

---

# Model Distillation for Revenue Optimization: Interpretable Personalized Pricing

---

Max Biggs<sup>1</sup> Wei Sun<sup>2</sup> Markus Ettl<sup>2</sup>

## Abstract

Data-driven pricing strategies are becoming increasingly common, where customers are offered a personalized price based on features that are predictive of their valuation of a product. It is desirable for this pricing policy to be simple and interpretable, so it can be verified, checked for fairness, and easily implemented. However, efforts to incorporate machine learning into a pricing framework often lead to complex pricing policies which are not interpretable, resulting in slow adoption in practice. We present a customized, prescriptive tree-based algorithm that distills knowledge from a complex black-box machine learning algorithm, segments customers with similar valuations and prescribes prices in such a way that maximizes revenue while maintaining interpretability. We quantify the regret of a resulting policy and demonstrate its efficacy in applications with both synthetic and real-world datasets.

## 1. Introduction

To remain competitive, many firms are seeking to implement data-driven pricing policies. This has become possible due to the large amounts of relevant data firms are collecting, and through recent advances in machine learning. This data rich environment has also enhanced personalized pricing, where customers are offered a personalized price based on features that are predictive of their valuation of a product. A common example of personalized pricing is targeted offers through loyalty programs. While personalized pricing may potentially lead to higher revenue, currently only a quarter of retail companies are supportive of ML adoption, the lowest acceptance rate across surveyed industries.<sup>1</sup> One of the

---

<sup>1</sup>Darden School of Business, University of Virginia, Virginia, USA. <sup>2</sup>IBM Research, Yorktown Heights, New York, USA. Correspondence to: Max Biggs <biggs@vms.darden.virginia.edu>.

main concern is that the most accurate prediction models are often non-parametric functions which are considered as a black-box. Their opaqueness makes it difficult for firms to understand and trust the underlying mechanics of how the prices are being formed, which may potentially lead to fairness and legal issues. Another challenge with a personalized pricing policy is that it might be very complex, making it difficult and cumbersome for a firm to implement and to verify the prices are sensible.

In the recent years, there has been a surge of interest in making machine learning models more interpretable (Letham et al., 2015; Ustun & Rudin, 2016; Lakkaraju et al., 2016; Bertsimas et al., 2019a). A popular approach to gain interpretability is knowledge distillation (Hinton et al., 2015), where a simple model (the *student*) is trained on a transformed dataset, representing the distilled knowledge extracted by a complex model (the *teacher*). The distillation process often shows to bring the student closer to matching the predictive power of the teacher. Among many potential candidate models as the student (e.g., linear models and generalized additive models), decision tree stands out due to its ability to capture interactions between features, scalability to large datasets, and perhaps most importantly, providing “human-friendly” explanations to aid interpretability (Frosst & Hinton, 2017; Bastani et al., 2017; 2018; Miller, 2019).

However, the bulk of existing work on interpretability focuses on predictive models. There has been significantly less work on interpreting policies from these models when embedded in an operational decision-making context, which is the topic of this paper. It is crucial to note that it is not straightforward to adapt predictive models effectively to a prescriptive setting. Consider the problem of extracting interpretable pricing rules under a revenue-maximizing objective; a potential method of adapting the student-teacher framework is to first train a black-box teacher model that predicts demand (i.e., a response function that relates sales to price and other relevant features) and use this model to determine the best price with respect to the given features. Next, one can then approximate these prices by training an interpretable student model, e.g., regress on the prices with a decision tree. In the appendix <sup>2</sup>, we provide a simple ex-

---

<sup>2</sup><https://arxiv.org/abs/2007.01903>

ample which illustrates the pitfalls of this naive application of the student-teacher framework in the prescriptive setting. The key insight is that a partition of data which leads to good predictive accuracy (e.g., approximating given prices) does not necessarily translate to good prescriptive decisions (e.g., maximize revenue). Pricing data also has the challenge that past pricing decisions are made based on customer features and their perceived willingness to pay. Making decisions without properly controlling for these effects leads to sub-optimal revenue, as documented in [Bertsimas & Kallus, 2016](#).

To address these issues, we propose an interpretable method for generating personalized pricing policies, by adapting the student-teacher framework specifically for a decision-making setting and introducing a new tree algorithm – prescriptive student trees. To achieve this, we first train an accurate but opaque predictive teacher model, such as Boosted Trees or Neural Networks to predict the demand. Using this model we are able to produce an optimal price for each item, but the pricing policy is often complex and not interpretable. Instead of regressing on prices as in the naive approach discussed earlier, the prescriptive student model seeks to distill this pricing policy to one which is simpler and more interpretable through a customized recursive partitioning algorithm. Each leaf in the prescriptive student tree defines a set of items which are given the same price. These items share similarities in their covariates, and are chosen in such a way that each item in a leaf has a similar optimal price, as evaluated by the teacher model. In other words, the role of the teacher in the prescriptive setting is to estimate the counterfactual outcomes that are missing from the data, and control for observed confounding variables in the student tree construction process.

One example of an interpretable pricing policy produced by our algorithm is shown in [Figure 1](#), with eight segments. This particular example is a personalized pricing policy for strawberries and is explored further in the grocery pricing case study in [Section 4.3](#). Each segment is assigned a price and the expected revenue per shopper is also shown. As an example, it suggests \$1.99 for a single male with income below \$125k, but \$2.99 for an older shopper with kids and an income above \$125k.

Our key contributions are as follows: we propose a method for prescribing interpretable policies, where we introduce a new impurity measure and algorithm for constructing a student prescriptive tree by incorporating a teacher model. We quantify the regret of the resulting policy and demonstrate its efficacy in applications with both synthetic and real-world datasets. We want to point out that while we focus on revenue optimization as the running example, this method can be easily extended to many other prescriptive settings, such as precision medicine and personalized rec-

ommendation. More details will follow in [Section 3](#).

## 2. Related literature

Popularized by [Hinton et al., 2015](#), in the vanilla version of the knowledge distillation approach, the student model is trained on a dataset where the observed outcomes are replaced by soft-labels predicted by the teacher model. Variations on this idea have been extensively studied in deep learning and for other non-parametric models, for example, SHAP: [Lundberg & Lee, 2017](#), LIME: [Ribeiro et al., 2016](#), Anchors: [Ribeiro et al., 2018](#), RuleFit: [Friedman et al., 2008](#), also [Baehrens et al., 2010](#); [Sanchez et al., 2015](#); [Buciluă et al., 2006](#). Of particular interest to this work is [Craven & Shavlik, 1996](#); [Bastani et al., 2018](#), which use decision trees as the approximating model. Rather than using a teacher-student framework to approximate a prediction model, we adapt it to approximate a prescriptive policy.

There has been a recent focus within the operations research community on prescriptive analytics, i.e., how to use data and machine learning to make better operational decisions ([Ban & Rudin, 2019](#); [Bertsimas & Kallus, 2020](#); [Biggs et al., 2017](#); [Mišić, 2017](#); [Bertsimas & McCord, 2018](#); [Anderson et al., 2020](#)). This includes personalized pricing models, where user and/or product data is used to customize prices for products ([Chen et al., 2015](#); [Ferreira et al., 2016](#); [Javanmard & Nazerzadeh, 2016](#); [Bertsimas & Kallus, 2016](#); [Ye et al., 2018](#); [Baardman et al., 2018](#); [Elmachtoub et al., 2018](#); [Ban & Keskin, 2020](#)). However, these approaches are not explicitly concerned with interpretability, and indeed may result in complex decision policies with no segmentation or limit on the number of prices offered.

We highlight some tree-based prescriptive approaches which are closest to the current work. [Kallus, 2017](#) introduce personalization trees, a customized tree algorithm which proposes an impurity measure based on prescribing the treatment which has the highest outcome on average within each set in the partition. [Bertsimas et al., 2019b](#) extend this approach by incorporating predictive accuracy into the tree construction algorithm, and exploring more sophisticated optimization techniques. [Athey & Imbens, 2016](#) introduce a causal tree algorithm for estimating heterogeneous treatment effects for binary treatments. [Zhou et al., 2018](#) apply a doubly robust estimation approach to estimating reward of a policy. [Ciocan & Mišić, 2018](#) show a tree-based approach for solving optimal stopping problems. [Elmachtoub et al., 2020](#) recently provides a tree-based implementation of the predict-then-optimize approach first introduced in [Elmachtoub & Grigas, 2017](#). While this is a promising approach, it is not clear how to apply this to the pricing setting where the parameter being estimated is not just a function of exogenous covariates, but also price which is the decision being made.

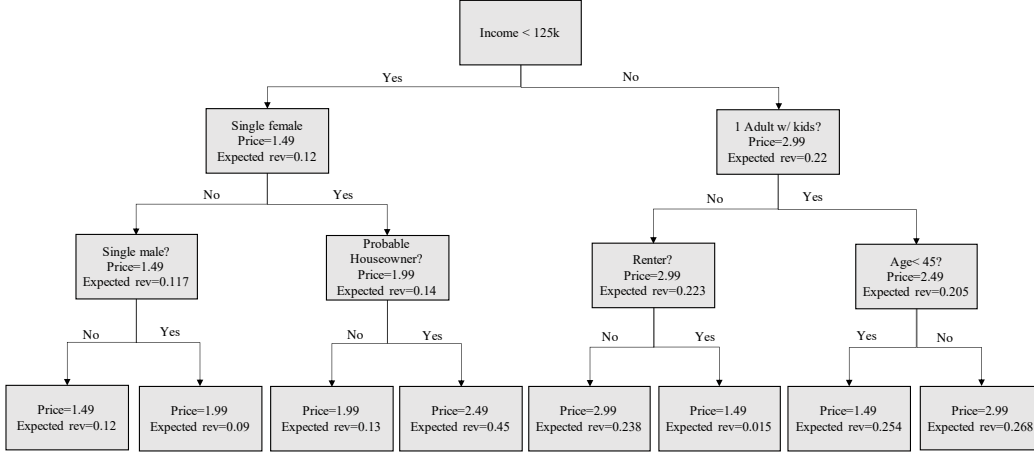


Figure 1: Interpretable personalized pricing algorithm for strawberries from Dunnhumby dataset

One implicit assumption behind the existing prescriptive tree-based approaches (Athey & Imbens, 2016; Kallus, 2017; Bertsimas et al., 2019b) is that the leaves are essentially homogeneous, so the confounding effects of the other (observed) variables are controlled for. This assumption means that the only difference between outcomes in the same leaf is due to the treatment assigned. This might be a reasonable assumption when the trees are very deep (i.e., resulting in smaller leaves where the covariates are similar, risking overfitting), but is unlikely to hold in shallow trees. In many prescriptive settings, shallow trees with fewer segments which translate into simpler pricing policies are desirable. By incorporating a teacher model to guide the pricing policy, our method controls for the observed confounding variables at any depth, rather than assuming they are the same, therefore ensuring that confounding effects are minimized. We also note that these approaches only work for discrete treatments, whereas our approach can be applied to continuous treatments.

### 3. Methodology

#### 3.1. Problem formulation

We assume we have  $n$  observational data samples  $\{(x_i, p_i, y_i)\}_{i=1}^n$ , where  $x_i \in \mathcal{X}^d$  are features which describe the  $i^{\text{th}}$  item,  $p_i \in R$  is the price assigned to the item, and  $y_i \in \{0, 1\}$  is whether the item sold (1) or not (0). For convenience of exposition, we refer to “item” as a selling unit of a product, where  $x_i$  may only contain product features as in the airline pricing case study in Section 4.4. It may also represent a (product, customer) pair when  $x_i$  also includes customer features, as in the Dunnhumby grocery store case study in Section 4.3.

As is common practice in the causal inference literature, we make the ignorability assumption (Hirano & Imbens,

2004), that is, there were no unmeasured confounding variables that affected both the choice of price and the outcome. Accordingly, we assume an underlying function  $f : \mathcal{X}^d \times R \rightarrow [0, 1]$  which maps the item features and price to a purchase probability  $f(x, p) = E[Y|X = x, P = p] = \mathbb{P}[Y = 1|X = x, P = p]$ . An optimal *personalized pricing* algorithm selects a price  $\tau^*(x)$  for each item based on its features to maximize the item’s expected revenue:

$$\tau^*(x) = \arg \max_{p \in R} pf(x, p) \quad (1)$$

In reality, the true response function  $f(x, p)$  is unknown, but can be estimated by solving an empirical risk minimization problem  $\hat{f} = \arg \min_{f \in \mathcal{F}} \sum_i^n L(x_i, p_i, y_i; f)$  over a class of functions  $\mathcal{F}$  and an appropriate loss function  $L$  for classification. We refer to  $\hat{f}$  as the teacher model.

If we substitute the proxy response function  $\hat{f}$  into the optimization in (1), the resulting personalized pricing policy may be too complex for a firm to verify and implement, with each feature vector potentially resulting in a different price. Furthermore, if the class of functions  $\mathcal{F}$  is complex, there may not be any insight into how the predictions are made. Our goal is to find an *interpretable personalized pricing* policy. We achieve this by restricting the pricing policy to the class of decision tree functions of depth  $k$ ,  $\mathcal{T}(k)$ , which we refer to as the prescriptive student model, and find the policy which achieves the highest revenue as predicted by the teacher model:

$$\max_{\tau(x) \in \mathcal{T}(k)} \frac{1}{n} \sum_{i=1}^n \tau(x_i) \hat{f}(x_i, \tau(x_i)) \quad (2)$$

A decision tree (see Breiman et al., 1984) is a class of functions which recursively partitions the feature space into leaves, each of which is associated with a prediction. The binary tree structure is defined by internal nodes and leaves.

At each internal node there is an axis-aligned split which checks if a feature is less than a threshold. If this condition is met when mapping a data point to a leaf then it proceeds to the left child node. Otherwise, it continues to the right child node until a leaf is reached.

Contrary to the well-studied predictive setting where each leaf has a prediction of the outcome, the leaves of the prescriptive tree specify a single price at which to sell the items. Decision trees result in interpretable pricing policies because specifying the depth of the tree restricts the number of unique prices in the policy. Furthermore, the set of items which are prescribed a price is defined by the single variable splits which lead to the leaf. Each of these rules can be easily verified by a practitioner. An example of a tree-based pricing policy is shown in Figure 1.

While we focus on pricing, the framework we propose can be easily adapted to broad application in prescriptive settings with a very minor change to Eq 1 from maximizing revenue  $pf(x, p)$  to maximizing any known function of a predictive model  $g(f(x, p))$ . Consider a Warfarin dosing application in personalized medicine (Kallus, 2017; Bastani et al., 2020). We can define a criterion to maximize the probability that a correct dosage is given; for personalized recommendations for a job training program (Kallus, 2017), the criterion can be to maximize the net income after enrollment cost.

### 3.2. Approximation guarantees

We show that the regret of a tree-based interpretable pricing policy optimized over the teacher model can be bounded relative to the optimal personalized pricing policy. Here the regret is defined as the difference in the expected revenue between the two policies. To do so, we assume uniform bounds on the error of the estimated teacher model and that the true expected revenue function is smooth. For convenience of exposition, we also assume  $x \in [0, 1]^d$ .

**Assumption 1** *Estimation error uniformly bounded:*  $|f(x, p) - \hat{f}(x, p)| \leq \mathcal{K}(n), \forall x \in [0, 1]^d, p \in R$

An example where assumption 1 is met is Honest trees and forests from Wager & Athey, 2018.

**Assumption 2**  $pf(x, p)$  is Lipschitz continuous in  $x$  with respect to the  $L_2$  norm:

$$pf(x, p) - pf(x', p) \leq L\|x - x'\|_2, \quad \forall x, x' \in [0, 1]^d$$

**Theorem 1** *Under the above assumptions, there exists a tree-based policy  $\hat{\tau}(x) \in \mathcal{T}(k)$ , trained on the proxy response function  $\hat{f}$  such that:*

$$\tau^*(x)f(x, \tau^*(x)) - \hat{\tau}(x)f(x, \hat{\tau}(x)) \leq 2^{-\frac{k}{d}+1}L\sqrt{d} + 2\mathcal{K}(n) \quad \forall x \in [0, 1]^d \quad (3)$$

The proof is given in the appendix. This theorem shows that the revenue gap between the tree-based pricing policy and the optimal personalized policy can be decomposed into an approximation error and an estimation error. The estimation error is inherited from the estimation error of the teacher model. The approximation error depends on the smoothness of the function and the dimension of the feature space, and decays rapidly with the depth of the tree.

### 3.3. Recursive partitioning algorithm for student prescriptive trees

We propose a simple algorithm for solving (2). Considering the task of constructing an optimal classification tree is known to be NP-complete (Laurent & Rivest, 1976), we use a heuristic based on the recursive partitioning (Breiman et al., 1984), whereby splits are chosen greedily. We achieve this by proposing a new splitting criterion, which utilizes the teacher model. Denote  $S_l \subseteq [n]$  as the subset of observations which belong to leaf  $l$  of the student tree. We define the *revenue maximization criterion* as follows:

$$\mathcal{R}(S_l) = \max_p \sum_{i \in S_l} p\hat{f}(x_i, p) \quad (4)$$

At each node in the tree, we consider a decision split partitioning the data into two sets  $S_1(j, s) = \{i \in [n] | x_{i,j} \leq s\}$  and  $S_2(j, s) = \{i \in [n] | x_{i,j} > s\}$ , where  $x_{i,j}$  is the  $j^{\text{th}}$  feature of observation  $i$ . We choose the split which results in maximum predicted revenue by giving each set formed a different price. We do this by optimizing over  $j$ , the feature chosen to split on, and  $s$ , the threshold, i.e.,  $\max_{j,s} \mathcal{R}(S_1(j, s)) + \mathcal{R}(S_2(j, s))$ .

On the training data this is limited to searching the  $d$  dimensions of the feature vector, while there are at most  $n$  unique splits in the data, one for each data point. We continue recursively splitting until a specified criteria has been met. We use tree depth as termination criteria which corresponds to the desired complexity of the pricing policy, but can easily incorporate other commonly used approaches such as minimum leaf size criterion, or minimum improvement in impurity/revenue. In each leaf, the price assigned is the maximizer of (4).

In many pricing applications, there is a discrete set of prices that can be chosen from,  $p \in \mathcal{P} = \{p_1, p_2, \dots, p_m\}$ . With this simplification, the estimated revenue for each item at each price  $r_{i,k} = p_k \hat{f}(x_i, p_k)$ ,  $k \in [m]$  can be calculated prior to tree construction. The revenue for each segment can be calculated and used instead of (4):

$$\mathcal{R}_m(S_l) = \max_{k \in [m]} \sum_{i \in S_l} r_{i,k} \quad (5)$$

We note that there are non-greedy approaches for constructing trees which could be adapted to solve (2), such as Bertsimas et al., 2019b. However, experiments in Bastani et al.,



2018 have shown that greedy trees are more intuitive as the variables leading to the largest improvement are split on first, matching human reasoning and providing more interpretability. Furthermore greedy algorithms are known to scale better and solve more quickly for large datasets.

## 4. Experimental results

### 4.1. Benchmarks

We compare our approach with two other state-of-the-art prescriptive tree-based algorithms. Personalization trees (Kallus, 2017), prescribe the treatment which has the highest average outcome in each leaf. This algorithm can be adapted to a personalized pricing setting by using the observed revenue as the outcome for each item. More specifically, if there are  $m$  prices  $t \in [m]$ , we calculate the impurity for a set of items  $S_t$  as follows, and use this as the splitting criterion in a greedy algorithm:

$$\mathcal{I}(S_t) = \max_{t \in [m]} \frac{\sum_{i \in S_t} p_i y_i \mathbb{1}\{p_i = t\}}{\sum_{i \in S_t} \mathbb{1}\{p_i = t\}} \quad (6)$$

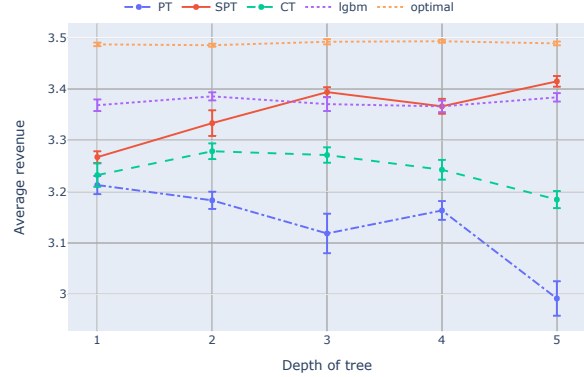
We also benchmark our algorithm against the causal trees algorithm from Athey & Imbens, 2016. The approach estimates heterogeneous treatment effects, rather than prescribing the optimal treatment. In particular, it only considers estimation of CATE (conditional average treatment effect) for binary treatments  $\delta(x) = E[Y|X = x, T = 1] - E[Y|X = x, T = 0]$ . Following the testing procedure in Kallus, 2017, we adapt the causal tree algorithm to prescribing the optimal treatment from a discrete set of alternatives using the “one-vs.-all” approach. In this approach, the treatment effect of each price is estimated by comparing the expected outcome of those items which received the price to those which received another price. Specifically, for each price  $t \in [m]$ , the treatment effect  $\delta(x, t)^{1vall} = E[Y|X = x, P = t] - E[Y|X = x, P \neq t]$  is estimated using a causal tree and the prescribed treatment is the one which achieves the highest improvement, i.e.,  $\max_{t \in [m]} \delta(x, t)^{1vall}$ . Note that this approach is no longer as interpretable as a single tree, given that there is no guarantee that the tree partitions associated with each treatment will align, resulting in a loss of the recursive tree structure and significantly more complex policies

To train the causal trees, we use the implementation from Athey & Imbens, 2016, using default parameter values. Due to the lack of open-source code for Kallus, 2017, we implement the impurity measure (6) in our own recursive partitioning procedure.

### 4.2. Results on simulated datasets

Simulated datasets allow us to accurately evaluate the counterfactual outcomes associated with changing prices, and

calculate the resulting revenue from a policy, since the underlying probability model of simulated dataset is known. The



(a) Revenue with varying tree depth



(b) Revenue with varying training data size

Figure 2: Experiment results on synthetic Dataset 4

datasets we examine come from the following generative model:

$$Y^* = g(X) + h(X)P + \epsilon, \quad Y = \begin{cases} 1, & \text{if } Y^* > 0 \\ 0, & \text{if } Y^* \leq 0 \end{cases} \quad (7)$$

- Dataset 1: linear probit model with no confounding:  $g(X) = X_0$ ,  $h(X) = -1$  and  $X \sim N(5, I_2)$  and  $P \sim N(5, 1)$ .
- Dataset 2: higher dimension probit model with sparse linear interaction  $g(X) = 5$ ,  $h(X) = -1.5(X'\beta)$ ,  $\{X_i\}_{i=1}^{20}$ ,  $\{\beta_i\}_{i=1}^5$ ,  $\epsilon_i \sim N(0, 1)$ ,  $P_i \sim N(0, 2)$ ,  $\{\beta_i\}_{i=6}^{20} = 0$ , where the purchase probability is only dependent on the first 5 features.
- Dataset 3: probit model with step interaction:  $g(X) = 5$ ,  $h(X) = -1.2\mathbb{1}\{X_0 < -1\} - 1.1\mathbb{1}\{-1 \leq X_0 < 0\} - 0.91\mathbb{1}\{0 \leq X_0 < 1\} - 0.81\mathbb{1}\{1 \leq X_0\}$ .
- Dataset 4: probit model with multi-dimensional step interaction:  $g(X) = 5$ ,  $h(X) = -1.25\mathbb{1}\{X_0 <$

Table 1: Revenue comparison on synthetic datasets

Data	Mean revenue				Maximum revenue			Minimum revenue		
	Optimal	PT	SPT	CT	PT	SPT	CT	PT	SPT	CT
1	3.28	3.07	<b>3.19</b>	3.13	3.24	<b>3.27</b>	3.26	2.88	<b>3.01</b>	2.99
2	2.94	1.75	<b>2.14</b>	1.70	1.93	<b>2.27</b>	1.79	1.58	<b>1.95</b>	1.62
3	3.42	3.20	<b>3.34</b>	3.21	3.36	<b>3.42</b>	3.37	2.78	<b>3.24</b>	3.02
4	3.49	3.13	<b>3.36</b>	3.24	3.29	<b>3.46</b>	3.33	2.82	<b>3.20</b>	3.08
5	3.35	3.11	<b>3.26</b>	3.06	3.29	<b>3.33</b>	3.26	2.79	<b>3.12</b>	2.47
6	2.58	2.01	<b>2.40</b>	2.12	2.34	<b>2.48</b>	2.41	1.53	<b>2.22</b>	1.87

$$-1\} - 1.1\mathbb{1}\{-1 \leq X_0 < 0\} - 0.9\mathbb{1}\{0 \leq X_0 < 1\} - 0.75\mathbb{1}\{1 \leq X_0\} - 0.1\mathbb{1}\{X_1 < 0\} + 0.1\mathbb{1}\{X_1 \geq 0\}.$$

- Dataset 5: linear probit model with observed confounding:  $g(X) = X_0, h(X) = -1, X \sim N(5, I_2)$ .
- Dataset 6: probit model with non-linear interaction:  $g(X) = 4|X_0 + X_1|, h(X) = -|X_0 + X_1|$ .

We set  $X = (X_0, X_1) \sim N(0, I_2), P \sim N(X_0+5, 2), \epsilon \sim N(0, 1)$ , i.i.d. unless otherwise mentioned. Dataset (1) and (5) are a common linear demand model used in the pricing literature, with and without observed confounding effects. Datasets (2)-(4) and (6), have heterogeneity in treatment effects. A visualization of all datasets can be found in the appendix. All datasets except dataset (1) and (2) have confounding of observed features, where the price observed is dependent on the features of the item. For the teacher model, we train a gradient boosted tree ensemble model using the lightGBM package (Ke et al., 2017). We use default parameter values, with 50 boosting rounds. The discretized price set is set to 9 prices, ranging from the 10<sup>th</sup> to the 90<sup>th</sup> percentile of observed prices in 10% increments.

We compare the personalization tree (PT), causal tree (CT) and student prescriptive tree (SPT) in terms of their expected revenue, along with the fully personalized policy of the teacher model (lgbm). We also benchmark against the true optimum policy (optimal), which can be found by identifying the price which results in the highest revenue according to the underlying probability model for each observation, and as such may vary depending on the observations in the sample.

We explore how the expected revenue changes with the depth of a tree ( $k = \{1, 2, 3, 4, 5\}$ ). The number of training samples is held constant at  $n = 5000$ . For each tree depth, we run 10 independent simulations for each dataset. Table 1 summarizes the best, the worst, and the average over the simulations of varying tree depth. We also compare their performance as the size of the dataset changes ( $n = \{100, 300, 1000, 3000, 10000\}$ ), with the tree depth set to  $k = 3$ , again for 10 independent samples of each dataset. In

all experiments each tree was grown to the full width for a given depth, i.e.,  $2^k$  leaves for a tree of depth  $k$ .

As an illustrative example, we show the performance on dataset (4) in Figure 2a and 2b, while detailed results for changing depth and dataset size for individual datasets can be found in the appendix. We observe that SPT generally improves as the tree gets deeper, whereas PT and CT appear to perform worse. Moreover, while all methods improve as the training data size increases, the student tree model significantly outperforms other methods on small sample sizes. One explanation is that when the trees are deep or the training size is small, the number of observations per action/price is also small. This increases the variance of the estimation of the outcome of that action and makes it less likely that the action with the highest average revenue is the one with the strongest signal, rather than noise. Having a teacher model can help reduce some of this noise within the data for small leaves, compared to PT and CT which only use the outcomes in that leaf. We also observe that the SPT appears to perform relatively better than PT or CT in dataset 5 compared to dataset 1, indicating that the teacher model also helps control for observed confounding effects. We conjecture that this is due to violations of the assumption PT and CT make that each leaf is sufficiently small that they are effectively controlling for  $x$  and therefore avoiding confounding effects. Observing the trees resulting from experiments, it is clear that this is not satisfied for the shallow interpretable trees we study and confounding effects lead to poor prices. Our model avoids this by using the teacher model to provide an estimate of selling probability, conditioned on  $x$  (and therefore controlling for  $x$ ). Furthermore the student model (SPT) often outperforms the teacher model (lgbm), suggesting it is less prone to overfitting given inherently simpler functional form.

We also observe that the SPT algorithm is relatively unaffected by the choice of price discretization relative to PT and CT. As the discretization becomes more granular, PT and CT are much more prone to overfitting, as the number of observations in the price bucket decreases. However, SPT seems less prone to this as the predictions from teacher model appears to be more smooth and stable, as we would

minsplit	SPT		PT		CT		lgbm	OPT
	mean rev	n_leaf	mean rev	n_leaf	mean rev	n_leaf*	mean rev	mean rev
50	3.39 (0.012)	120.4	2.29 (0.043)	158.2	2.92 (0.020)	108.6	3.37 (0.012)	3.49
150	3.40 (0.014)	64	2.55(0.040)	66.9	3.03 (0.020)	49.4	3.39 (0.010)	3.49
500	3.39 (0.013)	24.6	2.98 (0.037)	23.4	3.23 (0.013)	15.6	3.37(0.011)	3.49
1500	3.37 (0.011)	5.4	3.19 (0.043)	5.9	3.28 (0.010)	4.5	3.39 (0.019)	3.49

Table 2: Revenue comparison minsplit termination criterion for synthetic dataset 4 (n=5000)

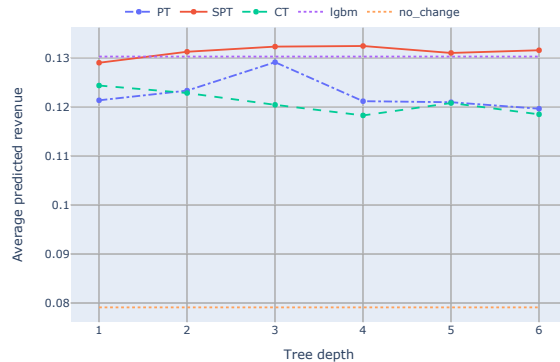
expect from a fitted line vs raw observations.

To show that our splitting criteria performs well with alternative termination criteria and to provide further insight into the interpretability of our approach, we show results where the termination criteria is determined by minsplit – if the number of observations at a potential split is less than a threshold, no further splits will occur on that branch. This also provides a test instance for deeper trees rather than wide trees. Table 2 summarizes the experiment results over 10 randomly generated instances, with standard error in the parenthesis. It shows that our method outperforms the tree benchmarks on average, while the trees/policies are of a comparable complexity as shown by n\_leaf column. We emphasize that the “one-vs.-all” causal tree is significantly less interpretable since each policy is formed by multiple trees. We also show our approach outperforms a naive implementation of a student-teacher approach, where we train a predictive teacher model that learns demand, then train a predictive student model to approximate the teacher, and lastly optimize to find a pricing policy based on the student model in the appendix.

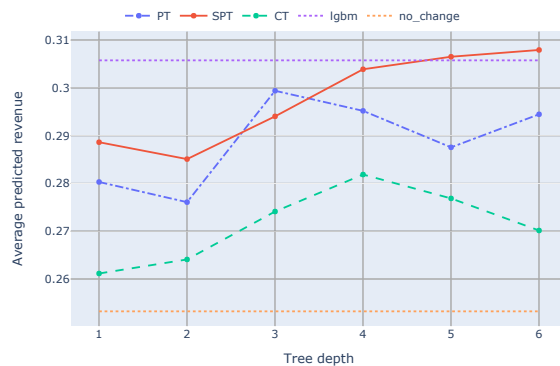
### 4.3. Dunnhumby grocery store case study

We benchmark the algorithms on a publicly available retail dataset collated by the analytics firm Dunnhumby.<sup>3</sup> “The complete journey” dataset contains household level transactions over two years from a group of 2,500 households who are frequent shoppers at a grocery retailer, along with demographic information on each household. The customer features used in the personalized pricing algorithm are the discretized age of the shopper, discretized household income, whether they are a homeowner or renter, and the composition of the household (single male, single female, 2 adults no kids, 2 adults with kids, 1 adult with kids, or unknown). The data is processed into a format where each row corresponds to a shopping trip for a customer parameterized by their features, with a label to indicate whether the item was purchased at a listed price. More details on the data pre-processing procedure can be found in the appendix. We focus on strawberries (1 pound) and milk (1 gallon) for the case study due to their relatively high sales

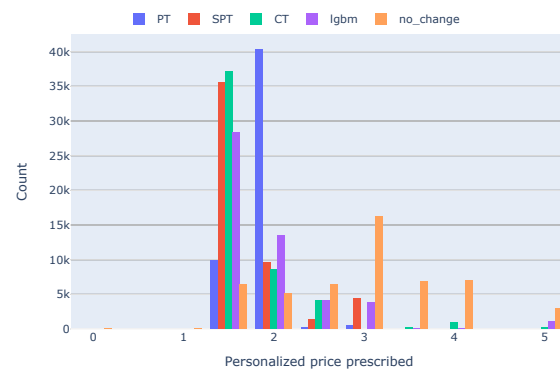
<sup>3</sup><https://www.dunnhumby.com/careers/engineering/sourcefiles>



(a) Predicted revenue on strawberries



(b) Predicted revenue on milk



(c) Prices prescribed for strawberries

Figure 3: Experiments from Dunnhumby dataset

and price variation which allows the models to accurately capture the price-demand relationship. Once processed, the strawberry dataset has 102080 shopping trips, with 3373 instances where strawberries are purchased, while milk has 89936 trips with 7688 purchases.

Due to the lack of counterfactuals in the dataset, the efficacy of the pricing algorithms are evaluated using counterfactuals estimated from an independent lightGBM model. That is, for every price prescribed, the evaluator model predicts what the selling probability will be and calculates the expected revenue. We divide the dataset in half. The evaluator model is trained on a dataset which is independent from the dataset used for the teacher model and the tree-based algorithms. This approach for evaluating prescriptive algorithms is used in Biggs et al., 2017. Both the evaluator and teacher lightGBM model achieve an out-of-sample AUC of 0.79 for strawberries and 0.74 for milk with 50 boosting rounds and default classification parameters.

Figure 3a and 3b show the performance of the pricing algorithms, the teacher model (lgbm) and the current pricing policy (no\_change) in terms of predicted revenue over a range of different depth trees. SPT is able to outperform the other tree approaches and substantially improve upon the current pricing strategy with the predicted revenue per customer with a 67% increase for strawberries and a 22% improvement for milk, with a suitably chosen depth.

An example of the interpretable pricing policy for strawberries using SPT ( $k = 3$ ) is shown in Figure 1. The model identifies that low income families are the most price sensitive and suggests a low price of \$1.49. High income shoppers who are not renters are given a high price \$2.99. The distribution of the prescribes prices corresponding to different pricing policies are shown in Figure 3c.

#### 4.4. Airline pricing case study

We worked with a large international airline and tested the prescriptive student tree method for pricing first class tickets. Interpretability is paramount as airlines need to file pricing rules with Airline Tariff Publishing Company to distribute and broadcast fares over the travel service network. A potential rule may look like the following: “if the fare is between SFO and JFK, advance purchase window is less than 7 days and the flight departs on a weekday before 11am, prescribe a price of \$400”. As regulation mandates that prices cannot differ based on customer features, the pricing algorithm works as an automated product differentiator by prescribing different prices to different tickets based on features such as advance booking window, time of the the flight, the origin and destination, distance of flight, the price of the main and basic cabin tickets, inventory level, whether the flight is one-way or round trip, etc. In total there are 65 features to train a teacher model, only 6 features are used to train a

SPT for each route: advance purchase window, inventory levels of the main and upgrade cabins, round trip indicator, whether a flight includes a Saturday night stay, and duration of a flight. Another requirement is to limit the number of fare rules. Hence, SPTs with  $k = 3$  or 4 are selected for each route.

For confidentiality reasons, some details of the study are omitted. The dataset used has 1,144,099 observations. A lightGBM model with 1000 rounds of boosting is used as the teacher model. The first class prices considered are split into \$5.00 increments over the range of prices observed in the historical data. Figure 4 shows the predicted revenue of our proposed method to the other benchmarks, training on a 10% ( $n=114,400$ ) sample and evaluated using a lightGBM model trained on the remaining 90% of the dataset. The student model is able to outperform the methods we benchmark against. Moreover, it is capable of achieving significant improvement over the current pricing with just a shallow tree.



Figure 4: Predicted revenue airline pricing

## 5. Conclusion

Our paper focuses on a research area which has received far less attention in the literature despite its paramount practical importance, i.e., generating interpretable policies for decision making. To address this issue, we proposed a method of knowledge distillation for the prescriptive setting by adapting the student-teacher framework. A high-capacity teacher model is incorporated to guide the student tree construction process as it estimates the counterfactual outcomes that are missing from the data, and controls for observed confounding variables. With a customized recursive partitioning algorithm, the prescriptive student tree is capable of simultaneously creating segments of customers with similar valuations and determining the optimal revenue-maximizing price for a segment. We quantified the performance of our algorithm in terms of the regret. Experimental results on both synthetic and real datasets showed that our approach outperforms comparable methods. Moreover, the prescriptive



student tree which produced interpretable rules is capable of approximating the opaque teacher model with relatively shallow trees.

While we focus on making better and interpretable pricing decisions in this paper, this framework can be easily adapted to broad prescriptive applications by replacing the splitting criterion in the student tree construction algorithm with an appropriate application-specific objective. As techniques such as knowledge distillation in the predictive setting have enabled a proliferation of deep learning models to be deployed outside their traditional fields such as computer vision and natural language processing (Che et al., 2015; Ding et al., 2018; Tang & Wang, 2018; Pan et al., 2019; Vongkulbhisal et al., 2019; Liu et al., 2020), it is our hope that the proposed prescriptive student-teacher framework will facilitate a faster ML adoption across industries, by enabling practitioners make better business decisions and gaining more stakeholder buy-ins with interpretable explanations behind those decisions.

## References

- Anderson, R., Huchette, J., Ma, W., Tjandraatmadja, C., and Vielma, J. P. Strong mixed-integer programming formulations for trained neural networks. *Mathematical Programming*, pp. 1–37, 2020.
- Athey, S. and Imbens, G. Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences*, 113(27):7353–7360, 2016.
- Baardman, L., Borjian Boroujeni, S., Cohen-Hillel, T., Pan-chamgam, K., and Perakis, G. Detecting customer trends for optimal promotion targeting. *Available at SSRN 3242529*, 2018.
- Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., and MÅžller, K.-R. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11(Jun):1803–1831, 2010.
- Ban, G.-Y. and Keskin, N. B. Personalized dynamic pricing with machine learning: High dimensional features and heterogeneous elasticity. *Available at SSRN 2972985*, 2020.
- Ban, G.-Y. and Rudin, C. The big data newsvendor: Practical insights from machine learning. *Operations Research*, 67(1):90–108, 2019.
- Bastani, H., Bastani, O., and Kim, C. Interpreting predictive models for human-in-the-loop analytics. *arXiv preprint arXiv:1705.08504*, pp. 1–45, 2018.
- Bastani, H., Bayati, M., and Khosravi, K. Mostly exploration-free algorithms for contextual bandits. *Management Science*, 2020.
- Bastani, O., Kim, C., and Bastani, H. Interpreting blackbox models via model extraction. *arXiv preprint arXiv:1705.08504*, 2017.
- Bertsimas, D. and Kallus, N. The power and limits of predictive approaches to observational-data-driven optimization. *arXiv preprint arXiv:1605.02347*, 2016.
- Bertsimas, D. and Kallus, N. From predictive to prescriptive analytics. *Management Science*, 66(3):1025–1044, 2020.
- Bertsimas, D. and McCord, C. Optimization over continuous and multi-dimensional decisions with observational data. In *Advances in Neural Information Processing Systems*, pp. 2962–2970, 2018.
- Bertsimas, D., Delarue, A., Jaillet, P., and Martin, S. The price of interpretability. *arXiv preprint arXiv:1907.03419*, 2019a.
- Bertsimas, D., Dunn, J., and Mundru, N. Optimal prescriptive trees. *INFORMS Journal on Optimization*, 1(2): 164–183, 2019b.
- Biggs, M., Hariss, R., and Perakis, G. Optimizing objective functions determined from random forests. *Available at SSRN 2986630*, 2017.
- Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. *Classification and regression trees*. CRC press, 1984.
- Buciluř, C., Caruana, R., and Niculescu-Mizil, A. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 535–541, 2006.
- Che, Z., Purushotham, S., Khemani, R., and Liu, Y. Distilling knowledge from deep networks with applications to healthcare domain. *arXiv preprint arXiv:1512.03542*, 2015.
- Chen, X., Owen, Z., Pixton, C., and Simchi-Levi, D. A statistical learning approach to personalization in revenue management. *Available at SSRN 2579462*, 2015.
- Ciocan, D. and Miřić, V. Interpretable optimal stopping. 2018.
- Craven, M. and Shavlik, J. W. Extracting tree-structured representations of trained networks. In *Advances in neural information processing systems*, pp. 24–30, 1996.
- Ding, T., Hasan, F., Bickel, W. K., and Pan, S. Interpreting social media-based substance use prediction models with knowledge distillation. In *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 623–630. IEEE, 2018.

- Elmachtoub, A. N. and Grigas, P. Smart "predict, then optimize", 2017.
- Elmachtoub, A. N., Gupta, V., and Hamilton, M. The value of personalized pricing. Available at SSRN 3127719, 2018.
- Elmachtoub, A. N., Liang, J. C. N., and McNellis, R. Decision trees for decision-making under the predict-then-optimize framework, 2020.
- Ferreira, K. J., Lee, B. H. A., and Simchi-Levi, D. Analytics for an online retailer: Demand forecasting and price optimization. *Manufacturing & Service Operations Management*, 18(1):69–88, 2016.
- Friedman, J. H., Popescu, B. E., et al. Predictive learning via rule ensembles. *The Annals of Applied Statistics*, 2(3):916–954, 2008.
- Frosst, N. and Hinton, G. Distilling a neural network into a soft decision tree. *arXiv preprint arXiv:1711.09784*, 2017.
- Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.
- Hirano, K. and Imbens, G. W. The propensity score with continuous treatments. *Applied Bayesian modeling and causal inference from incomplete-data perspectives*, 226164:73–84, 2004.
- Javanmard, A. and Nazerzadeh, H. Dynamic pricing in high-dimensions. *arXiv preprint arXiv:1609.07574*, 2016.
- Kallus, N. Recursive partitioning for personalization using observational data. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1789–1798. JMLR. org, 2017.
- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in neural information processing systems*, pp. 3146–3154, 2017.
- Lakkaraju, H., Bach, S. H., and Leskovec, J. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1675–1684, 2016.
- Laurent, H. and Rivest, R. L. Constructing optimal binary decision trees is np-complete. *Information processing letters*, 5(1):15–17, 1976.
- Letham, B., Rudin, C., McCormick, T. H., Madigan, D., et al. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics*, 9(3):1350–1371, 2015.
- Liu, D., Cheng, P., Dong, Z., He, X., Pan, W., and Ming, Z. A general knowledge distillation framework for counterfactual recommendation via uniform data. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 831–840, 2020.
- Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, pp. 4765–4774, 2017.
- Miller, T. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38, 2019.
- Mišić, V. V. Optimization of tree ensembles. *arXiv preprint arXiv:1705.10883*, 2017.
- Pan, Y., He, F., and Yu, H. A novel enhanced collaborative autoencoder with knowledge distillation for top-n recommender systems. *Neurocomputing*, 332:137–148, 2019.
- Ribeiro, M. T., Singh, S., and Guestrin, C. "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*, pp. 1135–1144, 2016.
- Ribeiro, M. T., Singh, S., and Guestrin, C. Anchors: High-precision model-agnostic explanations. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- Sanchez, I., Rocktaschel, T., Riedel, S., and Singh, S. Towards extracting faithful and descriptive representations of latent variable models. In *AAAI Spring Symposium on Knowledge Representation and Reasoning (KRR): Integrating Symbolic and Neural Approaches*, volume 1, pp. 4–1, 2015.
- Tang, J. and Wang, K. Ranking distillation: Learning compact ranking models with high performance for recommender system. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2289–2298, 2018.
- Ustun, B. and Rudin, C. Supersparse linear integer models for optimized medical scoring systems. *Machine Learning*, 102(3):349–391, 2016.
- Vongkulbhisal, J., Vinayavekhin, P., and Visentini-Scarzanella, M. Unifying heterogeneous classifiers with distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3175–3184, 2019.
- Wager, S. and Athey, S. Estimation and inference of heterogeneous treatment effects using random forests. *Journal*

*of the American Statistical Association*, 113(523):1228–1242, 2018.

Ye, P., Qian, J., Chen, J., Wu, C.-h., Zhou, Y., De Mars, S., Yang, F., and Zhang, L. Customized regression model for airbnb dynamic pricing. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 932–940, 2018.

Zhou, Z., Athey, S., and Wager, S. Offline multi-action policy learning: Generalization and optimization. *arXiv preprint arXiv:1810.04778*, 2018.