



# Model Based Recursive Partitioning for Customized Price Optimization Analytics

Jorge M. Arevalillo<sup>(✉)</sup>

Department of Statistics, Operational Research and Numerical Analysis,  
University Nacional Educación a Distancia,  
Paseo Senda del Rey 9, 28040 Madrid, Spain  
jmartin@ccia.uned.es

**Abstract.** Pricing is a relevant topic in revenue management that has awakened interest of researchers, practitioners and analysts in companies whose managerial decisions are supported by data-driven intelligent systems. This paper addresses the issue using an approach that combines model based recursive partitioning with price optimization in order to identify groups with differential price sensitivity, where optimal price allocation can be derived at a customer level. The approach is validated by application to the business case of an on-line auto lending company taking the interest rate as the price variable. The model based recursive method is used to get a tree that allows to estimate differential bid response functions across its terminal nodes; the tree is fitted on a training data set. The estimated bid response functions are combined with the bid revenue, calculated from data collected by loan applications, in order to carry out optimal price allocation maximizing the expected revenue. The expected revenue is compared on an independent test sample data set with the actual un-optimized revenue and with the revenue obtained by optimal price allocation using the standard *Logit* estimation of the bid response function; the proposed approach gives promising results that highlight new business opportunities.

**Keywords:** Pricing analytics · Revenue management ·  
Model based recursive partitioning · Optimal price allocation

## 1 Introduction

Pricing analytics is a relevant issue for revenue management in organizations that aim to develop their pricing strategies on the basis of data-driven decisions that use different layers of information like product characteristics, the purchase habits of customers, their socioeconomic and demographic attributes and some other related business inputs. Due to the increasing interest among business analysts, data scientists, and financial and revenue managers in data-driven pricing approaches, the issue has become a hot study topic [4, 5, 13, 18] with cross-sectorial applications in the banking sector [2, 19], insurance [8, 9],

the hospitality industry [3, 10] and the airlines business [7, 16, 17], just to give a non-exhaustive list with a few representative business applications.

In this paper we address the issue of optimal price allocation from a data-driven viewpoint: The model based recursive partitioning (MOB) machine learning method [12, 21], that allows to assess price sensitivity (PS) by means of the estimation of differential bid response functions, is used in combination with revenue optimization in order to calculate the expected revenue obtained at the optimal bid prices. The application of MOB for price sensitivity assessment is not new; actually, this paper is partially inspired in a recent work by [1] and serves as a complement of the results therein. In this paper the overall pricing picture is complemented by showing how the outputs provided by the MOB method can be used as business inputs to address the optimal price allocation problem on the basis of historical data.

The paper is organized as follows: The next section gives an overview on the decision trees, which includes classification and regression trees, conditional inference trees and the MOB method. Section 3 shows its role for addressing the optimal price allocation problem; when applied to historical data, it allows to identify groups with differential bid response functions; such differential functions will be used as inputs to solve the customized price optimization problem. In Sect. 4 we evaluate and validate the MOB approach by means of an application to an on-line auto lending company; the resulting revenue results are compared with un-optimized actual revenue and with the revenue obtained by optimal price allocation using the standard *Logit* estimation of the bid response function. The paper finishes with a section of concluding remarks that summarizes the approach and recapitulates the main findings.

## 2 The Decision Tree Modeling Approach

This section reviews some background about decision trees, as they are the instrumental tools of this work.

The decision tree methodology is a data-driven approach for the recursive partitioning of a data set by means of the search of splitting points within a set of segmentation variables collected in an input vector  $\mathbf{Z}$ . The splits are found by means of criteria that allow to quantify the relationship between the inputs and a given outcome variable  $Y$ . Essentially, the partitioning of the data responds to the construction of a segmentation guided by the outcome variable, which is carried out in a recursive fashion. Although there exists a large amount of algorithms that implement the method [15], here we focus on the classification and regression tree (CART) [6] and the conditional inference tree (CTREE) [11] algorithms, as they are two widely used tree methods.

### 2.1 CART Algorithm

CART is a classification and regression tree algorithm invented by [6]. The algorithm recursively segments the data through binary splits that generate a tree

structure in which child nodes represent the binary partition obtained by splitting each parent node. The splits are generated by assessing the impurity of the outcome variable at parent and descendant nodes using different impurity measures [6]. CART explores the set of input variables and looks for the variable and splitting point that maximizes the impurity decrease in the left and right descendants. CART decision trees are grown in a recursive way until a large tree structure is obtained, usually a tree with a minimum number of cases in the terminal nodes; then automated pruning of such a large tree is carried out on a test sample data by means of an intelligent strategy that eliminates uninformative branches and avoids overfitting. The resulting tree is the tradeoff between model complexity and predictive accuracy.

For additional details about CART tuning controls and some other functionalities of the algorithm, the reader is referred to the pioneer monograph [6]. An easy to use implementation of CART is provided by the *rpart* R package [14].

## 2.2 CTREE Algorithm

One weakness of CART is its bias towards the selection of splitting variables with many categories [6]. Unlike CART, the CTREE algorithm provides an alternative approach to overcome such a bias problem [11]. It takes the  $p$ -value obtained by permutation tests that use function-based statistics of the inputs as a criterion to find the best cutoff point; the  $p$ -values are calculated by asymptotic approximations or by Monte Carlo simulations. Although CTREE can control the splitting bias, it has the disadvantage of not having a pruning strategy like CART; so usually the stopping rule must be set in advance by the expert: it may consist of a threshold for the significance level of the aforementioned tests, above which a node is declared as terminal (its default is  $\alpha = 0.05$ ), or alternatively, a minimum size for the descendant nodes.

The *party* R package [12] provides an implementation of CTREE, along with other handy graphical functionalities of the algorithm.

## 2.3 Model Based Recursive Partitioning

The model based recursive partitioning method rests upon the decision tree methodology; its goal is twofold: firstly, the segmentation of a data set guided by a given outcome variable, and secondly, the fit of a parametric model in the terminal nodes derived by the tree partitioning mechanism so that the parametric fit of the model is embedded in the tree construction [21]. In order to describe the MOB method in brief, some previous notation is needed.

Let us denote by  $Y$  the outcome target variable. On the other hand, let  $\mathbf{X}$  be a vector of covariates used to explain the outcome  $Y$  by means of a parametric model  $\mathcal{M}(\mathbf{X}, Y; \theta)$ ; assume that the model is fitted by the optimization of an objective function  $\Psi(\mathbf{X}, Y; \theta)$ ; some standard objective functions are those ones defined by the ordinary least squares (OLS) or the maximum likelihood (ML)

methods. Finally, let us denote by  $\mathbf{Z}$  an input vector containing a set of segmentation variables like customer attributes, product characteristics and some other business inputs.

The goal of the MOB method is the search of non-overlapping groups in the data, defined by the segmentation variables, such that the parametric model  $\mathcal{M}(\mathbf{X}, Y; \theta)$  exhibits differential fits on each group. This goal is accomplished by assessing parameter model stability through fluctuation tests, which are well-established inferential tools for testing parameter stability [20, 21]. The MOB method is implemented by a greedy search algorithm that finds the segmentation variable yielding the highest instability, as assessed by the significance level of the corresponding fluctuation test. At each step, the data set is partitioned into two data subsets by a binary split in the segmentation variable which defines the rule yielding the descendant nodes; the recursive partitioning stops when the highest achievable significance for testing stability is above a specified significance level (default:  $\alpha = 0.05$ ), in which case, the node is declared as terminal. The algorithm can be summarized by the following steps.

---

**MOB recursive partitioning method**

---

Set the outcome variable  $Y$ , the vector of covariates  $\mathbf{X}$  and the vector  $\mathbf{Z}$  of segmentation variables

Set the significance level threshold to assess parameter instability (default  $\alpha = 0.05$ )

**Step 1.** Fit the parametric model  $\mathcal{M}(\mathbf{X}, Y; \theta)$  to the data (parent node)

**Step 2.** Test for parameter stability in the set of segmentation variables

**Step 3.** Find the most significant variable, say  $Z_i$

If its significance is higher than  $\alpha$  then stop and declare the node as terminal

else split in  $Z_i$ , by finding the cutoff point that locally optimizes  $\Psi$ , in order to get descendant nodes

**Step 4.** Go to step 1 and repeat the procedure for each one of the descendant nodes

---

The *partykit* R package [12] provides an easy to use implementation of the MOB method with fancy utilities for setting  $\alpha$  and the minimal size of terminal nodes, and also for the customization of the output.

### 3 The MOB Method for Pricing Analytics

In the context of the pricing problem, we assume that the outcome variable  $Y$  is a binary one taking the value  $Y = 1$  if a customer has accepted a bid and the value  $Y = 0$  if the customer didn't accept it. Now, we consider only one covariate,  $X$ , which is the price variable. Without considering exogenous factors, we can theoretically assume that the probability of acceptance of a bid increases as the price decreases; this is a natural observation that points to the *Logit* model as a reasonable one to estimate the *bid response function* [18]. Hence, in this case the general model  $\mathcal{M}(\mathbf{X}, Y; \theta)$  is given by the following equation:

$$\log \frac{P(Y = 1|X)}{P(Y = 0|X)} = \alpha_0 + \alpha_1 X. \quad (1)$$

The coefficients in (1) are fitted by the ML method. Once the model has been fitted from the data, a function of the purchase probability against the price can be obtained upon inversion of the *Logit* transform.

In this context we also consider a set of partitioning variables collected by a vector  $\mathbf{Z} = (Z_1, Z_2, \dots, Z_k)$  which may contain product characteristics, socio-economic and demographic customer attributes, and any other related business input. When applied to pricing, the goal of the MOB method is the search of a data partition leading to segments that exhibit differential purchase sensitivities, which are described by differential fits of the *bid response function* using the *Logit* Eq. (1). Hence, its application will allow to uncover groups that can be classified in accordance to their differential PS.

The MOB method is appealing and intuitive, as it provides a customized estimate of the *bid response function* easy to interpret in terms of the segmentation variables that came up as splitters in the resulting MOB tree. For each one of the terminal nodes of the tree, the customized *bid response function* can be expressed formally as follows:

$$g(r, \mathbf{z}) = P(Y = 1 | X = r, \mathbf{Z} = \mathbf{z}) \quad (2)$$

where  $r$  and  $\mathbf{z}$  are the price and the observed values of the vector of segmentation variables at a customer/bid level.

Note that for an observed instance of the segmentation variables,  $\mathbf{Z} = \mathbf{z}$ , we obtain a function with respect to the price  $r$ . Thus, in order to maximize the expected revenue for each bid, we state the following optimization problem.

**Statement 1.** *Let us assume that the vector of segmentation variables is such that  $\mathbf{Z} = \mathbf{z}$ . The optimal price allocation can be derived by solving the following optimization problem*

$$\max_r g(r, \mathbf{z})R(r, \mathbf{z}) \quad (3)$$

where  $R(r, \mathbf{z})$  is the revenue for a given bid with price  $r$ .

If we denote by  $r^*$  the optimal price that solves (3) then the maximum expected revenue of the bid is  $g(r^*, \mathbf{z})R(r^*, \mathbf{z})$ . Now, assume that the MOB method results in a data partition with  $H$  terminal nodes, each one of size  $N_k$ , such that  $N_1 + N_2 + \dots + N_H = N$  with  $N$  the total number of cases in the data; then the overall expected revenue of the resulting MOB optimization model can be calculated as follows.

**Statement 2.** *Let us denote by  $g_j$  the bid response function derived by the MOB method at the  $j$ th terminal node  $TN_j : j = 1, 2, \dots, H$ . Then the total expected revenue obtained by optimal price allocation is given by*

$$TotalRevenue = \sum_{j=1}^H \sum_{\mathbf{z}_i \in TN_j} g_j(r_i^*, \mathbf{z}_i)R(r_i^*, \mathbf{z}_i) \quad (4)$$

where  $r_i^*$  and  $\mathbf{z}_i$  are the optimal price and the observed attributes for the  $i$ th bid.

## 4 Business Case Application

This next section gives an application of the MOB method to a real business case. The results of optimal price allocation using MOB are compared with un-optimized actual prices and with the optimal prices obtained on the basis of the *Logit* estimation of the bid response function.

### 4.1 Data Description

An auto lending company collected historical data from loan applications during the period from July 2002 to November 2004. The data set contains 208085 approved applications. The 47210 applications for refinancing were removed from the analysis. In addition, we only considered applications that received approval at least 45 days prior to the investigation end date because of managerial reasons. Hence, we end up with 152965 applications for which the auto lender collected several sources of information measured by the set of variables in Table 1.

**Table 1.** Variables collected during the period of study.

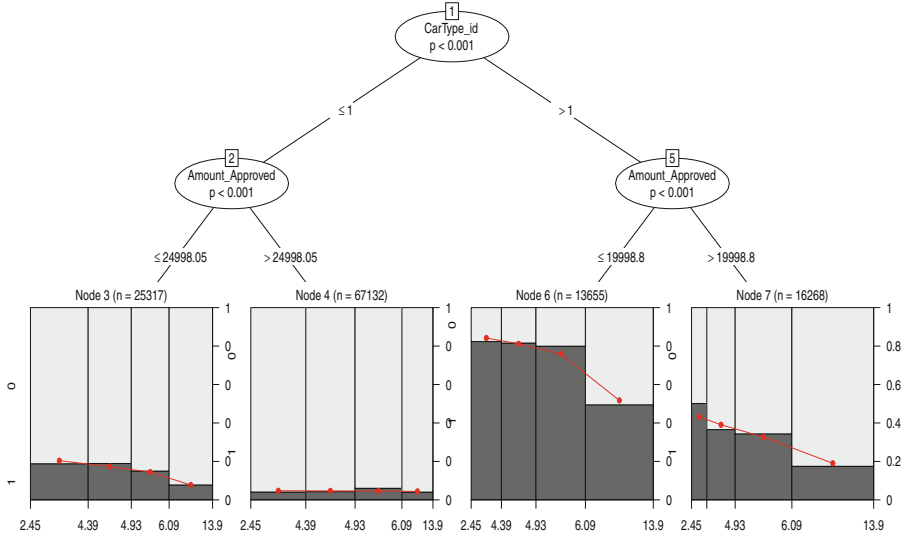
Set of variables for MOB modeling	
Tier	Classification of applicants based on FICO scores
Primary FICO	FICO score quantifying the applicant's risk in the range [594, 854]
Term	Loan term in months
Amount Approved	Amount of the loan in the range [5, 100000]
Competition rate	Interest rate of competitor
Car Type id	Type of car: new (1) or used (2)
term class	Four level segmentation of the Term variable
partnerbin	Segmentation based on partners (1: Direct. 2: Partner A. 3: Other partners)
rate	Interest rate of the application
onemonth	Prime rate
apply	Binary outcome variable with the purchase decision

### 4.2 MOB Modeling

The outcome  $Y$  is the *apply* variable which takes the value 1 if the applicant was funded and the value 0 otherwise. The records indicate that 26323 applications were funded; so the response rate is around 17.2%. In this case the price variable  $X$  is the interest rate (*rate* variable). On the other hand, the vector  $\mathbf{Z}$  for segmentation contains the following variables: *Tier*, *PrimaryFICO*, *Term*, *AmountApproved*, *Competitionrate*, *CarTypeid*, *termclass*, *partnerbin*.

The MOB tree model is fit on a training sample containing 80% of the entire data set. The algorithm is parameterized using the default significance level for node splitting,  $\alpha = 0.05$ , and a minimum node size of 5% the size of the entire training data set. In this case, a segmentation with  $H = 4$  terminal nodes is

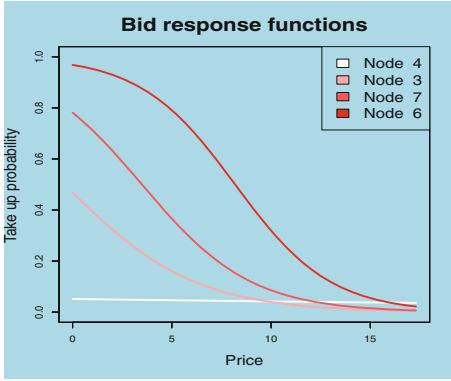
obtained; it is given by the binary rules shown in Fig. 1. The estimation of the parameters in (1) provides the *bid response functions* depicted by the plots in Fig. 2 for each terminal node; the equations of the corresponding *Logit* transforms are shown in the table aside.



**Fig. 1.** Segmentation obtained by the MOB method.

Recall that the coefficient  $\alpha_1$  in Eq. (1) is a non-decreasing function of the odds ratio ( $OR$ ), i.e.  $OR = e^{\alpha_1}$ . Hence, it can be interpreted as a measure for assessing PS, as it quantifies the decay of the likelihood of a positive response to the bid due to a unit price increase, with the more negative coefficients corresponding to the higher sensitivities [5]. Therefore, the resulting segments depicted by the tree of Fig. 1 can be classified as follows (see the table with the *Logit* equations of Fig. 2): The highest PS corresponds to the node 6 group, defined by loan applications for used automobiles with approved amount under \$20000; the nodes 7 and 3 may be considered as moderate to high PS groups, the former corresponding to applications for used automobiles with approved amount above \$20000 and the latter to loan applications for new automobiles whose approved amount is under \$25000; finally, the lowest PS group appears at the node 4 which corresponds to applications for new automobiles whose approved amount is above the \$25000 cutoff. Overall, we can interpret that for both new and used car applications the PS is higher for the lower approved amounts, as given by the cutoffs of about \$25000 and \$20000 respectively.

The results provide useful insights for undertaking managerial decisions: We could suggest a strategy for raising the price at the segments with low sensitivity, as we would expect a slight negative impact in the purchase decision; on the



Node ID	Logit equation
3	$-0.13 - 0.31X$
4	$-2.92 - 0.02X$
6	$3.42 - 0.42X$
7	$1.27 - 0.36X$

**Fig. 2.** Bid response functions in the terminal nodes given by the MOB tree. The table contains the *Logit* equation at each terminal node.

other hand, we could recommend a price reduction for the segments with high sensitivity in order to increase the response rate at the expense of a revenue loss. Section 4.4 addresses the issue of optimal price allocation.

### 4.3 Logistic Regression Modeling

Now we succinctly review the logistic regression method and show its application to customized pricing. In this scenario the method allows to estimate the bid response function by fitting a linear model to the *Logit* of the take up probability against the price variable with the incorporation of a set of covariables [4,5]. Mathematically, the model is formulated by the following equation:

$$\log \frac{P(Y = 1|X, \mathbf{Z})}{P(Y = 0|X, \mathbf{Z})} = \alpha_0 + \alpha X + \alpha_1 Z_1 + \alpha_2 Z_2 + \dots + \alpha_p Z_p. \tag{5}$$

Here,  $X$  is the price variable and  $\mathbf{Z} = (Z_1, \dots, Z_p)$  is the vector of covariables that measure bid characteristics and customer attributes. The coefficients in Eq. (5) are fitted by maximum likelihood. Upon inversion of the *Logit*, we get the take up posterior probability, ensuring values in the interval  $[0, 1]$ .

Recall that in our business case  $X$  is defined by the interest rate. Now, we fit two *Logit* models: the first one uses as covariables the segmentation variables that came up in the MOB tree of Fig. 1; for the second fit, we include the following set of variables: *PrimaryFICO*, *Term*, *AmountApproved*, *CompetitionRate*, *CarTypeid*, *partnerbin* in the vector  $\mathbf{Z}$  of covariables. The corresponding fits of model (5) lead to the *Logit* equations:

$$\begin{aligned} \text{Logit1} = & -0.5256 - 0.3150 \times \text{rate} \\ & -0.0001 \times \text{AmountApproved} + 2.1555 \times \text{CarTypeid} \end{aligned} \tag{6}$$



$$\begin{aligned}
 \text{Logit2} = & 1.5941 - 0.6022 \times \text{rate} - 0.0055 \times \text{PrimaryFICO} + 0.0473 \times \text{Term} \\
 & - 0.0001 \times \text{AmountApproved} + 0.3776 \times \text{CompetitionRate} \\
 & + 2.0908 \times \text{CarTypeid} - 0.2114 \times \text{partnerbin}
 \end{aligned} \tag{7}$$

The next section addresses the price allocation problem for revenue optimization using the MOB modeling approach. The expected revenues resulting from MOB optimal price allocation are compared with the expected revenues obtained by price allocation on the basis of the *Logit* approach.

#### 4.4 Optimization and Revenue Results

So far we have been concerned with the customized estimation of the *bid response function*; recall that the MOB method allowed to classify customers/bids in accordance to their PS. Now, the output given by the MOB tree is used as input to address the optimal price allocation problem and to calculate the expected revenues accordingly.

First of all, we need the revenue function  $R(r, \mathbf{z})$  involved in the optimization problem (3). Since we have at our disposal the term, amount and prime rate of each bid, the revenue can be calculated by

$$R(r, \mathbf{z}_i) = DP \cdot A_i \cdot T_i \cdot \left( \frac{r/12}{1 - (1 + r/12)^{-T_i}} - \frac{pr_i/12}{1 - (1 + pr_i/12)^{-T_i}} \right) \tag{8}$$

The quantities involved in this expression denote the following business inputs:  $DP$  is the probability of default which can be set at the value  $DP = 0.85$  as suggested by [5]. On the other hand,  $A_i$ ,  $T_i$  and  $pr_i$  are the approved amount, term and prime rate for the  $i$ th approved bid.

If we insert (8) in expression (3) using the bid response function of the corresponding terminal node, say  $g_j$ , and solve the optimization problem, we get the optimal rate  $r_i^*$  and the expected revenue  $g_j(r_i^*, \mathbf{z}_i)R(r_i^*, \mathbf{z}_i)$  for the  $i$ th bid as long as the bid belongs to the  $j$ th terminal node. The overall expected revenue can be calculated by computing (4) on an independent test sample; we also have at hand the expected revenues per node, which are given by

$$\text{Rev}_j = \sum_{\mathbf{z}_i \in TN_j} g_j(r_i^*, \mathbf{z}_i)R(r_i^*, \mathbf{z}_i) : j = 1, 2, 3, 4. \tag{9}$$

Note that the terminal nodes,  $TN_j : j = 1, 2, 3, 4$ , correspond to the nodes labeled by Node 3, Node 4, Node 6 and Node 7 in the tree of Fig. 1.

In order to calculate the revenue obtained from logistic regression, we must take into account that in this case there is a single bid response function,  $g(r, \mathbf{z})$ , defined by the fit of model (5); when inserted in (3), the overall expected revenue can be computed by

$$\text{TotalRevenue} = \sum_{i=1}^N g(r_i^*, \mathbf{z}_i)R(r_i^*, \mathbf{z}_i) \tag{10}$$

In our case, the function  $g(r, \mathbf{z})$  will be replaced by any of the *Logit* Eqs. (6) or (7), depending on the fit we aim to use. Hence, we can compare MOB revenue results with the revenues obtained from both *Logit* models.

The results are shown by Table 2 which contains the actual revenues, corresponding to current un-optimized prices, and the expected revenues obtained by optimal price allocation derived from MOB and both *Logit* fits (percentage node revenues appear in parenthesis). The lift columns provide the increase of the expected revenue with respect to the un-optimized revenue, quantified in percentages. We can observe that the expected revenue given by MOB optimal price allocation is higher than the status quo actual revenue with an overall 39.2% revenue increase, the largest lift appearing at node 4; moreover, the other nodes also exhibit increases in revenue with lifts 8.2%, 8.2% and 18.5%. These findings reveal the usefulness of the MOB method for highlighting new business opportunities and insights. We can also note that overall, with the striking exception of node 7, the revenues resulting from MOB optimal price allocation are greater than the revenues obtained from the *Logit* approach.

**Table 2.** Actual un-optimized revenues and the expected revenues given by optimal price allocation from MOB and the *Logit* fits (6) and (7) are shown in the columns *Actual*, *MOB*, *Logit1* and *Logit2* (amounts measured in \$ millions on the test sample). Their lifts are provided by the columns  $lift_{MOB}$ ,  $lift_1$  and  $lift_2$  respectively. All the values are rounded to the first decimal point.

Node ID	<i>Actual</i>	<i>MOB</i>	<i>Logit1</i>	<i>Logit2</i>	$lift_{MOB}$	$lift_1$	$lift_2$
3	255.9 (13.6%)	276.8 (10.6%)	250.7 (13.0%)	365.5 (15.2%)	8.2%	-2.0%	42.8%
4	444.6 (23.7%)	1008.6 (38.6%)	390.2 (20.2%)	601.8 (25.0%)	126.9%	-12.2%	35.4%
6	531.6 (28.3%)	630.0 (24.1%)	571.6 (29.5%)	565.1 (23.4%)	18.5%	7.5%	6.3%
7	644.3 (34.3%)	697.4 (26.7%)	722.8 (37.3%)	877.4 (36.4%)	8.2%	12.2%	36.2%
Total	1876.4	2612.8	1935.3	2409.8	39.2%	3.1%	28.4%

## 5 Summary and Concluding Remarks

In this paper we have addressed the customized price optimization problem. A proposal that combines the data-driven MOB method with optimal price allocation is presented using a two step price allocation strategy: firstly, differential *bid response functions* are derived using the MOB method and a segmentation with differential PS groups is obtained as a result. Secondly, optimal price allocation is carried out, taking the customized *bid response functions* as inputs for calculating the price that maximizes the expected revenue at a customer/bid level. The proposed approach was applied to the business case of an on-line auto lending company on the basis of the historical data of loan applications. A MOB tree is fitted on a training data set and the expected revenue is calculated on a test sample data set. The results show that application of the MOB optimal price allocation method may result in revenue increases with respect to the status quo scenario of un-optimized prices; the comparison with the standard *Logit*

method for price allocation also reveals overall revenue gains. Future research would consist of the validation of the MOB method in other business scenarios so that it can be proposed as a consistent and well-established tool for customized pricing analytics.

**Acknowledgements.** The author acknowledges the Center for Pricing and Revenue Management at Columbia University as the provider of the data. He is also grateful to two anonymous reviewers for their useful comments that helped to improve the paper.

**Funding.** This work has been funded by *UNED - Santander Bank Research and Transfer 2018 Award* with reference 2018V/PREMIO/07 under grant 18CF05.

## References

1. Arevalillo, J.M.: A machine learning approach to assess price sensitivity with application to automobile loan segmentation. *Appl. Soft Comput.* **76**, 390–399 (2019)
2. Batmaz, İ., Danişoğlu, S., Yazıcı, C., Kartal-Koç, E.: A data mining application to deposit pricing: main determinants and prediction models. *Appl. Soft Comput.* **60**, 808–819 (2017)
3. Beck, J.A., Kim, M., Schmidgall, R.S.: The pricing for same-day arrival guests in the hotel industry. *Int. J. Bus. Appl. Soc. Sci.* **4**(1), 1–18 (2018)
4. Bodea, T., Ferguson, M.: Pricing Segmentation and Analytics. Marketing Strategy Collection. Business Expert Press, New York City (2012)
5. Bodea, T., Ferguson, M.: Segmentation, Revenue Management and Pricing Analytics. Routledge (2014). <https://books.google.es/books?id=bu-buAAACAAJ>
6. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees. Wadsworth and Brooks, Monterey (1984)
7. Granados, N., Kauffman, R.J., Lai, H., Lin, H.: À la carte pricing and price elasticity of demand in air travel. *Decis. Support Syst.* **53**(2), 381–394 (2012)
8. Guelman, L., Guillén, M.: A causal inference approach to measure price elasticity in automobile insurance. *Expert Syst. Appl.* **41**(2), 387–396 (2014)
9. Guelman, L., Guillen, M., Pérez-Marín, A.M.: A survey of personalized treatment models for pricing strategies in insurance. *Insur. Math. Econ.* **58**, 68–76 (2014)
10. Guo, X., Ling, L., Yang, C., Li, Z., Liang, L.: Optimal pricing strategy based on market segmentation for service products using online reservation systems: an application to hotel rooms. *Int. J. Hospitality Manag.* **35**, 274–281 (2013)
11. Hothorn, T., Hornik, K., Zeileis, A.: Unbiased recursive partitioning: a conditional inference framework. *J. Comput. Graph. Stat.* **15**(3), 651–674 (2006)
12. Hothorn, T., Zeileis, A.: partykit: a modular toolkit for recursive partytioning in R. *J. Mach. Learn. Res.* **16**, 3905–3909 (2015)
13. Kuyumcu, H.A.: Emerging trends in scientific pricing. *J. Revenue Pricing Manag.* **6**(4), 293–299 (2007)
14. Liaw, A., Wiener, M.: Classification and regression by randomforest. *R News* **2**(3), 18–22 (2002). <http://CRAN.R-project.org/doc/Rnews/>
15. Loh, W.Y.: Fifty years of classification and regression trees. *Int. Stat. Rev.* **82**(3), 329–348 (2014)
16. Morlotti, C., Cattaneo, M., Malighetti, P., Redondi, R.: Multi-dimensional price elasticity for leisure and business destinations in the low-cost air transport market: evidence from easyJet. *Tourism Manag.* **61**, 23–34 (2017)

17. Mumbower, S., Garrow, L.A., Higgins, M.J.: Estimating flight-level price elasticities using online airline data: a first step toward integrating pricing, demand, and revenue optimization. *Transp. Res. Part A Policy Pract.* **66**, 196–212 (2014)
18. Phillips, R.: *Pricing and Revenue Optimization*. Stanford Business Books. Stanford University Press, Palo Alto (2005)
19. Phillips, R.: Optimizing prices for consumer credit. *J. Revenue Pricing Manag.* **12**(4), 360–377 (2013)
20. Zeileis, A., Hornik, K.: Generalized m-fluctuation tests for parameter instability. *Stat. Neerl.* **61**(4), 488–508 (2007)
21. Zeileis, A., Hothorn, T., Hornik, K.: Model-based recursive partitioning. *J. Comput. Graph. Stat.* **17**(2), 492–514 (2008)