This price panel decision is dynamic, meaning that it will change over time depending on the current state variables and on the exogenous information revealed by the new customer arrival. More specifically, the online retailer needs to decide the delivery fee f_{wdk} for booking a delivery at time slot w in period d in decision epoch k. This fee can be chosen from the set of price points $\mathscr{P}$. Each price point is associated with a price π_p, hence $f_{wdk} \in \{\pi_0,...,\pi_P\}$. We assume that all price points are possible for all time slots. Let $\mathbf{f}_k = (f_{wdk})_{W\times D\times 1}$ be the matrix denoting the decisions made in decision epoch k regarding delivery fees of each time slot. Note that a customer can always choose to walk-away for free by selecting $(w,d) = (0,0)$ and this option is associated with a fee $f_{00k} = 0$

2.3.4 Transitions

Customers are known to have heterogeneous preferences concerning delivery time slots. The online retailer does not fully know these preferences, but they can be inferred from past online orders using a choice model to estimate the probabilities $\mathbb{P}_{wd}(\mathbf{f}_k)$ that a customer that just arrived chooses time slot w and period d. The probability for a customer to walk-away without booking a delivery is, therefore, given by $\mathbb{P}_{00}(\mathbf{f}_k) = 1 - \sum_{w \in \Delta, d \in \mathscr{D}} \mathbb{P}_{wd}(\mathbf{f}_k)$. The online retailer is able to manipulate this time slot choice probabilities according to its decisions regarding the price panels to be shown. Whenever a customer books an AHD, the retailer receives a contribution composed by the value of the shopping basket b_k purchased by the customer and a delivery fee f_{wdk}. Additionally, the retailer incurs in a delivery cost, as it is committed to serve the customer within the chosen time slot during the service period in which the vehicle routes will be executed. Here, the system transitions from state S_k to a new state that is defined with the operator $\Gamma(S_k, w, d)$, which updates the variables regarding the customer that arrived at decision epoch k, changing S_{ik} to (w,d). Adopting this notation, the transition to the next state is given by

$$S_{k+1} = \begin{cases} S_k, & \text{if the customer walks-away} \\ \Gamma(S_k, w, d), & \text{if the customer chooses time window } w \text{ on on } d \end{cases} \tag{2.1}$$

2.3.5 Contributions

In each decision epoch k, the online retailer receives a contribution that is computed considering the value of the basket b_k to be purchased by the new customer, the delivery fee that is charged for the chosen time slot (w', d'), and an opportunity cost $o_{wdk}(S_k)$ for booking time slot (w,d) in state S_k (i.e, an expected profit loss due to using the capacity of a time slot).

$$R_k = b_k + f_{w'd'k} - o_{w'd'k}(S_k) \tag{2.2}$$

The first two terms b_k and $f_{w'd'k}$ are determined straightforwardly, as the basket value is known immediately after the customer selects an item list and chooses a time slot for a price set by the retailer, respectively. The key factor to achieve the overarching profit maximization goal resides on the opportunity cost $o_{w'd'k}(S_k)$ estimation. Section 3 delves into the approach used to quantify this term, which involves estimating the customer's contribution to the final transportation costs, as well as assessing how his selection will let the real slot occupation curves approximate the tactically defined ones.

2.3.6 Objective

The objective function to be pursued by the online retailer is to maximize the total expected profit coming from following an optimal policy. This policy consists on choosing the best price panels to show for all customer arrivals during the booking horizon, and to optimally solve a VRPTW to

be executed during the service period. In each decision state S_k, the possible decision concerning delivery fees f_{wdk} are evaluated to solve the online pricing problem.

$$\max_{\mathbf{f}_k} \left\{ \sum_{w \in \Delta} \sum_{d \in \mathscr{D}} \mathbb{P}_{wd}(\mathbf{f}_k) \left(b_k + f_{wdk} - o_{wdk}(S_k) \right) \right\} \tag{2.3}$$

Note that to solve this problem, a suitable customer choice model must be specified and an optimization problem must be solved. Due to the large number of possible decision (and problems to solve), this dynamic problem must be solved using clever problem decomposition and approximation methods.

2.4 Explanations

The problem formalized in Section 2.3 can be solved in several steps by solving a set of subproblems. Depending on the solution methods used to solve it, different types of explanations can be used.

At the current state of the project, explanations are provided for a particular phase of the solution approach presented later in this document, which concerns to the prediction of the WTP of each customer. In this subproblem, which is a supervised machine learning problem, a decision function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is learned based on tabular training data with d dimensions, composed by pairs $(x_1, y_1, ..., x_n), y_n$. The training points are given by x_i, and the training labels are given by y_i.

In this use case, the prediction of the WTP is treated as a binary classification problem and is solved both through GP and Gradient Boosting Machine (GBM) techniques, that are described later in this document. These two approaches allow for different types of explanation and typically achieve different levels of performance. In this project, we rely on two different types of explainability, ad-hoc and post-hoc [3].

2.4.1 Ad-hoc explanation

When referring to ad-hoc explainability, we focus on explainable-by-design models, which, in the context of TRUST-AI, are symbolic expressions. These symbolic expressions can be interpreted by humans and are often represented in trees to facilitate their visualization. They are typically obtained through an evolutionary process and can consider several terminals (i.e., features) and mathematical and logical operators [11]. During the evolutionary process, individuals (i.e. symbolic expressions) are mutated or crossed-over iteratively to generate new individuals. In each generation, a selection procedure is performed in order to maintain the best individuals in the population (i.e, the ones with the best fitness) that will migrate to the next generation. the evolutionary process is stopped when certain stopping criteria are met (e.g. a maximum number of generations).

The symbolic expressions obtained by this process are typically considered explainable for three main reasons. First, they are seen as white-box models that can be easily inspected in terms of input and output variables. Second, decision makers and domain experts can be included in the evolutionary process, suggesting feature combinations that describe important business-related

issues. Third, the size of the models is controllable and can be used to obtain expressions with small sizes, which tend to be easier to understand and interpret.

Regarding the prediction of the WTP of each customer, we consider that the symbolic expressions generated by the adopted GP algorithms are explainable-by-design. The process of selecting the expressions that are better understood by the decision makers is the real explanation problem we face.

2.4.2 Post-hoc explanation

Post-hoc explanations are typically obtained by an explanation algorithm E that operates on the decision function f with the objective of explaining it. In local post-hoc explanations, the algorithm E is queried with a data point x and the corresponding decision y, and provides an explanation $E(x,y)$. This explanation should explain why decision function f made decision y for x. Commonly used post-hoc explanation algorithms are LIME, SHAP, and DiCE [17, 14, 15], which do not provide a global view on the decision function, but try to explain individual decisions $y = f(x)$.

To obtain explanations for the WTP binary classification problem, we resort to the techniques based on Shapley values (used in SHAP), which aim at quantifying the influence of each input-feature for a particular decision. The resulting explanations correspond to the linguistic form "The high price of this time window was relevant for the customer walk-away".

SHAP is based on additive feature attribution methods, which are based on an explanation model that is a linear combination of binary variables:

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i \tag{2.4}$$

where $z' \in \{0,1\}^M$, M is the number of simplified input feature, and $\phi_i \in \mathbb{R}$.

Methods with explanation models matching equation 2.4 attribute an effect ϕ_i to each feature, and summing the effects of all feature attributions approximates the output $f(x)$ of the original model. An attribute of the class of additive feature attribution methods is the presence of a single unique solution in this class with three desirable properties, namely. The first property, local accuracy, requires the explanation model to at least match the output of f for the simplified input x' (which corresponds to the original input x). The second property, missingness, requires features missing in the original input to have no impact, if the simplified inputs represent feature presence. The third property, consistency, ensures that if a model changes so that some simplified input's contribution increases or stays the same regardless of the other inputs, then that input's attribution should not decrease. Only one possible explanation model g follows equation 2.4 and satisfies the referred properties.

$$\phi_i(f,x) = \sum_{z' \subseteq x'} \frac{|z'|!(M-|z'|-1)!}{M!}[f_x(z') - f_x(z' \setminus i)] \tag{2.5}$$

where $|z'|$ is the number of non-zero entries in z', and $z' \subseteq x'$ represents all z' vectors where the non-zero entries are a subset of the non-zero entries in x'.

SHAP is a unified approach that improves previous methods, preventing them from unintentionally violating any of the three properties. It provides a unified measure of feature importance. These are the Shapley values of a conditional expectation function of the original model; thus, they are the solution to Equation 2.5.

Chapter 3

Solution Approach

The methodology developed to address the Dynamic Time Slot Pricing Problem (DTSPP) comprises three distinct modules. Section 3.1 describes how these are arranged to provide slot prices while taking into account the aforementioned problem trade-off. Then, sections 3.2 and 3.3 present the customer choice behavior model and the transportation cost estimation model, respectively. Finally, the procedure used to prescribe time slot prices is explored in Section 3.4.

3.1 Approach Overview

The overarching goal of the DTSPP is to prescribe time slot prices capable of maximizing profit while ensuring customer satisfaction and operational efficiency. Thus, while presenting a time slot price panel offering, we need to know how the customer will respond to it, i.e., which time slots will he be more willing to select or if there is an increased chance that he will withdraw the order. Simultaneously, anticipating the cost to deliver the customer order in each time slot is essential so that price can act as a mechanism to steer him way from selecting those more costly from an operational standpoint (or, conversely, nudge him into selecting those whose delivery would be more efficient).

The diagram presented in Figure 3.1 provides a scheme of the approach taken to generate prices weighing the service-efficiency trade-off. By considering the features of the time slot offering and the customer, as well as those pertaining to how he relates to each time slot option (e.g., historical percentage of orders where the customer selected a particular time slot), it is possible to model his choice behavior. Knowing the location of the customer and logistics constraints, e.g., geographical dispersion of previously accepted customers or the current occupation of each time slot, an estimation for the delivery cost can be obtained. Finally, using the output of these two models, the prescriptive heuristic can generate alternative time slot price panels and select the one whose contribution to the objective is more favorable.

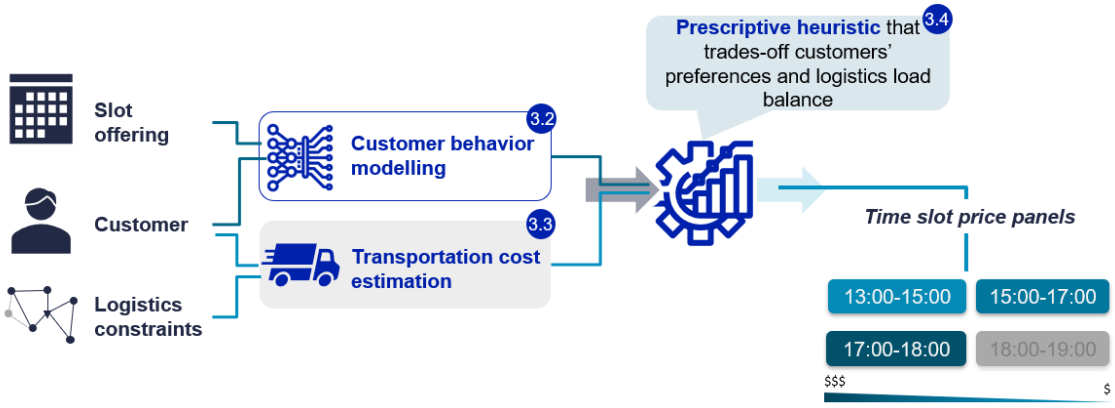

Figure 3.1: Overview of the solution approach for the DTSPP

3.2 Willingness To Pay Model

The Willingness To Pay (WTP) model consists of a customer choice behavior model whose goal is to anticipate customers' responses to different time slot price panels. In the context of the DTSPP, an approach capable of determining the customer's preference for each displayed slot or the withdrawal option as a function of the prices prescribed is desirable. Ultimately, we want to determine the selection probability of each option, and there should be coherence in terms of price sensitivity, i.e., increasing the price of a slot decreases its selection probability and leads to an increase in the average choice probability of competitor slots, and likewise for the opposite case.

3.2.1 Modeling Approach

The method used to obtain the WTP model was selected based on the data provided by the retailer. The analyzed dataset contained information pertaining to customers that confirmed their order by selecting a time slot. Customers that withdraw their orders after being presented a set of time slots are not tracked by the company. Therefore, it is not possible to infer the withdrawal probability based on the available data. To overcome this adversity, we estimate the walkaway probability using the WTP model.

3.2.1.1 Customer Selection Probability Modeling

To obtain the WTP model, we treated this problem as a binary classification problem. Therefore, our WTP model determines the selection probability for each $(customer, slot)$ pair. It could be argued that this problem could be modeled as a multi-label classification problem, where a score would be computed for each alternative time slot. However, due to the dynamic nature of the problem, the slots, and, therefore, the labels, would vary substantially as some slots would close due to capacity or cutoff times. Ultimately, the meaning of each label would depend on the instant the order was placed. Alternatively, a regression to predict the customer reservation price could be implemented. Nevertheless, historical data does not provide us with the reservation price as we do

not know whether the customer would still be willing to book the same slot for a higher price than the one accepted.

The algorithm selected to perform this predictive task was the gradient boosting machine (GBM). Gradient Boosting Machine (GBM) is a machine learning (ML) technique used for both regression and classification problems that has proven to outperform other approaches such as artificial neural networks in particle identification problems [6].

The performance metric used to assess the derived models was the log-loss. The log-loss is an error metric widely used in classification problems as it quantifies how far off are the prediction probabilities outputted by the model from the true value for the observation. Considering a dataset with N points, $\mathbf{p} = [p_i]_N$ to be the prediction probability of observation i belonging to class 1, and $\mathbf{y} = [y_i]_N$ the actual value of each observation, the log-loss is given by the expression described in equation (3.1).

$$log\text{-}loss = -\frac{1}{N} \cdot \sum_{i=1}^{N} [y_i \cdot ln\, p_i + (1 - y_i) \cdot ln\, (1 - p_i)] \tag{3.1}$$

3.2.1.2 Walkaway Probability Estimation

Another aspect of paramount importance when analyzing transitions between states is the likelihood of the occurrence of walkaways, as a fleeing customer entails a loss in revenue equal to the total basket value. In real terms, a customer may walkaway from an order when being provided with a price panel that surpasses his/hers willingness to pay (also referred to as reservation price) on all time slot options available.

Given that the retailer does not currently collect data on orders not finalized, we devised an approach for walkaway probability estimation that relies on the assumption that there is a given prevalence of walkaways (α^*) for the current pricing policy.

Borrowing inspiration from research on diffusion models for innovations, the walkaway probability is assumed to have an S-shape dependence with the maximum non-scaled probability of selection of a time slot belonging to a price panel [4]. In particular, it is assumed that this relationship follows the equation 3.2. The localization parameter c_1 controls the inflection point, while the shape parameter c_2 controls the first derivative, tied to the rate of decrease of the response variable.

$$w(x) = \frac{1}{1 + e^{-c_2 \cdot (x - c_1)}} \tag{3.2}$$

This approach has empirical backing on the principle of diminishing returns. The probability of a walkaway decreases as the maximum probability of selection increases. For very high time slot prices, all time slot options surpass the customer's reservation price, culminating in a walkaway probability that tends to 1. As the slot prices decrease and get closer to the customer's reservation price the rate of walkaway probability decreases at an ever-growing pace. The principle of diminishing returns governs slot price drops that go beyond the inflection point c_1. As slot prices tend to 0, the walkaway probability slowly progresses towards 0. This behavior is presented in Figure 3.2.

The challenge, then, becomes in determining the set of constants c_1 and c_2 that instantiate the generic equation 3.2 to a pre-determined global walk-away prevalence α^*. To this extent, we made use of Optuna, a flexible and scalable search framework that combines efficient parameter sampling and pruning to traverse the parameter search-space [1]. The goal is to explore different regions of the search space composed of the admissible ranges for c_1 and c_2, exploiting those that lead to the more promising fitness function values depicted in 3.3.

$$\min \left| \alpha^* - \frac{1}{N} \cdot \sum_{i=1}^{N} w_{c_1,c_2}(x_i) \right| \tag{3.3}$$

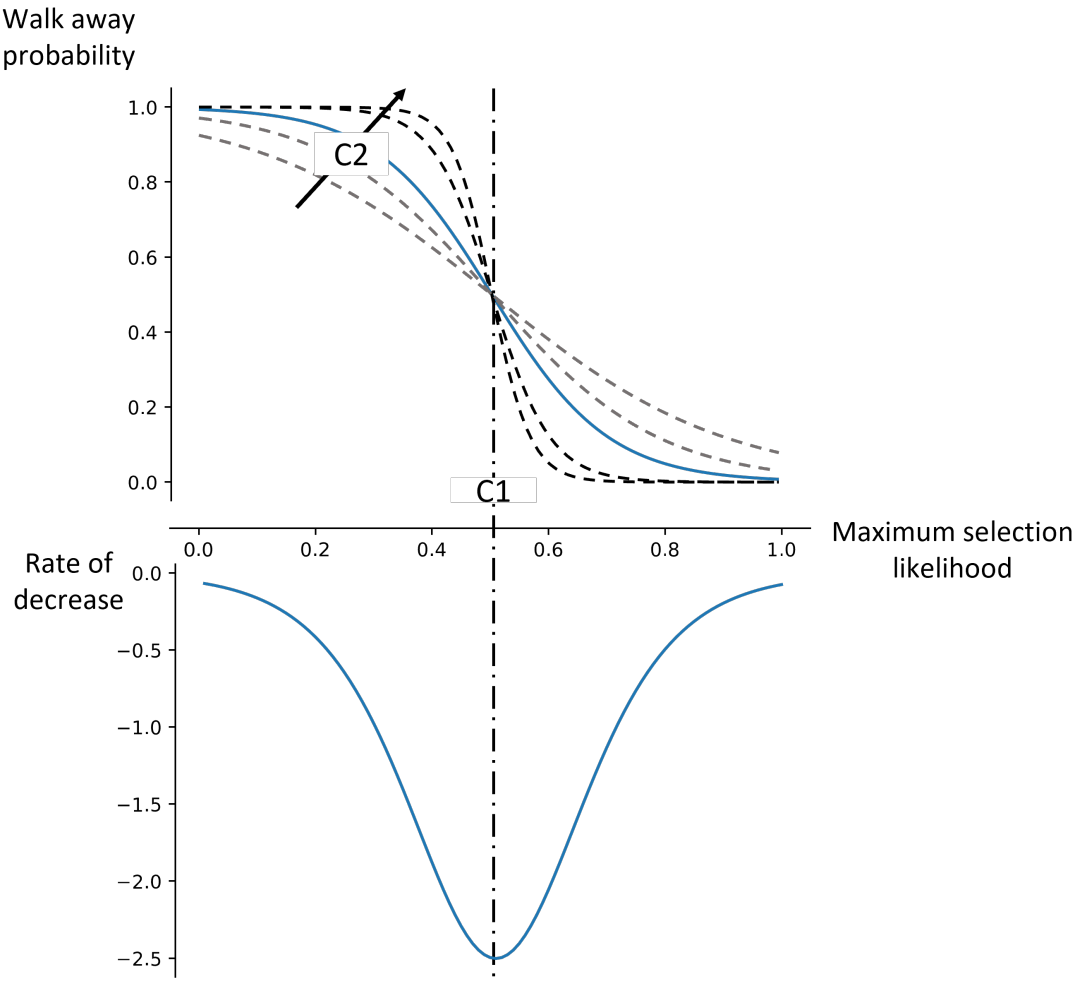

Figure 3.2: S-shaped evolution of walkaway probability (above) and rate of walkaway decrease (below) as a function of the maximum selection probability for a given panel

3.2.2 Feature Engineering

The DTSPP concerns pricing decisions at a customer level. To present a price panel tailored to each customer's preference, the WTP model should contemplate the features of the customers and their corresponding orders. Table 3.1 presents the features pertaining to these two partitions. Regarding

the characterization of customers, we extracted features indicating their longevity and loyalty concerning the AHD service, using, for example, the number of days since the first purchase and the number of purchases, respectively. In terms of the order itself, the focus centered on capturing the level of urgency or the absence of it, e.g., an order with a considerable portion of fresh produce might indicate urgency as the customer might want to receive them quickly as these are typically essential in a household.

Ultimately, the WTP model should provide the customer selection probability for each time slot that composes the panel. Therefore, it is important to provide features that might influence the utility attributed by the customer to each slot option. In Table 3.2, we include features intrinsically related to the slot (S), but also those capturing the relationship between a customer and a slot (CS). Regarding the first partition, beyond including the price of the slot, which is a decision variable of the DTSPP, we also add features mentioned in the literature as being important drivers of a customer preference in the context of AHD: service speed, measured by the distance between the moment of ordering and delivery (*slot_start*), and width. In terms of partition CS, we tried to capture customers typical selection behavior by determining if the day of the week of the slot under analysis is frequently chosen by them (*freq_slot_dow*), or by assessing the proportion of times that a customer selected a particular slot in the past (*exact_selection_customer_perc*).

We model the response of a customer to a given price panel as a binary classification problem. Thus, we characterize the relationship between pairs of customers and time slots. As a consequence, our methodology inherently disregards the existence of competition among time slots and the interdependent nature of the selection probabilities of the options in a given panel. To overcome this shorted-sighted approach, we created a feature partition that captures the price distribution of the whole price panel, P, which is detailed in Table 3.3. By including such features, we ensure that the derived WTP model deems that the selection probability of a given time slot is sensible to price changes in competitor slots.

3.2.3 XAI Approach

Although UC2's goal is to provide a *toy problem* and its corresponding solution approach, which will progressively integrate the TRUST framework, the WTP modeling problem was used as a testing tube to benchmark explainable artificial intelligence (XAI) techniques. More specifically, we tested two approaches: (1) a two-step approach in which, building on the Black-Box Model (BBM) generated by the GBM, we applied SHAP (SHapley Additive exPlanations) to build *a posteriori* explanations; and (2) learning an explainable-by-design symbolic model through regression using GP-GOMEA.

3.2.3.1 Post-hoc explainability approach

SHAP is a Python library that explains the output of ML models by using the concept of Shapley values from game theory [13]. In the context of ML, Shapley values determine a feature's contribution to the prediction made by the model. Summarily, the contribution of each feature is computed

Table 3.1: Customer (C) and order (O) feature sets for the WTP model

Feature	Description	Partition
npurchases	total number of past purchases made by the customer	C
expanding_freq_weekend_delivery	proportion of past purchases where the customer requested a delivery on a weekend	C
previous_weekend_delivery	flag indicating wether the last slot selection was on a weekend or not	C
expanding_avg_days_to_delivery	average number of days between delivery date and order date	C
freq_dow_selections	proportion of past slot selections made by the customer for a day of the week (dow) (where dow: Sunday - Saturday)	C
expanding_{measure}_slot_price	historical minimum, maximum and average delivery price paid by the customer	C
previous_slot_price	delivery price paid by the customer in his last purchase	C
first_online_purchase	flag indicating whether it is the first time the customer uses the AHD service	C
days_since_last_order	number of days elapsed since the last purchase made by that customer	C
days_since_first_purchase	longevity of the customer	C
days_between_purchase	average number of days between historical purchases made by the customer	C
total_last_purchase	total amount paid in the last purchase	C
expanding_total	sum of the total historical amount that a customer has spent up to that point	C
expanding_total_avg	average historical amount that a customer has spent per purchase up to that point	C
order_dow	order dow	O
total	total amount paid by the current shopping basket	O
nsku	number of individual articles in the current shopping basket	O
freshpercent	percentage of fresh products in the current shopping basket	O
discountpercent	discount percentage on the current bill	O
orderperiod	time of day of order placement, e.g., morning or afternoon	O
previous_requested_amount	total requested amount on the previous purchase made by the customer	O
expanding_successful_picking_amount	expanding sum of the successfully picked amount in purchases made by the customer up to that point	O
expanding_substitution_picking_amount	expanding sum of the substituted amount in purchases made by the customer up to that point	O
expanding_requested_amount	expanding sum of the historical requested amount made by a customer up to that point	O
expanding_successful_picking	average picking success rate (weighted in by amount) in the purchases that the customer had made up to that point	O
expanding_substitution_picking	average substitution rate (weighted in by amount) in the purchases that the customer had made up to that point	O
previous_successful_picking_amount	amount of the items successfully picked in the last purchase	O
previous_substitution_picking_amount	amount of the items substituted in the last purchase	O

Table 3.2: Slot and customer-slot relationship feature sets for the WTP model

Feature	Description	Partition
slotcost	the slot price presented to the customer	S
slot_dow	the delivery dow (0: Sunday - 6: Saturday)	S
slot_start	the start of the time slot in minutes since the moment of ordering	S
slot_end	the end of the time slot in minutes since the moment of ordering	S
slot_start_w	the start of the time slot in minutes since the start of the week	S
slot_end_w	the end of the time slot in minutes since the start of the week	S
slot_width	the slot width in minutes	S
weekend_delivery	flag indicating whether the slot dow is on a weekend	S
slot_time	slot start and finish times in the following format 'start_time - end_time', e.g., '14:00 - 18:00'	S
partial_selection_customer_perc	proportion of past slot selections intersecting with the dow and time frame of the slot	CS
exact_selection_customer_perc	proportion of past slot selections that exactly match the dow and time frame of the slot	CS
freq_slot_dow	proportion of past slot selections on the same dow as the slot's dow	CS
preferred_slot_dow	flag indicating whether the current slot dow is the one the customer historically prefers	CS

Table 3.3: Price panel feature set for the WTP model

Feature	Description	Partition
min_cost	minimum slot price presented to the customer on the current order	P
q1_cost	first quartile of slot price presented to the customer on the current order	P
median_cost	median slot price presented to the customer on the current order	P
q3_cost	third quartile of slot price presented to the customer on the current order	P
max_cost	maximum slot price presented to the customer on the current order	P
cv_cost	coefficient of variation in the slot prices show to the customer in the current order	P
iqr_cost	interquartile range of the slot prices presented to the customer on the current order	P

by comparing the original predictions for each instance by the predictions made on instances where the feature is replaced by its mean value [18].

By relying on domain knowledge, it is possible to state that a model determining the selection probability of a time slot by a customer should be highly dependent on its price and time slot attributes. For instance, slot width is a determining factor since the customer will have a more precise estimate of the time of delivery for narrower time slots. Additionally, features such as distance to the beginning of the time slot, i.e., delivery speed, and the number of displayed time slots, i.e., availability, are also important drivers for selection [2]. Furthermore, the past relationship between the customer and the time slot is essential as the selection probability should be higher if the customer selected the time slot in the past. To assess the importance of these and the remaining features used by our model, we based our analysis on the SHAP bar plot. Figure 3.3 displays an example plot for an ad-clicking prediction problem where it is possible to observe that features are ranked according to their importance, i.e., their mean absolute SHAP value.

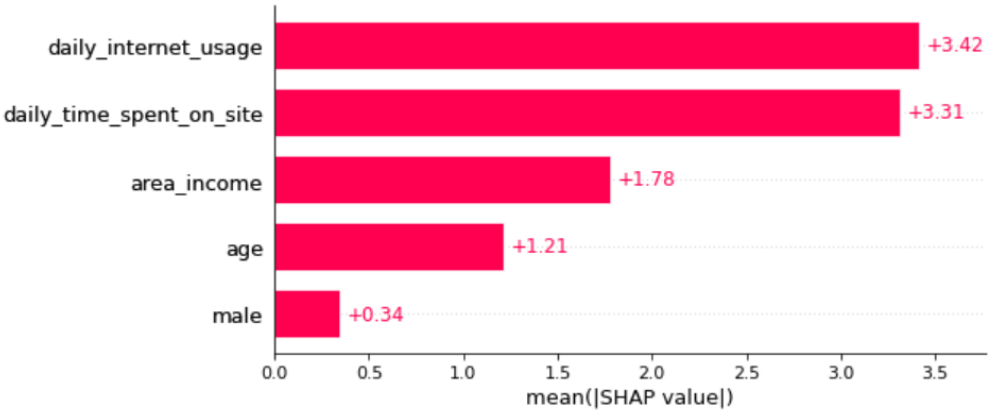

Figure 3.3: Example of a SHAP bar plot for an ad-clicking prediction problem

As explained in 3.1, price is an important feature in the context DTSPP. To guide our pricing heuristic, it is desirable that our WTP model is sensible to price and that its impact is coherent on selection probability variations, i.e., positive price shifts decrease the selection probability of a time slot, vice-versa. With the purpose of assessing the consistency of our models, we plotted the SHAP summary graph to visualize the effect of each feature over the dataset. Using the same ad-clicking prediction problem as an example, Figure 3.4 presents the respective SHAP summary plot.

Similarly to the bar plot, the SHAP summary plot also orders the features in terms of decreasing importance. For each feature, instances are plotted on the x axis according to their SHAP value and color grading based on their feature value, i.e., lower values assume blueish colors, while higher values tend to a purplish color. Focusing on the most important feature, as computed by SHAP, it is possible to observe that users who consume more internet on a daily basis are less likely to click on an add.

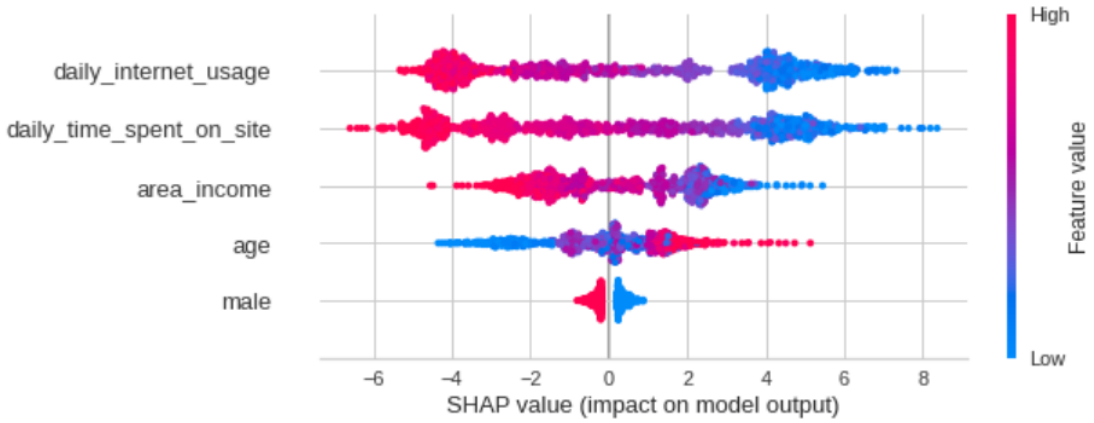

Figure 3.4: Example of a SHAP summary plot for an ad-clicking prediction problem

3.2.3.2 Ad-hoc explainability approach

With the purpose of presenting an alternative to the traditional post-hoc explainability approach, we set out to model customer selection behavior using explainable-by-design symbolic expressions. Centering decisions and explanations on the same model allows the decision-maker to directly interpret and adjust the model in an iterative way as it is learned, i.e., achieve human-guided symbolic learning [7].

The analytical WTP expression was derived through symbolic regression using GP-GOMEA [20]. GP-GOMEA consists of a Genetic Programming (GP) algorithm that applies the Gene-pool Optimal Mixing Evolutionary Algorithm (GOMEA), a model-based EA. The distinctive element of GP-GOMEA relative to standard GP is a variation operator capable of exploiting linkage models by combining partial solutions [19].

Performing symbolic regression entails searching the space of mathematical expressions that can be formed using a primitive set $\mathscr{P}$ encompassing a group of functions and variables, to minimize a loss function $\mathscr{L}$. The set of functions Ω could be composed of basic algebraic operators, e.g., addition, subtraction, multiplication, or division. Additionally, a set Φ of relevant problem features, such as x_1, x_2, and a set of constants Γ could define the group of variables. A special concept used in GP literature to represent probability distributions from which constants can be sampled are *ephemeral random constants*. For this particular case, we could consider operator $\mathscr{R}$ that samples any real number in $\mathbb{R}$. As a result, the primitive set would be given by $\mathscr{P} = \{\Omega \cup \Phi \cup \Gamma\} = \{+, -, \times, /, x_1, x_2, \mathscr{R}\}$. Therefore, the search space $\mathscr{F}$ is composed of mathematical functions that can be expressed using any combination of primitive set elements, while ensuring arity and variable type constraints. The symbolic regression problem can be expressed by equation (3.4) [21].

$$f^{gp} = \underset{f \in \mathscr{F}}{arg\,\min}\, \mathscr{L}\left(\boldsymbol{y}, f\left(\boldsymbol{X}\right)\right) \tag{3.4}$$

In the context of the customer choice model estimation, $\boldsymbol{X}$ denotes a matrix of n historical data on *time slot, customer* combinations with dimension $n \times |\Phi|$. Vector $\boldsymbol{y}$, with dimension $1 \times n$, indicates the outcome of such combination, i.e., whether the customer selected the slot or not, which is a Boolean variable. As any solution approach for optimization problems, GP-GOMEA evolves better models when the loss function is continuous (reference?). Thus, we defined f to be a symbolic expression with codomain $\mathbb{R}$, which allows $\mathscr{L}$ to be the log-loss function and to benchmark GP-GOMEA against the GBM. Alternatively, we could have defined f to also output Boolean variables. In that case, the fitness function would need to be updated by a measure assessing the degree of dissimilarity between $\boldsymbol{y}$ and $f(\boldsymbol{X})$, e.g., the Hamming distance.

To minimize the log-loss, GP-GOMEA needs to evolve models that provide selection probabilities for each combination as close to the actual outcome. Therefore, its codomain should lie within the $[0,1]$ interval. There are two approaches that could be deployed to achieve this:

1. Let GP-GOMEA learn functions whose output lies within the aforementioned interval;
2. Apply a sigmoid or logistic regression transformation to the models evolved by GP-GOMEA to ensure that an initial codomain in $\mathbb{R}$ is transformed in one in $[0,1]$.

Since learning this constraint using GP would be computationally expensive, we adopted the second strategy. Therefore, considering f^{gp} to be a model learned through GP-GOMEA, our symbolic expressions f are given by equation (3.5).

$$f^{wtp} = \frac{1}{1+\exp(-f^{gp})} \tag{3.5}$$

With respect to interpretability, applying the sigmoid transformation does not compromise model interpretability. In fact, through equation (3.5) it is possible to observe that $f^{gp} \propto f^{wtp}$. Therefore, the evolved symbolic expressions will be proportional to the selection probability.

3.3 Cost-To-Serve Model

The clever problem decomposition alluded to in 2.3.6 and detailed in 3.1 also entails obtaining accurate estimates of the delivery cost for each incoming order, at the moment it is placed. The minimization of delivery costs, to a certain extent, contributes to the overarching goals of achieving operational efficiency and maximizing profit. The Cost To Serve (CTS) model is served with the task of providing the prescribing heuristic with these antecipated delivery cost estimates.

3.3.1 Modeling Approach

In academia, most delivery cost estimation methodologies rely on explicit routing decisions, through some heuristic approach to the Vehicle Routing Problem With Time Windows (VRPTW). Alternative approaches include seed-based approximations where the routing costs are subdivided into two separate components: depot-to-seed and seed-to-customer costs.

Recent research has presented a third and novel alternative by leveraging Machine Learning (ML) regression models. It entails engineering a set of regressors x_i, that fully describe the state of the system, and that are then used to predict the response variable - the final delivery cost. The regression model's objetive is to estimate feature coefficients β that minimize deviations (e.g. quadratic error) from the actual values y:

$$\min_{\beta} \sum_{k=1}^{N} \left(y^k - \left(\beta_0 + \sum_{i=1}^{I} \beta_i x_i^k \right) \right)^2 \tag{3.6}$$

Borrowing from this approach, we propose a CTS model that trains on instances that describe the state of the system at every point in time that an order was placed. The discussion of the features that provide a complete view of the system is brought along in subsection 3.3.2.

As previously pointed out, demand for each slot-area is indivisible and always fulfilled by the same depot. Thus, the state of the system is mostly defined within the boundaries of each individual area. Delivery routes may traverse area boundaries for contiguous areas allocated to the same depot, but they never include customers placed in areas allocated to different depots.

For modeling purposes, given that demand is indivisible among depots and areas (both categorical variables) it is expected that categorical variables will play an important role in obtaining accurate predictions. With a native ability to preprocess categorical variables, even those with high cardinality that are expected to dominate the outcomes in this use case, the Yandex developed Catboost regression algorithm has demonstrated paramount performance in similar use cases. Thus, the CTS ML-based approach will rely on a CatBoost instance, subject to hyperparameter optimization and trained with a randomized cross-fold validation strategy.

3.3.2 Feature Engineering

Recall from 2.3 that customers are associated with unique delivery locations $i \in \mathscr{C} = \{1,...,C\}$. The total area of operation of the retailer is subdivided into abstract but permanent boundaries that constitute depot's area of influence. Delivery routes are designed only within each of these smaller spatial areas, from now on referred to as delivery areas $a \in \mathscr{A} = \{1,...,A\}$. As time slot offerings for a given delivery area contain only time slots with non overlapping service times, we can define delivery area - time slot combinations (ATCs).

For each of these combinations, we aggregate customer and routing information. A comprehensive list of the regressors is provided in Table 3.4.

Besides contextual information, we aimed to capture the dispersion of the delivery locations of the orders allocated to each ATC by measuring the relative positioning of the already allocated customers and the serving depot. Prior knowledge regarding past delivery costs to each customer is also included. Since larger orders are associated with higher unloading times that tend to drive up delivery costs, the feature space is also broaden with some shopping basket information.

Each incoming order contributes information towards the final state of the system. Yet, at any point in the booking horizon prior to the time slot cutoff, the regression model will have

Table 3.4: Feature set for the CTS model

Feature	**Feature name**	**Data partition**
Average customer haversine distance	average haversine distance between customers already allocated to a ATC	ATC
Customer centroid haversine distance to depot	haversine distance between the ATC customer centroid and the fulfilling depot	ATC
Number of customers	number of customers already allocated to an ATC	ATC
Average customer-depot bearing	the average of all bearings between each already allocated customer and the serving depot	ATC
Standard deviation of the customer-depot bearing	the standard deviation of all bearings between each already allocated customer and the serving depot	ATC
Time to service period start	time elapsed between the order timestamp and the start of the service period	T
Delivery day of week	the service period's day of week	T
Store identifier	the store identifier	D
Area indentifier	the area indentifier	A
Number of unique items	the number of unique items in the shopping basket	O
Total number of items	the total number of items in the shopping basket	O
Basket value	overall shopping basket value	O
Previous delivery cost	delivery cost of the previous delivery to the same customer	C
Average time stopped	average time spent unloading in previous deliveries to the same customer	C
Average past delivery cost	the average cost of all previous deliveries to the same customer	C
ATC occupation	the atc occupation	A
Depot occupation	the depot occupation	D
Customer's first order	boolean indicating whether the current record is the customer's first order	C

an incomplete view of the final system. Features like time to service period start help with contextualizing the gap between the current prediction and the future moment when all routing information is known.

Finally, ATC occupation and overall depot occupation may provide insight into a stressed or congested operation.

3.4 Prescriptive Heuristic

The developed prescriptive heuristic is to be applied in a dynamic setting. Therefore, its solutions will respond to the sequential decision problems arising in each epoch. Thus, to assess the performance of our solution approach, the whole booking period needs to be simulated. In section 3.4.1 we start by describing the approach taken to perform the simulation. Section 3.4.2 details the structure used to represent decisions. The fitness function is defined in Section 3.4.3. Finally, Section 3.4.4 discusses the procedure to generate time slot price panels.

3.4.1 Simulation Configuration

In dynamic problems, the appropriateness of a given solution approach can only be assessed after simulating the whole decision horizon. Therefore, we modeled the booking horizon dynamics of the DTSPP as described in Algorithm 1. Before the beginning of the booking horizon, we define problem entities such as the set of available time slots, Δ, and of incoming customers, $\mathscr{C}$. Additionally, the weights of each problem objective need to be provided. Let w_ϕ and w_ω denote the weights attributed to profit rewards and occupation targets, respectively. Moreover, the set of parameters of the prescriptive heuristic, Φ, needs to be provided.

Algorithm 1: *simulateBookingHorizon*

Result: S_T
Data: Δ, $\mathscr{C}$, w_ϕ, w_ω, Φ
1 **for** $t_k \in T$ **do**
2 $i \leftarrow C_k^{new}$;
3 Determine $\Delta^i \subseteq \Delta$, the subset of available time slots for customer i;
4 Load the base price panel $\mathbf{f}_k^b$ generated by the retailer through its tactical pricing strategy;
5 $\mathbf{f}_k \leftarrow generatePricePanel\left(\mathbf{f}_k^b, \Phi\right)$;
6
7 **if** $rand() \geq walkawayProbability\left(\boldsymbol{f}_k, i\right)$ **then**
8 Selected time slot (w, d) obtained by sampling one element from the pdf $\mathbb{P}_{wd}\left(\mathbf{f}_k\right)$;
9 $S_{k+1} = \Gamma\left(S_k, w, d\right)$;
10 **else**
11 Customer withdraws order;
12 $S_{k+1} = S_k$;

For each time instant t_k of a decision epoch k, a customer arrives and the set of available time slots, Δ^i, is determined, as indicated in lines 2 and 3. Determining set Δ^i entails not only filtering

those time slots that were tactically made available to his/her delivery area, as well as those currently open. A slot is deemed open if the customer arrives after its opening time and before its cutoff, and if its occupation has not yet reached the capacity. Then, taking as an initial solution the price panel generated by the retailer's tactical pricing approach, denoted as $\mathbf{f}_k^b$, the pricing heuristic is applied to refine it, as described in lines 4 and 5. From line 7 to line 12, the slot selection process is simulated using the customer choice models developed in Section 3.2 and the system state, S_{k+1}, is updated accordingly.

3.4.2 Solution Representation

The pricing heuristic is applied in each decision epoch k to perform the corresponding decision x_k. This decision consists of a time slot price panel, denoted as $\mathbf{f}_k = [f_{wdk}]_{W \times D \times 1}$. Therefore, we represent solutions as a bi-dimensional $W \times D$ matrix as illustrated by Figure 3.5.

$\mathbf{f}_k$	1	...	d	...	D
1	f_{11k}	...	f_{1dk}	...	f_{1Dk}
...	...	...	...	...	...
w	f_{w1k}	...	f_{wdk}	...	f_{wDk}
...	...	...	...	...	...
W	f_{W1k}	...	f_{Wdk}	...	f_{WDk}

Figure 3.5: Solution representation as a $W \times D$ matrix

3.4.3 Solution Evaluation

In sequential decision problems, estimating the opportunity cost of current decisions is of uttermost importance. In the context of the DTSPP, even though we are dealing with a stochastic state transitions, as these are determined by a customer choice behavior model, we want to anticipate the future impact of letting the customer use one unit of the capacity of each time slot. In practice, this opportunity cost will represent the potential lost sales of future that was seeking the particular slot but was unable to due to capacity restrictions. Alternatively, this cost component can also represent the added transportation cost resulting from the fact that future demand for that slot is placed in distant locations from the customer.

The expression used to evaluate the fitness of each candidate solution seeks to capture the opportunity cost component, beyond the immediate reward. Therefore, we consider a so-called immediate reward $\phi(\mathbf{f}_k)$ that determines the expected profit from applying price panel $\mathbf{f}_k$, further detailed in Section 3.4.3.1. In Section 3.4.3.2, we will introduce another component of the fitness function that encapsulates the opportunity cost and ensures the solution approach is not greedy and consumes the capacity of the most attractive slots early on the booking horizon. This component, the occupation target, $\omega(\mathbf{f}_k)$, measures how will the current price panel bring the system occupation curves approximate the desired Advanced-Booking Time (ABT) curves. Expression 3.7 presents

fitness function w, which combines the aforementioned two objectives in a single expression weighted by their corresponding weight coefficients.

$$w(\mathbf{f}_k) = w_\phi \cdot \phi(\mathbf{f}_k) + w_\omega \cdot \omega(\mathbf{f}_k)\ ,\ w_\phi, w_\omega \in \mathbb{R}_0^+ \tag{3.7}$$

3.4.3.1 Immediate Reward

The immediate reward component $\phi\left(\mathbf{f}_k^b\right)$ calculates the expected profit associated with serving customer i. Using the WTP model described in Section 3.2 to obtain selection probabilities $\mathbb{P}_{wd}(\mathbf{f}_k)$, and the CTS model to obtain an anticipated delivery cost-to-serve estimate, c_{wdk}, customer i in time slot (w, d), the immediate reward can be computed under expression 3.8.

$$\phi(\mathbf{f}_k) = \sum_{w \in \Delta} \sum_{d \in \mathscr{D}} \mathbb{P}_{wd}(\mathbf{f}_k) \cdot [b_k + f_{wdk} - c_{wdk}] \tag{3.8}$$

3.4.3.2 Occupation Target

Our approach to model the cost-to-serve uses information on the system state S_k to anticipate the contribution of the customer i to the final transportation cost. However, this cost estimate does not provide a complete perspective of the opportunity cost of letting customer i choose a given time slot. More specifically, it does not quantify the potential future revenue loss of letting go a customer j with a high basket value, b_j due to a lack of capacity to serve them in that same time slot. Therefore, using solely the immediate reward component may lead to an overly greedy solution approach that would try to steer customer choices to more profitable scenarios. As a consequence, the most attractive time slots would close early on the booking period due to capacity constraints as our heuristic would incentive their choice as they are safe choice to secure the customers basket value and avoid a withdrawal.

To avoid developing a greedy solution approach, we adopted the concept of Advanced-Booking Time (ABT) curves from the revenue management literature [22]. In the context of the DTSPP, we define the ABT curve as a function $l^*_{wd}(t)$ that states the desired occupation level (as a percentage of slot capacity) at each time instant between the opening and cutoff times of each time slot (w, d). The ABT curves provide information to steer pricing decisions towards bringing the effective slot occupation level $l^e_{wd}(t)$ closer to the levels defined before the beginning of the booking period. Figure 3.6 presents a visualization of the mechanism introduced by the ABT curves. In region (a), the real occupation level of the slot is higher than the ABT. Thus, the prescribing heuristic will generate higher delivery fees or decrease the average prices for alternative options to discourage the customer from selecting the slot. In region (b), the opposite effect happens, as the occupation level needs to keep up with the ABT curve.

Our solution approach needs to guarantee that all slot occupation curves approximate the defined target ABT curves. In other words, we want to minimize the distance of both curves over time across all slots. Therefore, the concept of the Mean Absolute Percentage Error (MAPE) can be adapted from the time-series forecasting literature to provide a distance measure to evaluate the

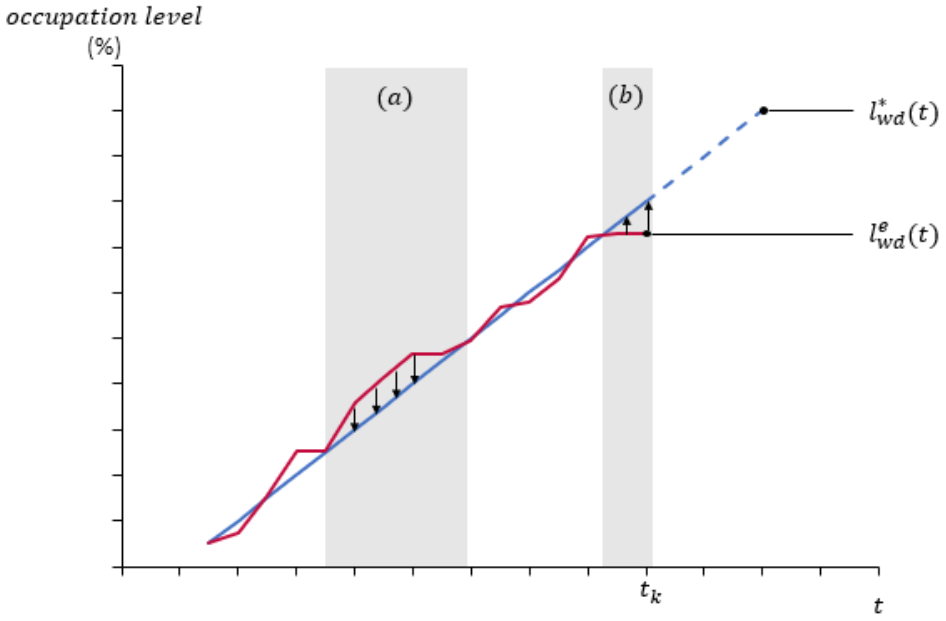

Figure 3.6: Illustration of the effect induced by the ABT curves approximation

degree of curve approximation. Since it is desirable to have a measure that is sensible to the prices generated, we calculate the overall MAPE of the expected occupation resulting from applying a price panel $\mathbf{f}_k$. Equation 3.9 presents the expression for the distance measure used. Focusing on a particular time slot (w, d) and epoch k, we compute the absolute percentage error in between the expected occupation caused by decision $\mathbf{f}_k$, given by $l^e_{wd}(t_k) + \mathbb{P}_{wd}(\mathbf{f}_k)$, and the target level occupation, $l^*_{wd}(t_k)$.

$$\omega(\mathbf{f}_k) = -\frac{1}{W \cdot D} \cdot \sum_{w \in \Delta} \sum_{d \in \mathscr{D}} \frac{\left| \left[l^e_{wd}(t_k) + \mathbb{P}_{wd}(\mathbf{f}_k) \right] - l^*_{wd}(t_k) \right|}{l^*_{wd}(t_k)} \tag{3.9}$$

Since the DTSPP deals with a maximization objective, we define the occupation target component of the objective function as a negation of the MAPE of the expected occupation.

3.4.4 Probability-Guided Prescriptive Heuristic

Beyond responding to the trade-off imposed by the DTSPP, the solution approach must be efficient as the available computational budget is reduced. Indeed, in the context of online retailing, after selecting their shopping basket, the customer should be presented with a time slot offering in real time, often to a fraction of a second. Thus, we develop a so-called Probability-Guided Prescriptive Heuristic (PGPH) to perform price adjustments to the initial time slot price panel computed by the retailer. Briefly, this procedure improves the initial solution by reducing the search space under two strategies: (1) focusing on the price of the top τ time slots with respect to their likelihood of selection; and (2) performing a search over sets of discrete price points. The rationale behind this improvement heuristic consisted in taking advantage of the reduced time budget to

explore variations in slot options with a reasonable chance of being considered by the customer. Furthermore, exploring discrete sets of price points offers the decision-maker a more actionable tool to perform pricing decisions. In fact, he/she can decide the level of granularity of the prices explored by setting the unit price shift parameter δ. Algorithm 2 describes the procedure used to perform pricing decisions.

Algorithm 2: *generatePricePanel*

Result: $\mathbf{f}_k$
Data: $\mathbf{f}_k^b, \Phi = \{n, \tau, \pi^-, \delta, \pi^+\}$
13 Compute the selection probabilities for the base price panel $\mathbf{f}_k^b$;
14 Order the available slots (w,d) in descending order of $\mathbb{P}_{wd}\left(\mathbf{f}_k^b\right)$;
15 Determine set $\Delta^* \subseteq \Delta^i$ composed by the top min $\left(\tau; |\mathbf{f}_k^b|\right)$ slots;
16
17 **for** $(w,d) \in \Delta^*$ **do**
18 Determine $\mathscr{P}_{wd}$ the set of price points that result from applying δ m.u. shifts to base price f_{wdk}^b, such that $\pi^- \leq f_{wdk} \leq \pi^+$;
19 Generate $\mathbf{F}_k$ a set of n price panels by sampling random prices from sets $\mathscr{P}_{wd}$ where each element $\mathbf{f}_k'$ respects the following condition:
20 $\mathbf{F}_k = \{\mathbf{f}_k' : f_{wdk}' \in \mathscr{P}_{wd}, \forall\,(w,d) \in \Delta^* \vee f_{wdk}^b, \forall\,(w,d) \notin \Delta^*\}$;
21 $\mathbf{f}_k \leftarrow \underset{\mathbf{f}_k' \in \mathbf{F}_k}{arg\max}\ w\left(\mathbf{f}_k'\right)$;

The PGPH can be regarded as a local search operator that performs a n-tournament selection under a neighborhood structure composed of neighbor price panels resulting from the simultaneous price variations for τ time slots. From lines 13 to 15, the *generatePricePanel* procedure starts by applying the WTP model to the base price panel and selects the top τ time slots in terms of the highest selection probability. Alternatively, if the number of available slots is lower than τ, all slots are chosen. Then, line 18 determines, for each time slot (w,d), the set $\mathscr{P}_{wd}$ composed of the discrete price points within the interval $[\pi^-;\pi^+]$. These price levels are obtained by multiplying the unit price shift, δ, by k, where $k \in \mathbb{Z}$. Finally, from lines 19 to 21, using the feasible price points for each time slot in Δ^*, the heuristic generates n random price panels and selects the one that maximizes fitness function w.

Chapter 4

Preliminary Results

This chapter presents the performance of the models described in Chapter 3, as well as explainability discussions for the Willingness To Pay (WTP) modeling task in particular. The results will be presented using the same order as the one applied in Chapter 3: Section 4.1 will discuss the WTP model; afterwards, the Cost To Serve (CTS) model will be explored in Section 4.2; finally, Section 4.3 evaluates the Prescriptive Heuristic. In each section, the data used to train models or the instances under optimization will be discussed before exploring the achieved results.

4.1 WTP Results

This section intends to analyze the performance and validate the outputs of the trained customer choice behavior models. First, in Section 4.1.1, we present a description of the data provided by our retail partner, as well as the assumptions taken to define the scope of the WTP model. Second, Section 4.1.2 provides performance results for the Gradient Boosting Machine (GBM) model and, afterwards, in Section 4.1.3, we benchmark it again the symbolic expression generated by GP-GOMEA. This benchmark entails performance comparisons as well as an analysis of the comparative interpretability between both methods.

4.1.1 Data

The dataset disclosed by our retail partner contains information pertaining to online orders over the course of a year, from October 2016 to September 2017. As previously mentioned, since the retailer only keeps track of confirmed orders and the booked time slot, the time slot price panel presented to the customer is not known, as well as the information of lost sales due to walkaways. To overcome the absence of walkaways, we adopted the approach described in Section 3.2.1.2, which provides a price panel-sensitive walkaway probability estimation model that can be trained to offer an overall walkaway percentage for the current tactical pricing strategy set by the decision maker, e.g., 5% of the customers do not confirm their order after selecting the basket value. To obtain the alternative slots seen by each historical order, we applied a reverse engineering procedure similar to the one presented by [2]. This procedure consists of first estimating the capacity of each time slot. Then,

the set of historical orders is analyzed, from the most recent one to the oldest, and the capacity of each slot is deducted each time an order selects it. It is considered that the slot was open while the available capacity is larger than zero. With this procedure, for each historical order, the set of alternative slots becomes known. The price at which the time slot was presented is given by the average price displayed to orders that selected the slot in time instants close to the time at which the order was placed.

Beyond the application of this two strategies to facilitate modeling customer choice behavior, we took some assumptions that impact the simulation environment:

In-store pickups Our retail partner offers not only Attended Home Delivery (AHD) services, but also allows the customer to select a time slot to pick his order at its physical stores. As this service offering deviates from the scope of the Dynamic Time Slot Pricing Problem (DTSPP), since it does not require routing optimization, we discarded data for orders asking for such a service and we do not consider them in our simulation.

Free-shipping service For a fixed monthly price, the retailer lets customers order without a delivery fee. Similarly to in-store pickups, we also disregard these customers due to the fact that there are no pricing decisions involved when it comes to addressing them, as they can book any time slot free of charge.

Slot visibility The retailer offers a set of time slots up to 60 days into the future. Since more than 98% of the orders booked a time slot within 7 days into the future, we limit the set of alternative slots based on that time frame.

4.1.2 Performance Discussion

This section analyzes the performance of the two alternative WTP models developed. We start by presenting the results for the black-box machine learning model and the GP-GOMEA-generated symbolic expression separately in Sections 4.1.2.1 and 4.1.2.2. Then, we benchmark the two models in terms of their applicability to model the WTP in Section 4.1.2.3.

4.1.2.1 Black-box model

The process of obtaining a black-box ML model capable of providing accurate selection probabilities for each *(customer, slot)* consisted of, first, performing a grid search among distinct models and hyper-parameters and, then, evaluating its applicability to capture the competition effect of offering a set of time slots. This section focuses on the grid search and model choice, since applicability discussions will be handled in Section 4.1.2.3.

The model choice process can be divided into two distinct phases: (1) selection of the ML model and (2), for this model, find the set of hyper-parameters that minimize the estimation error. During Phase (1), we tested the following ML algorithms:

Random Forest (RF) An ensemble of decision trees (DT) individually modeling their respective partition of the training data. The output of the model consists of the aggregate result of each DT after a voting mechanism.

Gradient Boosting Machine (GBM) A RF where the training of each DT is performed sequentially with a specific learning rate. The rationale behind this algorithm is to refine the training of subsequent estimators.

Logistic Regression (LR) Similar to the approach taken with GP-GOMEA. Instead, here model form is not learned and is rather assumed to be a linear function in the dataset features.

Neural Network (NN) The epitome of a black-box ML model. A NN consists of layers of nodes interconnected by edges. Nodes receive real numbers as inputs and apply non-linear functions to transform them into outputs that can be directed to subsequent nodes.

The grid search evidenced that GBMs outperformed the remaining models in terms of log-loss regardless of the chosen hyper-parameters. Ultimately, after refining model hyper-parameters, the GBM was capable of presenting a cross-validated log-loss of 0.135.

4.1.2.2 Symbolic expression

Using GP-GOMEA to derive symbolic expressions capable of modeling the WTP involved several iterations and study of different algorithm and instance configurations. Before delving into the results obtained, in Table 4.1 we present the parameters studied and their respective values.

Table 4.1: Parameters used to train symbolic expressions using GP-GOMEA

Parameter	Description	Values
	Instance	
N	number of training observations	1176, 3529, 5881
	Solution space	
Ω	set of functions	$\{+, -, \times, aq\}$, $\{+, -, \times, aq, exp, ln_p\}$, $\{+, -, \times, aq, exp, ln_p, \sqrt{\ }, {}^2\}$
Φ	set of features	small ($\lvert\Phi\rvert = 11$), large ($\lvert\Phi\rvert = 26$)
	Algorithm	
M	population size	1000, 2000
(h_i^{max}, h^{max})	maximum tree heights for the initial population generation and throughout the algorithm, respectively	(3, 5), (5, 10), (5, 15)
p_c	crossover probability	0.4, 0.5, 0.6, 0.8
p_m	mutation probability	0.2, 0.4, 0.5, 0.6
p_r	reproduction probability	0.0, 0.1, 0.2, 0.4
IMS	application of the Interleaved Multistart Scheme	$True$, $False$

As evidenced in Table 4.1, the size of the training data sets provided to GP-GOMEA was reduced. This strategy was employed due to the fact that GP is a very time consuming algorithm,

which does not allow to train algorithms using data sets with comparable size to those provided to black-box models. In terms of functions, we chose numerical operators exclusively. All of them keep their mathematical properties, except for functions aq and ln_p, which possess distinct definitions to handle domain restrictions. Operator aq consists of the *analytic quotient*, a function commonly used in symbolic regression problems to represent the mathematical division. Since a function $f(x) = \frac{1}{x}$ does not allow x to be null, the analytic quotient protects the division operation by performing the computation described by Expression (4.1). Regarding the protected function ln_p, its output consists of either the natural logarithm of a number x if $x > 0$, or 0 if $x \leq 0$.

$$aq(a,b) = \frac{a}{\sqrt{1+b^2}} \tag{4.1}$$

In terms of the configuration of GP-GOMEA, all parameters and their definition are consistent with the standard GP algorithm introduced by [11], except for *IMS*. IMS consists of the Interleaved Multistart Scheme, a procedure that eliminates the need for tuning parameters by running the algorithm with different settings throughout the evolution. Briefly, at every g generations of a run, GP-GOMEA initiates another run with double the population size. At the same time, every 2 runs, the maximum tree height is increased by 1. Throughout the execution of the algorithm, runs are progressively terminated due to several stopping criteria, e.g., the entire population converges to an identical solution, or a new run R' with a bigger population size achieved a better fitness than the analyzed run R or another run R'' with a population size bigger than R.

From our initial experiments, using *IMS* leads to better results. Therefore, in Table 4.2 we present the obtained performance and complexity of models obtained with parameter combinations selected from those in Table 4.1, $IMS = True$ and computational time limit of 1h. Instead of presenting a drill-down of the results for all parameter combinations, we only display $(|\Omega|, |\Phi|, M, (h_i^{max}, h^{max}))$ combinations for the sake of brevity and to expose the most relevant results. Column *size* is a measure of model complexity that counts the number of characters present in the symbolic expression. Performance measures $log\text{-}loss^{train}$, $log\text{-}loss^{test}$ and $log\text{-}loss^{test}_{bal}$ refer to the log-loss error for the training dataset, the test dataset (identical for all trained models) and the test dataset with class imbalance corrected through down-sampling. The log-loss error metric is only comparable between data sets with the same proportion of positive samples. Therefore, to evaluate the performance decay from the training to the test settings, we present $\%\Delta^{train}_{test}$. This metric provides the percentual variation between $log\text{-}loss^{train}$ and $log\text{-}loss^{test}_{bal}$, both presenting a balanced class proportion.

From Table 4.2, it is possible to observe that, for the same $(|\Omega|, |\Phi|, M)$ combination, increase the maximum admissible tree height increases model size, i.e., complexity, but leads to lower $log\text{-}loss^{train}$ values, i.e., better fitted models to the training data. In terms of the population size M, from the obtained results, it is not possible to say that increasing the population size automatically leads to better models. Having a larger population size is beneficial to ensure diversification by allowing the appearance of several individuals being more proficient in modeling specific patterns on the data, which can later be recombined by genetic operators. However, increasing population

Table 4.2: Performance of GP-GOMEA for modeling the WTP

$\|\Omega\|$	$\|\Phi\|$	M	(h_i^{max}, h^{max})	*size*	*log-loss*^{train}	*log-loss*^{test}	*log-loss*$^{test}_{bal}$	$\%\Delta^{train}_{test}$
4	12	1000	(5,10)	142	0.352	0.352	0.474	34.7%
			(5,15)	207	0.343	0.416	0.448	30.6%
		2000	(3,5)	49	0.406	0.327	0.578	42.4%
	27	1000	(3,5)	53	0.403	0.447	0.459	13.9%
			(5,10)	196	0.327	0.342	0.493	50.6%
		2000	(3,5)	57	0.388	0.406	0.515	32.7%
			(5,10)	171	0.362	0.371	0.502	38.7%
6	12	1000	(3,5)	43	0.461	0.492	0.553	20.0%
			(5,10)	167	0.345	0.348	0.498	44.7%
			(5,15)	164	0.321	0.340	0.506	57.3%
		2000	(3,5)	48	0.403	0.406	0.461	14.4%
			(5,10)	129	0.416	0.399	0.552	32.6%
			(5,15)	183	0.316	0.368	0.505	59.8%
	27	1000	(3,5)	54	0.386	0.380	0.539	39.7%
			(5,10)	137	0.369	0.406	0.474	28.5%
			(5,15)	165	0.327	0.361	0.491	50.2%
		2000	(3,5)	55	0.386	0.397	0.488	26.4%
			(5,10)	153	0.317	0.377	0.515	62.6%
			(5,15)	146	0.320	0.352	0.494	54.1%
8	12	1000	(3,5)	49	0.408	0.288	0.592	45.1%
			(5,10)	195	0.321	0.344	0.474	47.7%
		2000	(3,5)	49	0.451	0.563	0.546	21.2%
			(5,10)	151	0.328	0.334	0.509	55.4%
			(5,15)	143	0.382	0.407	0.536	40.3%
	27	1000	(5,10)	169	0.340	0.372	0.490	44.1%
		2000	(5,10)	134	0.366	0.404	0.494	34.9%

size leads to higher computation effort to handle the genetic operations and the assessment of the fitness of the whole population. Since we trained all our models with a time limit of 1h, runs having $M = 2000$ might have resulted in a lower total number of generation, leading to under-fitted symbolic expressions. Regarding parameters $|\Omega|$ and $|\Phi|$, our results do not display an apparent trend regarding their impact on model performance and complexity.

Comparing the training phase and the testing phase, it is possible to observe that there is a significant performance decay, as evidenced by the positive values of $\%\Delta^{train}_{test}$. This result may be caused by two factors: (1) the reduced number of training instances used and (2) the tight computational budget used to train the models. These two factors combined may have lead to under-fitted models incapable of accurately modeling the WTP problem.

The selected model was the one which minimized the *log-loss*^{test}. This model presented a *log-loss*^{test} of 0.288 and was obtained by running GP-GOMEA with the combination ($|\Omega|$, $|\Phi|$, M, (h_i^{max}, h^{max}), p_c, p_m, p_r) = (8, 12, 1000, (3, 5), 0.4, 0.4, 0.2). Figure 4.1 presents a tree

representation of the derived symbolic expression.

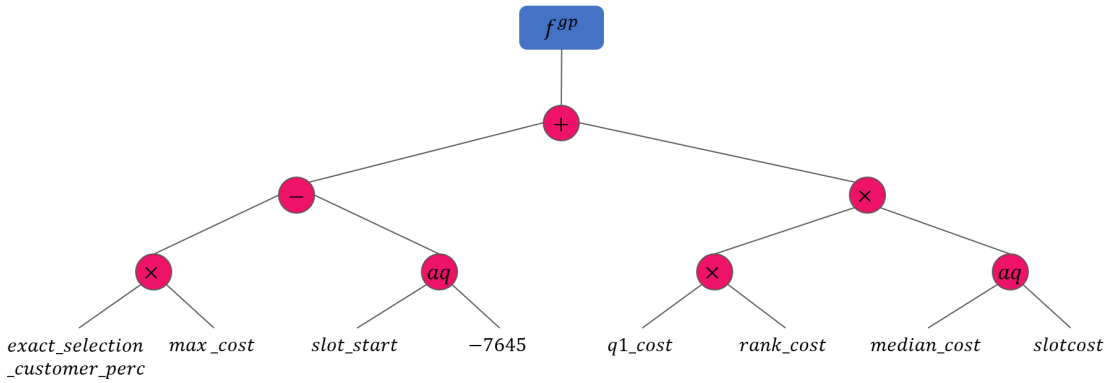

Figure 4.1: Symbolic expression chosen to be deployed in the simulation

From the observation of the expression of f^{gp} it is possible to understand some of the behavior of this model. Focusing on the sub-tree that forms expression $(exact\_selection\_customer\_perc \times max\_cost)$ - $aq(slot\_start, -7645)$, one can conclude that the historical percentage of customer selection of a given slot is proportional to the selection probability. Additionally, we see that time slots starting earlier tend to be preferred, as $slot\_start$ impacts negatively f^{gp}. Focusing on sub-tree $(q1\_cost \times rank\_cost) \times aq(median\_cost, slotcost)$, it is possible to observe the price of a time slot is inversely proportional to its selection probability.

4.1.2.3 Model comparison

From the results presented in Sections 4.1.2.1 and 4.1.2.2, it is already possible to determine that the GP-GOMEA-derived symbolic expressions are incapable of competing with the GBM for the WTP prediction problem. Comparing the performance of both methods on the same testing dataset, the performance discrepancy is very accentuated, with the GBM presenting a log-loss of 0.095, whereas the symbolic expression is only capable of achieving a value of 0.277.

It is true that GP-GOMEA falls behind the GBM for this classification problem. However, for the purpose of solving the DTSPP, these WTP models should be capable of estimating the customer choice probability distribution function and of separating the most preferred time slot from the remaining set of options. To identify if the models were accomplishing this goal, we built the graph presented in Figure 4.2. This graph is obtained by applying the WTP models for each historical order and sorting in a descending order the obtained selection probabilities. Afterwards, the percentage of orders where the selected slot fell inside the top n most preferred slots was obtained.

From Figure 4.2, it is possible to see that the GBM and GP-GOMEA are capable of guessing the correct time slot in 29% and 24% of the orders, respectively. This result might seem unsatisfactory, but knowing that each order is presented with several time slots, the task of identifying the option that the customer will select is extremely demanding. To demonstrate the usefulness of the developed WTP models, we can compare this same result for a naïve model that assumes that each

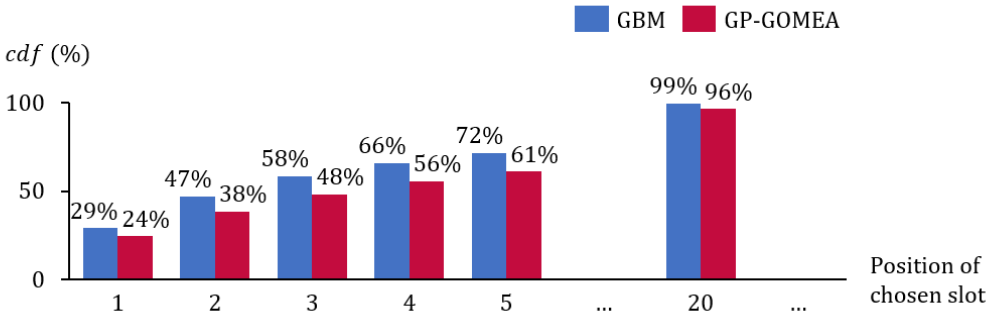

Figure 4.2: Percentage of cases where the chosen time slot belonged to the top n most likely options

customer will select the same time slot as the one chosen in the previous order. Applied to the same test dataset, this model is only capable of correctly guessing the chosen time slot in 17% of the cases. Additionally, by looking at the 5 most preferred options as considered by the models, it is possible to see that in 72% and 61% of the cases for the GBM and GP-GOMEA, respectively, will contain the selected time slot, which is a very satisfactory result.

4.1.3 Explainability Discussion

The goal of this section is to explore the explainability of both models individually and to analyze if their behavior is congruent. Additionally, since price is a key component of the DTSPP, we perform a sensitivity analysis to this factor in order to ensure trust on the outputted predictions.

Deriving a symbolic model through GP consists of an explainable-by-design approach. Through solution space constraints, e.g., limiting the tree height of individuals, or solving a multi-objective problem considering the minimization of error estimates and model complexity, GP can offer analytic expressions that can provide explanations at a model-level. In fact, Section 4.1.2.2 has already delved into such explanations by exploring the effect of features such as the historical customer slot choice percentage, the starting time of a slot and its price, on the probability of selection. The symbolic expression in Figure 4.1 is rather interpretable, apart from the features pertaining to the pricing distribution of the presented slot panel. Indeed, upon varying values of *slotcost*, the remaining price features *max_cost*, *q1_cost*, *rank_cost* and *median_cost* could potentially vary as well, which impedes performing an univariate analysis of the effect of each feature.

In terms of the BBM model, explainability is treated as two-step process. After learning the GBM, model explanators are obtained to interpret its output. Figure 4.3 presents the SHAP summary plot with the 4 most important features. For each feature, the graph plots feature values in terms of their impact on the selection probability. Moreover, each plotted point presents a color in accordance to the specified color grading, i.e., higher feature values lean towards the purple color while lower feature values tend to the blue color.

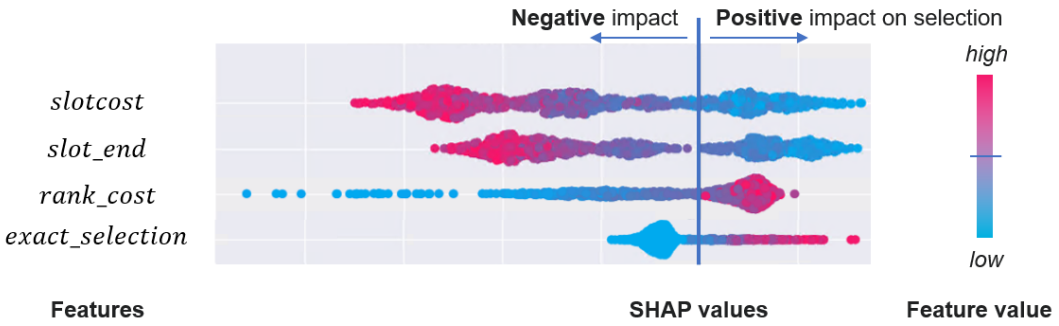

Figure 4.3: SHAP summary plot for the trained GBM model

Interestingly, the 4 most important features for the GBM coincide with the features selected by GP-GOMEA to build the regression tree. The only difference lies on *slot_end*, as the symbolic expression contains the feature *slot_start*. However, these two features are highly correlated as *slot_end* can obtained by adding the slot width to the *slot_start*, and vice-versa. Furthermore, the GBM learned the same effect as the one modeled by the symbolic expression for the considered features on the selection probability. Indeed, higher values of *slotcost* and *slot_end* (hence, *slot_start*) reduce the selection probability. Conversely, higher values of *rank_cost* and *exact_selection_customer_probability* lead to higher selection probabilities.

Even though the developed models are price sensitive, we need to assess if the effect of price shifts leads to intuitive model behavior. In other words, not only does the slot selection need to increase (decrease) upon price discounts (mark ups), but also the overall selection probabilities of the remaining time slots should vary accordingly, i.e., their average selection probability should decrease (increase). To assess this phenomenon, we present Figure 4.4 that presents the percentage of cases where the behavior displayed by the model corresponded to what was expected for different degrees of price shifts ranging from -0.5 to 0.5. The upper graph presents the percentage of cases where the selection probability of time slot (w,d) increased and decreased for negative and positive price shifts, respectively. Conversely, the lower graph presents evaluates the cases where the average selection probability of the remaining unaffected time slots increased upon price increases for time slot (w,d), and vice-versa.

Surprisingly, GP-GOMEA beats the GBM machine for every price shift point while modeling the selection probability of the affected time slot. Indeed, the symbolic expression is capable of displaying the correct behavior in every single case. Moreover, we see that the GBM reacts better to time slot price markdowns. For the unaffected slots, GP-GOMEA is also better at understanding that the their average selection probability should increase upon discounts on the affected time slot. Nevertheless, for price mark-ups, both methods compete in trying to indicate that the attractiveness of the remaining slots should increase. However, for a price shift of $+0.5$ m.u., the GBM reacts better.

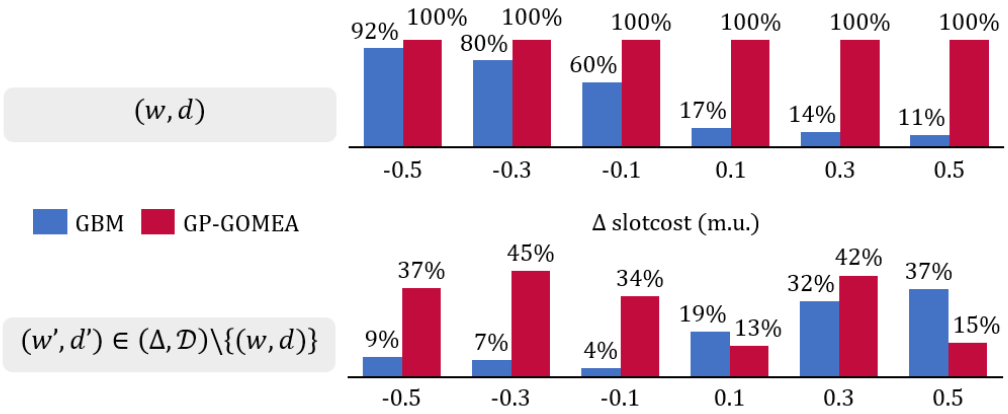

Figure 4.4: Price sensitivity analysis comparison for the GBM and the symbolic expression generated through GP-GOMEA

4.2 CTS Results

This section delves into the performance of the method developed to anticipate potential routing costs. First, Section 4.2.1 will expose the dataset used to train the models. Then, Section 4.2.2 will analyze the performance of the CatBoost regressor in handling the CTS predictive problem.

4.2.1 Transportation Data

In the context of its AHD, the retailer relies on third-party logistics service providers (3PLs) to ship the orders. Since both the WTP and CTS models will be deployed in the prescriptive heuristic, the retailer provided routing data pertaining to the same period as the one analyzed for the WTP, from October 2016 to September 2017. Among the 3PLs, we received data from the two main partners. Therefore, we initially had to carry out some data cleaning and standardization procedures to make sure that information from two distinct sources was consolidated.

After having gathered all data, we cleaned routes where at least one order was canceled, as we only consider that a customer can walkaway during the slot selection process. Moreover, since we are relying on operational data filled by truck drivers, some information was incongruous. Indeed, we had to exclude orders where the time of arrival at each point was not registered, whose route included duplicated orders, or even where the waiting time at a given stage was negative.

Even though the dataset presented several inconsistencies, we decided to derive an approach based on them. We took this decision because we do not wish to provide the retail partner with a routing solution approach to solve the Vehicle Routing Problem With Time Windows (VRPTW). The most appropriate procedure would be to apply a VRPTW solution approach to historical orders and consider the originated routes as ground truth for the ML problem that tries to predict the final transportation cost for each incoming order. However, such a strategy would entail the retailer building its routes using the same solution approach. Assuming that the routing method used by the retailer matches the performance of the VRPTW solution approach, even in that case, such a

method would not capture unpredicted events that may occur in this operation, e.g., canceled orders or customers absent from home.

4.2.2 Performance Discussion

The application of the CatBoost regressor on the described CTS prediction problem fell short of expectations. Even though the mean absolute error (MAE) was only 0.73 m.u., the R square metric was 0.251. In other words, our CTS model explains around a quarter of the variability in the testing dataset. Figure 4.5 plots the transportation cost distributions for real data and the predictions made by our model. As can be observed, although our model incurs on a low bias, it is very conservative. Indeed, it produces predictions mainly centered at the second and third quartiles of the real transportation cost distribution.

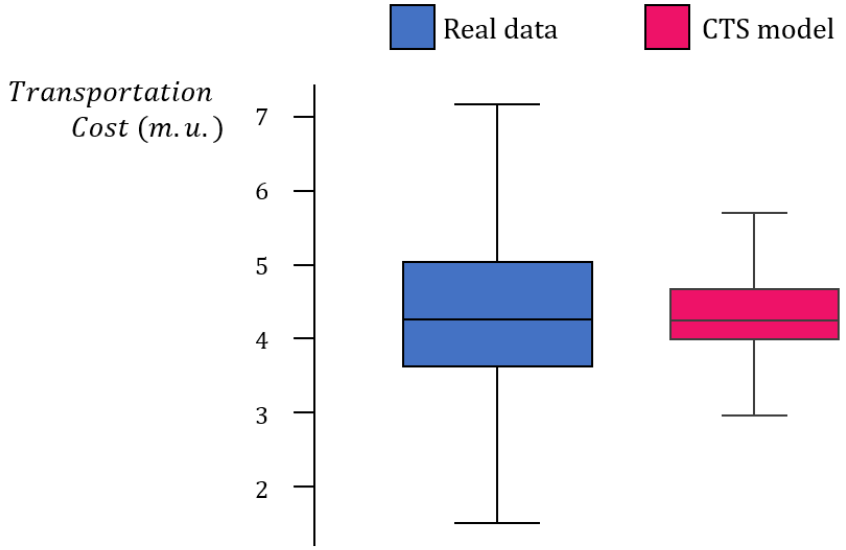

Figure 4.5: Comparison of real and predicted transportation cost distributions

Despite its poor performance, the CTS model displays a relatively stable MAE throughout time. Figure 4.6 presents the evolution of the MAE as the prediction is made progressively more distantly from the time slot cutoff. As expected the error increases slightly as the distance to the cutoff increases, since less information on the final state of the system is known by then.

4.3 PH Results

This section unveils the results achieved by our solution approach on a test instance obtained by isolating a set of contiguous orders from the historical records kept by the retailer. Section 4.3.1 delves into the assumptions taken to create the test instance. Afterward, Section 4.3.2 performs a sensitivity analysis on objective function weights to compare profit- and efficiency-driven pricing

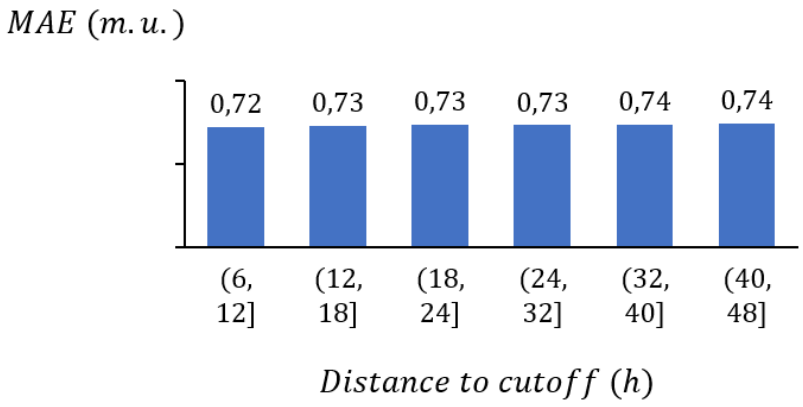

Figure 4.6: Evolution of the MAE throughout time

policies. Additionally, we contrast the performance of static pricing strategies against the proposed dynamic pricing policy.

4.3.1 Instance

The DTSPP presents a feature that simplifies not only the derivation of a solution approach but also the obtention of a testing instance: demand is indivisible, i.e., a customer can only be served by a single store. So, we started by selecting the store with the highest number of past orders. Among all areas served by this store, we chose two contiguous delivery areas to account for cases where vehicles perform routes that address customers in different regions. Having defined the geographical scope of the test, we proceeded to selected a sequence of $C = 1300$ past orders that occurred in the middle of the provided data set, between late March and early April of 2016. Considering the inter-arrival intervals, this sequence of orders corresponds to a booking horizon of 4 days. The set of time slots $(\Delta, \mathscr{D})$ consists of a union of the time slots seen by every simulated customer $i \in \mathscr{C}$. Furthermore, we filtered the set of time slots based on the assumptions mentioned in Section 4.1.1.

Regarding time slot features, it is necessary to define their initial occupation and capacity in number of orders. We set the initial occupation of each slot to be the respective number of booked orders since their opening time until the simulation start time. We sized time slot capacity according to the number of expected orders in the simulation divided by the average number of time slots in a price panel for each delivery area. To avoid having a constricted instance in terms of capacity, we added a buffer of 5% of the computed capacity. Additionally, we considered that a time slot is available while its occupation has not reached capacity during the period ranging from 7 days to 4 hours prior its earliest time, a_{wd}.

Finally, we set the Advanced-Booking Time (ABT) curves to be a linear function that links the initial occupation at its opening time to its capacity at cutoff.

4.3.2 Performance Discussion

To assess the performance of our solution approach, we deployed it using the parameters listed in Table 4.3. Additionally, we defined several ratios of objective function weights $r_w = \frac{w_\omega}{w_\phi}$ to perform a sensitivity analysis. Analyzing the simulation outcome upon varying objective function weights enables identifying whether the solution approach offers consistent results when the target shifts from profit to efficiency.

Table 4.3: Solution approach parameters used in the simulation

Number of candidate solutions and affected time slots			
n	1000	τ	$0.25 \times \lvert(\Delta, \mathscr{D})_{k=0}\rvert$
Price parameters			
δ	0.5	(π^-, π^+)	(2, 9)
Objective trade-off			
r_w		$\{0.01, 1, 100, 1000, 2000, 5000\}$	

Section 4.3.2.1 presents the sensitivity analysis to the ratio $\frac{w_\omega}{w_\phi}$, while Section 4.3.2.2 compares the dynamic pricing strategy against the retailer's current static pricing policy.

4.3.2.1 Sensitivity Analysis

Considering lower values of r_w, i.e., profit-oriented decisions, might lead to a strategy that to customer preferences. Thus, it is likely that the most popular slots will close early. Additionally, customer walkouts will reduce. The profit associated with each customer consists of the balance between the revenue captured by the basket value and slot price subtracted by the delivery cost. Therefore, the prescribing heuristic will tend to drive prices upwards to try to capture the customer's willingness to pay while at the same time not leading to a walkout which enables securing the basket value.

To confirm this hypothesis, we ran 24 simulations for each of the specified r_w values, as the heuristic and customer selections are random. Figure 4.7 compares the evolution of the *walkaway ratio* and early slot closures for decreasing levels of r_w. The metric *early closure* corresponds to the sum of the total time each slot was closed before its cutoff in days.

As r_w increases, the focus of the prescribing heuristic shifts from guaranteeing a balanced slot occupation and becomes concerned in extracting as much revenue from each customer possible. Therefore, the percentage of customer that walks out decreases, as the price offers are no longer trying to push customer selections to underused time slots, and are focused on offering the best price possible for the time slots that the customer values the most. In contrast, the total time of early slot closures increases, as the most popular time slots close long before their cutoff. When slot occupation targets are valued the most, it is possible to observe that this metric is better controlled.

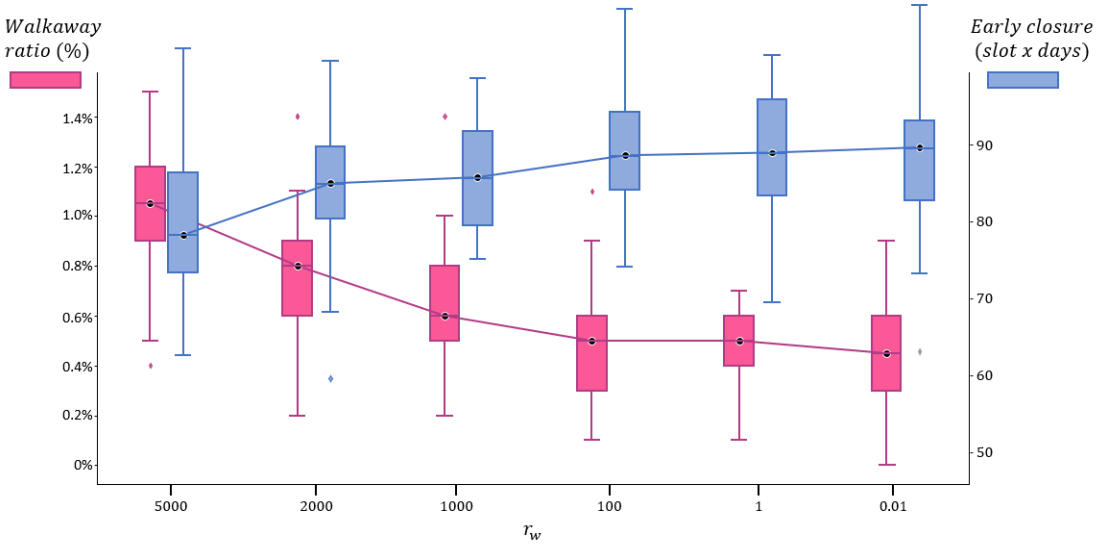

Figure 4.7: Impact of objective function weights on the percentage of walkaways and slot availability

The DTSPP is a very comprehensive problem involving a customer service component, as well as a delivery efficiency point-of-view. Therefore, beyond the aforementioned KPIs, several others could be monitored. In Table 4.4, we present the impact on profit, given as a percentage of the maximum profit obtained using $r_w = 0.01$. Additionally, we provide the average price at which time slots are selected, as well as the average price of the presented price panels to the customers. Finally, we also disclose the average number of slots the customer has available for selection.

Table 4.4: Impact of trading-off profit and slot occupation balancing on relevant business KPIs

r_w	5000	2000	1000	100	1	0.01
Profit (%)	99.18	99.47	99.63	99.93	99.96	100.00
Selection price (m.u.)	5.90	5.90	5.90	6.01	6.20	6.26
Panel average price (m.u.)	6.31	6.30	6.30	6.34	6.36	6.36
Available slots (units)	27.81	27.86	27.72	27.52	27.56	27.42

Naturally, as r_w decreases, profit gains become more important for the pricing heuristic, which leads to economical improvement. By observing the average selection price, we can assess the usefulness of our WTP model. By focusing on profit rewards, our prescribing heuristic attends to customer preferences. Instead of offering discounts to secure the purchase of a customer, using the WTP model, the prescribing heuristic is capable of estimating the willingness to pay of each customer and of actually increase the price at which customers select time slots.

Interestingly, even though we did not model such a restriction, throughout the several values of r_w, the average price of the successive time slot panels presented to customers were relatively stable. From a business perspective, this is a desirable phenomenon, as the retailer might not want to change drastically the level of prices presented to customers, as it might change their perception

of the value of time slots. Finally, since rewarding profit disregards time slot load balancing, early closures become more frequent which also reduce the number of alternative time slot options displayed to the customer.

4.3.2.2 Effect of Dynamic Pricing

Currently, the retailer has no concern with ensuring a balanced occupation of the offered time slots. Indeed, its tactical time slot management process consists of adjusting the capacity of time slots according to historical customer preferences. Therefore, we compare its current time slot pricing strategy against a dynamic pricing profit maximizing scenario, i.e., $r_w = 0.01$. Table 4.5 compares both strategies along the KPIs analyzed so far. The only addition is the *load balancing* measure, which determines the percentage of slots that hit 95% of the planned capacity.

Table 4.5: Impact of adopting a dynamic time slot pricing strategy on relevant business KPIs

KPI	*units*	Static Pricing	Dynamic Pricing	Impact
Profit	% of baseline profit	100.0	103.8	3.8
Selection price	m.u.	6.26	6.26	0.00
Walkaways	%	4.20	0.45	-3.75
Slot availability	number of slots	28.2	27.4	-2.8%
Load balancing	% of slots hitting 95% capacity	31.77	44.79	13.02

From Table 4.5, it is possible to observe that the dynamic pricing strategy was capable of increasing the operational profit by 3.8 percentage points (p.p.). This result was achieved not because of an increase in the average selection price, but due to a reduction on customer walkaways by 3.75 p.p.. Additionally, even with a greedy dynamic pricing strategy, time slot occupation improved as almost half of the time slots achieved 95% of their capacity. This result represents an improvement of 13.08 p.p. over the static strategy. Finally, due to the reduction in customer walkaways and a better occupation of time slots, the number of slots presented to the customer reduced by 2.8%.

4.4 Practitioners' Validation

The initiatives taken to validate the present models can be grouped into three validation phases: (1) problem framing, (2) model development, and (3) business results, which are discussed in this section.

4.4.1 Problem Framing Validation

The initial interactions with Sónia Germano, Operations Manager for E-commerce Transportation at Sonae MC at the time, augmented our perception of the challenges involved in time slot management and of our retail partner's approaches. These interactions helped define the scope of the problem and establish useful assumptions, e.g., each customer location is addressed exclusively by one store. This previous fact enables assuming there is indivisible demand and validates that time slot management can be performed at each store separately.

Regarding the validation of explainability requirements, we conducted an interview with Sónia Germano to obtain information to guide the user study designed by the University of Tartu, reported on D2.1 'User studies on the realisation of explanations' (for the complete interview transcription, refer to Appendix 6 of the same deliverable). From the perspective of the interviewed operations manager, explainability through directly presenting the mathematical expression would not be preferable. Therefore, further explanations that can run on top of the models (such as counterfactuals) are necessary to interact with this person. This does not preclude the need for symbolic expressions, as explainability by design is still important to the model developer.

4.4.2 Model Development Validation

The developed symbolic expression to model customer behavior yielded satisfactory results with respect to explainability. In fact, the expression includes an anticipated price sensitivity relationship where the slot selection probability decreases for increasing prices. Another interesting finding is that the percentage of past slot selections deems the likelihood of the customer's current selection. Furthermore, the customer has a higher preference for time slots closer to the moment of ordering, as $slot\_start$ increases affect negatively the selection probability. The latter fact is aligned with the study conducted by [2], which indicates that customer value delivery speed.

In terms of the prescription of time slot prices, the validity of the results of our prescriptive heuristic was proved by our sensitivity analysis. As reported in Section 4.3, the impact on operational profit and customer walkaways when the relative importance of the profit and operational efficiency objectives changes is intuitive. Indeed, in a scenario where the retailer substantially prefers the efficiency objective over profit, customer walkaways increase as price offerings will be adjusted to discourage the selection from customers whose delivery incurs a high transportation cost. Alternatively shifting the focus to profit maximization, customer walkaways decrease as price offerings will set prices that grant a customer selection and, as a consequence, the associated basket value.

4.4.3 Business Results Validation

After obtaining the complete solution approach for the DTSPP, we scheduled a meeting with Fábio Santos, current Operations Manager at Sonae MC. During the meeting, we had the opportunity to explain our methodology and disclose our model validation analysis. The operations manager's perception was aligned with the development team in what concerns model intuitiveness. Moreover,

the decision-maker deemed this solution approach as relevant, since it can be used as a test tube to explore time slot pricing policies.

Chapter 5

Conclusions and Future Developments

In this chapter, we derive the main conclusions and point possible research directions to be explored in the following iterations of this use case.

5.1 Final considerations

This research project has now achieved important milestones regarding the objectives of the considered online retail use case. We implemented a first version of a prescriptive heuristic that is capable of prescribing time slot price panels, while maximizing profit and following Advanced-Booking Time (ABT) curves. This heuristic is composed of two predictive models (Willingness To Pay (WTP) and Cost To Serve (CTS)) and a price panel generation procedure. Explainability methods were applied to the WTP predictive problem. Furthermore, to evaluate the performance of the prescriptive heuristic in solving Dynamic Time Slot Pricing Problems (DTSPPs), a simulator of booking horizons was implemented. At this point, each component is at a different state of maturity regarding performance and explainability metrics.

The WTP predictive problem was solved by two alternative approaches, Genetic Programming (GP) and Gradient Boosting Machine (GBM). Both approaches consider real-word data such as the previous slot selections, pricing dispersion measures, and the current shopping basket of the customer. The output of a GP algorithm is a symbolic expression that is considered to be a white-box model. The output of the GBM approach is seen as black-box model that needs to be explained using SHAP. In terms of raw predictive power, we observed that the obtained GP expressions are not yet able to compete with the GBM model. Nonetheless, the GP results suggest that symbolic expressions can make bolder predictions for positive samples. Additionally, it was possible to verify that larger tree heights result in better models, but we believe GP models are under-fitted due to the hardness of the optimization problem that needs to be solved. It is clear that GP cannot be used in a "plug-and-play" fashion and that ad-hoc explainability comes at a cost. Therefore further work is needed on tweaking terminals, operators, and objective functions considered in the evolutionary process. Regarding the post-hoc explanations obtained for the GBM model, it was possible to verify that the GBM model is price sensitive and to understand the features with highest impact on

time slot choices. For instance, the slot price has a high negative impact on the probability to select a certain time slot. Note that SHAP showed to be a valuable method to provide local explainability. The CTS predictive problem was approached using a regression model that works states of the transportation system. These states, which are described by a number of transportation-related features such as customer dispersion, time slots occupation, and remaining time before the booking period cut-off, are used to predict the delivery cost of each order. Note that this delivery cost prediction is also part of the necessary inputs for the time slot pricing heuristic. The results of the CTS model suggest that predicting delivery costs with an incomplete view of incoming customers several days before the delivery date is very challenging. Although the model shows very low bias, it predicts conservative delivery costs with a low explained cost variance. Further work will be necessary to improve two aspects of this approach. First, the dataset needs to be improved, as the operational data regarding deliveries is scarce and shows some incongruities. Second, we believe that a approach to the problem could improve the current results. For instance, besides a GP approach, advanced reinforcement learning approaches could be tried. Furthermore, different decomposition approaches can also be tried, dividing the transportation cost into three components: waiting time, traveling time, traveling distance.

Finally, we were able to propose a first attempt at solving the DTSPP by means of a prescriptive heuristic that selects a set of time slot prices, a price panel, to be shown to each customer. The heuristic receives WTP and CTS approximations and generates price panels through a predefined procedure. The price panel to be shown among all generated price panels is selected in a greedy randomized fashion, considering a subset of price panels characterized by larger expected profits. A sensitivity analysis, performed to weight parameters figuring in the objective function, allowed to provide evidence that the heuristic is behaving as expected. On one hand, when the objective function is focuses on profit, fewer walk-ways occur, as the basket value is the main contributor to profit. On the other hand, when the focus is on slot load balancing, early slot closures are minimized, ensuring that the customers have a wider range of options through the booking horizon. We observe that the propose prescriptive heuristic is able to pursue the desired time slot occupation, modeled by means of ABT curves. Therefore, from a practical perspective, operations managers are able to control the logistic capacity demand by a certain booking horizon and, at the same time, they are able to balance profit and service level.

5.2 Future developments

Despite the valuable contributions achieved so far in the online retail use case, there are a few improvements and developments we expect to introduce in the near future.

Improve WTP explainable model – The WTP explainable model developed in Section 3.2 is one of our first attempts at building explainable models. Therefore, we are confident that improvements can be introduced both in terms of performance and explainability metrics. To improve performance, new terminals and operators need to be tested, so as to capture

the main drivers of the WTP of each customer. To improve explainability of the symbolic expressions, new constraints and objective function penalty components should be explored. Note that due to the computational complexity that is inherent to most GP frameworks, the referred experiments require a considerable amount of computational resources (such as RAM and CPU). Therefore, the exploration of different hyper-parameter combinations needs to be meticulously planned and efficiently implemented.

Propose CTS explainable model – The CTS model presented in Section 3.3 is not explainable. Hence, in the following developments of this use case, we intend to develop an explainable CTS model that is able to predict the cost to serve a customer during the service period in an explainable manner. Given the complexity of the final prescriptive approach, it is of utmost importance to maintain the explainable character of each of its components. Note that the complex process of exploring transportation related features to improve the performance and explainability of these models is also expected to require extensive of computational experiments. Nonetheless, developing efficient and explainable algorithms to provide transportation cost approximations can result in widely applicable models, given that there are numerous sequential decision problems which incorporate transportation components.

Develop pricing policy recommendation algorithm – Our first approach to the DTSPP, presented in Section 3.4, is a prescriptive heuristic demanding a considerable set of parameters. This heuristic is based on the generation of a set of time slot pricing panels, which are classified according to the profit their are expected to generate in case they are shown to a certain customer. At the moment, the selection of the best price panel to be shown is performed by a greedy-randomized procedure, which chooses a random panel among the 5 best in terms of expected profit. In fact, there is no real pricing policy being extracted, that is, we already implemented a pricing policy. However, we intend to implement a GP algorithm running over the simulator of the booking horizon, in order to extract a policy to choose the best time slot pricing panels to maximize the expected profit including immediate and future rewards. The idea is to find a policy (a symbolic expression) that runs fast and that is explainable. In this case, explainability should allow the user to understand what are the best panels to be shown for certain WTP profiles that have to be served with a certain CTS.

Comparisons with other sequential decision problem approaches – We aim at providing extensive computational experiments to compare the developed approach against different policies and against different policy extracting methods. For instance, given that artificial neural networks is a typical approach to extract policies (i.e., the neural network parameters), it is important to compare them against our explainable approaches. Additionally, it would be interesting to compare symbolic model-based policies against policies that are obtained using approximate dynamic programming approaches to solve the Bellman equation [5]. In this type of approach, policies are extracted after attributing a value to each state that can be visited by the system. These state-value and action-quality-based policies have been

successful in transportation-related applications in the past, thus their consideration should not be overlooked.

TRUST framework functionalities – Finally, it would be interesting to improve TRUST framework so that it further helps the development of explainable approaches to solve problems that are similar to the DTSPPs. At the current state of this online retail use case, we believe it would be interesting to implement two main new functionalities. First, the TRUST framework should allow users to perform several training iterations where there is the possibility to incorporate aspects that were learned in a previous training session. For instance, it would be interesting to combine some features to create a new feature to figure in a future training session. Second, given that the current approach for this use case is using three different models, developing a model pipeline builder would be useful. That way, it would be possible to integrate the WTP, CTS, and prescribing heuristic in a single loop.

5.3 Recommendations for TRUST-AI Framework

Our symbolic expression for the WTP prediction problem was derived using GP-GOMEA, one of the genetic programming algorithms contemplated in the framework's toolkit. From our experience, the framework should allow interrupting the algorithm's evolution to inspect and refine specific symbolic expressions in the population. In the resulting WTP model derived through genetic programming, represented in Figure 4.1, the feature $slot\_start$ is being divided by -7645. Through the analysis of the feature distribution, it was possible to determine that this value is close to the mean $slot\_start$ for the training dataset. Thus, instead of using an ephemeral random constant as a terminal, a new variable could be added to the genetic programming optimization problem with the mean $slot\_start$. Alternatively, we could remove the ephemeral random constant and normalize the feature using its mean value.

Our experience with solving the DTSPP leads us to claim that the framework should be capable of simulating different individuals under different scenarios to assess their performance. In the context of predictive problems, e.g., the CTS prediction problem, being able to assess the performance of the trained model with different datasets or relevant segments of a dataset would be desirable. Indeed, it would allow inspecting model behavior, such as assessing whether cost predictions are worse when made earlier in the booking horizon. For prescriptive settings, changing the conditions of the simulation would be ideal. In our case, it would enable performing the same sensitivity analysis as the one described in the present deliverable.

Appendix A

Notation

A.1 Binomial Classification problem

Indices

i	training observation as a customer, time slot combination

Parameters

N	number of training observations
p_i	selection probability for observation i
y_i	the true outcome for observation i

A.2 Walkway Probability Modeling

Indices

i	training observation as a customer, pricing panel combination

Parameters

α^*	proportion of walkaways for current static pricing policy
c_1	localization parameter of the s-shaped walkaway curve
c_2	shape parameter of the s-shaped walkaway curve

Variables

x_i	maximum customer selection probability in a price panel
$w(x_i)$	walkaway probability for the given customer, price panel combination

A.3 GP-GOMEA

Mathematical functions

$\mathscr{L}$	the loss function to be minimized
f	a symbolic expression derived through GP-GOMEA
f^{gp}	the symbolic expression that minimizes $\mathscr{L}$
f^{wtp}	the expression that applies a sigmoid transformation to function f^{gp}

Parameters

$\mathbf{y}$	array storing the true outcome for each training observation
$\mathbf{X}$	matrix containing the features of all training observations

Sets

$\mathscr{P}$	primitive set
Ω	set of functions
Φ	set of features
Γ	set of constants
$\mathscr{F}$	solution space of symbolic expressions f

A.3.1 Cost To Serve Estimation Model

Indices

k	training observation
i	regressor

Parameters

N	number of training observations
I	number of regressors
β_i	coefficient of feature x_i
y_k	cost to serve of observation k
x_i^k	value of regressor i for observation k
$W \cdot D$	number of time slots

A.4 Prescriptive Heuristic

Indices

k	decision epoch
i	customer
(w, d)	time slot w on delivery day d

Variables

S_k	system state at decision epoch k
$\mathbf{f}_k$	the prescribed price panel at epoch k
f_{wdk}	the prescribed price for time slot (w, d) at epoch k
$\mathbf{f}_k^b$	the base price panel obtained at a tactical level k

Parameters

w_ϕ	weight attributed to profit rewards
w_ω	weight attributed to occupation targets
$\mathbb{P}_{wd}(\mathbf{f}_k)$	selection probability for time slot (w, d) upon decision $\mathbf{f}_k$
c_{wdk}	the cost to serve a customer i on time slot (w, d) at epoch k
$l^e_{wd}(t_k)$	the effective occupation percentage of time slot (w, d) at the time instant of the decision epoch k
$l^e_{wd}(t_k)$	the desired occupation percentage of time slot (w, d) at the time instant of the decision epoch k
n	the number of candidate price panels to generate for each incoming customer
τ	the number of time slots affected with price variations to generate candidate price panels
δ	price step variation applied to time slot prices
π^-	minimum admissible time slot price
π^+	maximum admissible time slot price

Sets

Δ	the set of available time slots
Δ^i	the set of available time slots for customer i
$\mathscr{C}$	set of customers
Φ	set of prescriptive heuristic parameters
$\mathbf{F}_k$	set of n candidate price panels generate at decision epoch k

Functions

$\Gamma(S_k, w, d)$	transition function that changes state S_k to contemplate the decision of customer i of booking time slot (w, d)
$\phi(\mathbf{f}_k)$	expected profit obtained through the application of price panel $\mathbf{f}_k$
$\omega(\mathbf{f}_k)$	expected average percentual deviation of real to target occupation ABT curves through the application of price panel $\mathbf{f}_k$

References

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[2] Pedro Amorim, Nicole Dehoratius, Fredrik Eng-Larsson, and Sara Martins. Customer Preferences for Delivery Service Attributes in Attended Home Delivery. Chicago Booth Research Paper No. 20-07. Available at SSRN: https://ssrn.com/abstract=3592597, 2020.

[3] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115, 2020.

[4] Frank M. Bass. A new product growth for model consumer durables. *Management Science*, 15(5):215–227, 1969.

[5] Richard Bellman. *Dynamic Programming*. Dover Publications, 1957.

[6] Hongzhen Chen, Zhichao Shen, Le Wang, Chongchong Qi, and Yinghui Tian. Prediction of undrained failure envelopes of skirted circular foundations using gradient boosting machine algorithm. *Ocean Engineering*, 258:111767, 2022.

[7] EU Comission. Transparent, Reliable and Unbiased Smart Tool for AI. CORDIS. https://cordis.europa.eu/project/id/952060, 2021. (accessed July 11, 2022).

[8] Ethan Cramer-Flood. Global ecommerce 2020, Jun 2020.

[9] Robert Klein, Jochen Mackert, Michael Neugebauer, and Claudius Steinhardt. On the approximation of opportunity cost for dynamic pricing in attended home delivery. *SSRN Electronic Journal*, 01 2016.

[10] Sebastian Koch and Robert Klein. Route-based approximate dynamic programming for dynamic pricing in attended home delivery. *European Journal of Operational Research*, 287(2):633–652, 2020.

[11] John R. Koza. Genetic programming - on the programming of computers by means of natural selection. In *Complex adaptive systems*, 1993.

[12] Connie Lee. Know the new e-shoppers, Jun 2022.

[13] Scott Lundberg. Welcome to the SHAP documentation. Forbes. https://shap.readthedocs.io/en/latest/index.html, 2018. (accessed July 11, 2022).

[14] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc.

[15] Ramaravind K. Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* '20, page 607–617, New York, NY, USA, 2020. Association for Computing Machinery.

[16] Warren B. Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality (Wiley Series in Probability and Statistics)*. Wiley-Interscience, USA, 2007.

[17] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery.

[18] Khuyen Tran. SHAP: Explain Any Machine Learning Model in Python. Towards Data Science. https://towardsdatascience.com/shap-explain-any-machine-learning-model-in-python-24207127cad7, 2021. (accessed July 11, 2022).

[19] Marco Virgolin, Tanja Alderliesten, Cees Witteveen, and Peter A. N. Bosman. Scalable genetic programming by gene-pool optimal mixing and input-space entropy-based building-block learning. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '17, page 1041–1048, New York, NY, USA, 2017. Association for Computing Machinery.

[20] Marco Virgolin, Tanja Alderliesten, Cees Witteveen, and Peter A. N. Bosman. Improving model-based genetic programming for symbolic regression of small expressions. *Evolutionary Computation*, 29(2):211–237, 2021.

[21] Marco Virgolin and Solon Pissis. Symbolic regression is np-hard. *ArXiv*, 07 2022.

[22] Timothy Webb, Zvi Schwartz, Zheng Xiang, and Manisha Singal. Revenue management forecasting: The resiliency of advanced booking methods given dynamic booking windows. *International Journal of Hospitality Management*, 89:102590, 2020.

[23] Xinan Yang, Arne Strauss, Christine Currie, and Richard Eglese. Choice-based demand management and vehicle routing in e-fulfillment. *Transportation Science*, 50:140807110218007, 08 2014.