ISSN 2281-4299

DIPARTIMENTO DI INGEGNERIA INFORMATICA
AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA
UNIVERSITÀ DI ROMA

Random Forest-driven Heuristic for Margin Optimal Regression Trees

Virginia Marcelli
Laura Palagi
Marta Monaci
Ilaria Ciocci

Technical Report n. 03, 2025

Random Forest-driven Heuristic for Margin Optimal Regression Trees

Virginia Marcelli[1], Laura Palagi[1], Marta Monaci[1,2], and Ilaria Ciocci[1]

[1]Department of Computer, Control and Management Engineering, Sapienza University of Rome, Rome, Italy
[2]Universitas Mercatorum, Università telematica delle Camere di Commercio Italiane, Rome, Italy.

Abstract

This report introduces two heuristic approaches designed to generate high-quality feasible solutions for an Optimal Margin Regression Tree (MARGOT^{REG}) formulated as a Mixed Integer Quadratic Programming (MIQP) problem. MARGOT^{REG} aims to combine the interpretability of decision trees with the strong generalization capabilities of Support Vector Regression (SVR), and it takes inspiration from MARGOT for classification [9]. The proposed framework offers a promising balance between interpretability, predictive accuracy, and model robustness, contributing to the integration of exact optimization methods and machine learning approaches for regression tasks. The first proposed heuristic, the Local SVR Heuristic, employs a recursive top-down strategy that applies weakened SVR models at branch nodes and powerful SVR models at leaf nodes. The second one, the Proximity Heuristic, leverages Random Forest-derived proximity matrix to introduce constraints that enforce the grouping of samples with high proximity values between them. Both heuristics aim to efficiently warm-start the optimization process in order to enhance the performance of an off-the-shelf branch-and-bound solver used to solve the MIQP problem. Empirical results on multiple benchmark datasets demonstrate that both heuristics provide high-quality initial solutions that can improve the performance of the branch-and-bound solver, with the Proximity Heuristic yielding superior reductions in the objective function.

Keywords: Optimal regression trees; Support vector regression; Random Forest; Mixed integer programming.

1 Introduction

Decision trees are among widely used techniques in Machine Learning for both classification and regression tasks thanks to their high level of interpretability. They partition the input feature space into regions through a sequence of hierarchical decisions, where internal (branch) nodes define splitting rules and terminal (leaf) nodes provide predictions. Their popularity stems from their intuitive structure and interpretability, as the logic of predictions can be directly traced through the path from the root to the leaves.
Traditionally, decision tree methods have relied on simple iterative heuristic approaches based on greedy strategies. One of the most known heuristic algorithms for learning univariate trees is CART (Classification and Regression Trees) [3]. The goal is to optimize the trade-off between the training loss in the leaves and the complexity of the tree (i.e., the number of splits). CART is based on a top-down approach for growing the tree and, once the tree is built, it includes a pruning phase to handle its complexity. However, the objective function is not optimized globally, but rather through a specific heuristic approach. While this results in a model that is interpretable, easy to explain, and fast to train, the greedy nature of the algorithm brings the drawback of making myopic decisions, where each split is determined locally without considering the potential impact of future splits in the tree. As a result, this heuristic procedure (as well as other decision trees methods, C4.5 and ID3, proposed by [13, 14] can lead to models which may fail to capture the underlying structure of the dataset, leading to poor generalization performance.
The use of iterative heuristics to build decision trees was not due to the belief that such procedures were inherently better than constructing an optimal binary decision tree. Rather, this choice was driven by the fact that, at that time, the computational power was not enough to solve such a large NP-hard problem [12].
However, the last thirty years have experienced great advances in integer programming coupled with an incredible speedup in the computational power of computer hardware. Thanks to this, commercial mixed-integer solvers such as CPLEX [6] and Gurobi [10] are able to solve linear mixed-integer problems of considerable size and have led to many recent successes in addressing statistical problems using modern integer programming based techniques.
Thus, this progress has made it possible to define global exact optimization approaches to find an Optimal Decision Tree, formulating the problem as a Mixed Integer Programming (MIP) model. Such approaches allow the decision tree to be defined in its entirety through the resolution of a single optimization model, where each branching rule is defined with full knowledge of all the remaining ones.
The first exact formulations for learning optimal decision trees via Mixed Integer Programming were proposed by Bertsimas and Dunn. Initially focused on classification [1], their framework was later extended to regression [2], marking a significant breakthrough in the use of global optimization techniques for interpretable machine learning.
Among the Optimal Classification Tree approaches, recently the Margin Optimal Classification Tree (MARGOT) [9] was proposed which encompasses maximum margin multivariate hyperplanes nested in a binary tree structure. The definition of the hyperplanes, at each branch node t, is obtained by using a linear soft Support Vector Machine (SVM) paradigm. The resulting mathematical programming problem is a structured Mixed In-

teger Quadratic Programming (MIQP). Building on this, we are working on extending MARGOT to regression tasks. In this context, the Margin Optimal Regression Tree (MARGOT^{REG}) model [5] is proposed as part of ongoing research aimed at constructing optimal regression trees by embedding Support Vector Regression (SVR) models at each leaf node. Exploiting the flexibility and robustness of SVR, particularly its ε-insensitive loss function and regularization capabilities, this approach can capture complex patterns in data while controlling overfitting and promoting sparsity in the solution. The key idea, as it was in MARGOT for Optimal Classification Trees, is to exploited the statistical learning properties of SVR in the leaves of the tree. Thus, tree splits at branching nodes use hyperplanes which recursively partition the input space, and solve a regularized SVR problem at leaf nodes.
In Mixed Integer Programming (MIP), it is well known that providing a good quality initial feasible solution can drastically reduce the solution time by allowing the solver to prune large parts of the search space early in the Branch-and-Bound process. Motivated by this, we introduce two novel heuristics designed to efficiently generate feasible warm start solutions for the MARGOT regression model: the Local SVR Heuristic and the Proximity Heuristic. The first method is based on a recursive top-down greedy strategy that solves weakened SVR problems at each node, while the second exploits the proximity structure extracted from a Random Forest model to enforce clustering of similar instances into common leaves.
Both heuristics are tailored to the structure of the MIQP formulation used by MARGOT^{REG} and aim to balance computational efficiency with the quality of the solution. The proposed models were evaluated on 23 datasets obtained from OpenML [16] and the UCI Machine Learning Repository [8]. The empirical results demonstrate that these heuristics are capable of generating good-quality initial solutions that can accelerate the optimization process, often outperforming the default heuristics embedded within Gurobi. Therefore,this work contributes to the growing body of research that bridges exact optimization and machine learning, offering practical tools for building interpretable and high-performing regression models in a wide range of applications. The paper is organised as follows. In section 2 we report some preliminaries on SVR and Random Forest; in section 3, we present the formulation of MARGOT^{REG} used in the experimental setting presented in section 5. In section 4 we present the two heuristics.

2 Preliminaries

2.1 Support Vector Regression

The sparse solution and good generalization properties of SVMs led to their adaptation also to regression problems, where the regression model estimates a continuous-valued multivariate function, instead of predicting an output from a finite set of values. Support Vector Machines (SVMs) have been extended to handle regression tasks through the Support Vector Regression (SVR) framework. This generalization introduces an ε-insensitive region, known as the ε-tube, around the prediction function. Within this tube, deviations between predicted and actual values are considered negligible and do not contribute to the loss function. Consequently, only prediction errors that fall outside the ε-tube are

penalized using slack variables. The objective of SVR is to determine the narrowest possible tube that captures the underlying structure of the data, while maintaining a balance between model complexity and prediction accuracy. The hyperplane which best approximates the continuous function is represented in terms of support vectors, those training samples which lie outside or exactly on the boundary of the $\varepsilon-$tube, and are thus the most influential in affecting its shape.

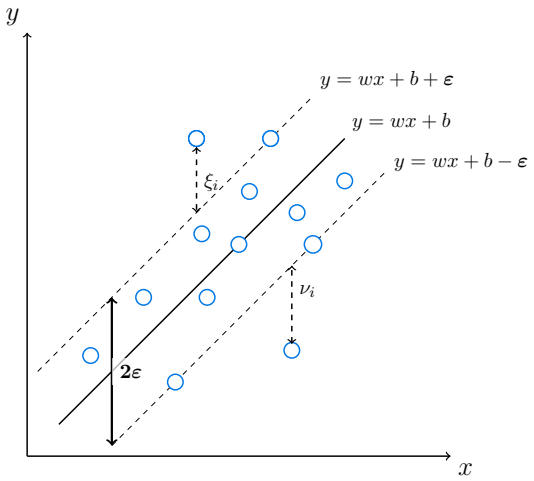

Figure 1: Example of 1-dimension SVR, showing the $\varepsilon-$tube and slack variables ξ_i and ν_i for points that lie outside the $\varepsilon-$tube.

The use of a $\varepsilon-$insensitive region allows the model to be more robust, since it will be less sensitive to noise in the input. The value of ε determines the width of the tube, where a smaller ε implies a stricter tolerance for error, which in turn increases the number of support vectors and reduces the sparsity of the solution. In contrast, increasing ε results in a wider tube and more data points inside the $\varepsilon-$tube, leading to a sparser solution.

Similarly to the classification setting, in regression we distinguish between Linear Hard and Soft SVR. In the hard margin case, assuming the existence of a function that fits all training points within a tolerance ε, the objective is to find a flat function $f(x)$ whose predictions deviate from the actual targets y^i by at most ε for all $i = 1, \dots, P$ [15].
To achieve flatness is sufficient to minimize the ℓ_2 norm of w in the objective function, which also acts as a regularization term. The formulation of the optimization problem for the linear hard case is the following:

$$\begin{aligned} \text{(Hard-SVR)} \quad \min_{w \in \mathbb{R}^n} \quad & \frac{1}{2}\|w\|^2 & \\ \text{s.t.} \quad & y^i - (w^T x^i + b) \leq \varepsilon, & i = 1, \dots, P, \\ & (w^T x^i + b) - y^i \leq \varepsilon, & i = 1, \dots, P. \end{aligned}$$

In the linear soft case instead, as with soft-margin SVM, slack variables are introduced to penalize the instances that lie outside the tube. However, unlike in classification, two

different slack variables are introduced in SVR

- ν_i is used for those samples which lie "above" the ε-tube.
- ξ_i is used for those samples which lie "below" the ε-tube.

These two variables cannot be simultaneously greater than zero, as a point cannot lie both above and below the ε-tube at the same time.
The formulation for the linear Soft SVR is the following:

$$\begin{aligned}
\text{(Soft-SVR)} \quad \min_{w\in\mathbb{R}^n} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{P}(\xi_i+\nu_i) && && (1)\\
\text{s.t.} \quad & y^i - (w^T x^i + b) \le \varepsilon + \nu_i, && i=1,\dots,P && (2)\\
& (w^T x^i + b) - y^i \le \varepsilon + \xi_i, && i=1,\dots,P && (3)\\
& \xi_i \ge 0, \quad \nu_i \ge 0, && i=1,\dots,P. && (4)
\end{aligned}$$

The hyperparameter C balances the bias-variance trade-off: the higher C, the higher the variance of predictions, but the lower the bias. Both $\varepsilon > 0$ and $C > 0$ are user-dependent parameters.

2.2 Random Forests

Random Forests [4] are ensemble models that generate multiple unpruned CART trees using a technique called bagging (bootstrap aggregation), which enhances model stability and reduces variance. Bagging involves training each tree on a bootstrap sample, that is a set of samples drawn uniformly and with replacement from the original training set.
Moreover, at each decision node, the algorithm applies the random subspace method [11], which limits the split candidate features to a randomly selected subset of the available ones. This dual introduction of randomness, through both data sampling and feature selection, leads to diversity among the trees in the ensemble, helping to mitigate overfitting, a common problem in single decision trees.
Once the ensemble is trained, its final prediction is derived by combining the outputs of all individual trees: through averaging in regression tasks and majority voting in classification tasks. Although this approach may slightly increase the bias of the model, reducing the variance significantly generally leads to better overall performance.

3 MARGOT Regression

MARGOT regression is an extension of MARGOT classification to solve regression tasks. Given a dataset $\{(x^i, y^i),\ i \in \mathcal{I}\}$, $x^i \in \mathbb{R}^n,\ y^i \in \mathbb{R}$ we aim to learn a mapping function

$$f : \mathbb{R}^n \to \mathbb{R}.$$

Similarly to the approach used in MARGOT classification [9], the idea in MARGOT regression (MARGOT^{REG}) is that of exploiting the generalization capabilities of the soft

SVR approach to obtain an Optimal Regression Trees. Specifically, at each branch node t, a separating hyperplane partitions the feature space, and at each leaf node ℓ in the tree, an SVR model is instead used to predict the continuous outcome. In the optimization model we have to model the fact that the hyperplane at each node t needs to be trained on just a subset of samples. To this aim, let $\mathcal{I}_t \subseteq \mathcal{I}$ be the index set of samples routed to $t \in \mathcal{T}$. We define the set of branch nodes as $\mathcal{T}_B$ and the set of leaf nodes as $\mathcal{T}_L$. Therefore, starting from the root node, the samples (x^i, y^i), $i \in \mathcal{I}_t$, can be split among the right or left child node of t depending on the following rule:

$$\begin{cases} i \in \mathcal{I}_{R(t)}, & \text{if } w_t^T x^i + b_t \geq 0 \\ i \in \mathcal{I}_{L(t)}, & \text{if } w_t^T x^i + b_t < 0 \end{cases}$$

where sets $\mathcal{I}_{R(t)}$ and $\mathcal{I}_{L(t)}$ are the index sets of samples assigned to the right and left child nodes of t, respectively, thus $\mathcal{I}_{R(t)} \cup \mathcal{I}_{L(t)} = \mathcal{I}_t$ and $\mathcal{I}_{R(t)} \cap \mathcal{I}_{L(t)} = \emptyset$. For each branch node $\in \mathcal{T}_B$, we define as $\mathcal{S}(t)$ the set of nodes of the sub-tree rooted at node t, as $\mathcal{S}_R^\ell(t)$ the set of leaves of the sub-tree rooted under the right branch of node t and as $\mathcal{S}_L^\ell(t)$ the set of leaves of the sub-tree rooted under the left branch of node t. A set of routing constraints is therefore needed, for each sample $i \in \mathcal{I}_t$, in order to impose the correct sign of the hyperplane function in x^i, $h_t(x^i) = w_t^T x^i + b_t$.
It is important to note that the route of samples in the tree is not preassigned, but the definition of subsets $\mathcal{I}_t$ with $t \in \mathcal{T}_B$, is defined by the optimization procedure. Hence, as in the case of the classification problem, the routing constraints at node $t \in \mathcal{T}_B$, must activate only for the subset of samples $\mathcal{I}_t$. In order to model the activation of these constraints, binary variables $z_{i,\ell}$ are introduced, which are used determine the unique path of each sample. The binary variables $z_{i,\ell}$ for each $i \in \mathcal{I}, \ell \in \mathcal{T}_L$, are defined as follows:

$$z_{i,\ell} = \begin{cases} 1 & \text{if } i \text{ is assigned to leaf } \ell \in \mathcal{T}_L \\ 0 & \text{otherwise} \end{cases}$$

It is sufficient to model the assignment of a sample i only to a leaf node to reconstruct the unique path of that sample in the tree. Since each data point x^i will be assigned to exactly one leaf node, we impose the following assignment constraint:

$$\sum_{\ell \in \mathcal{T}_L} z_{i,\ell} = 1, \qquad i \in \mathcal{I}.$$

Since each data point x^i is assigned to a unique leaf node ℓ, it is necessary to enforce the SVR constraints only within the corresponding leaf. To this end, we employ Big-M constraints, which enable the selective activation of the SVR constraints exclusively for the samples assigned to leaf ℓ:

$$y^i - (w_\ell^T x^i + b_\ell) \leq \varepsilon + \xi_{\ell,i} + M_\xi(1 - z_{i,\ell}), \qquad i \in \mathcal{I},\ \ell \in \mathcal{T}_L, \tag{5}$$

$$(w_\ell^T x^i + b_\ell) - y^i \leq \varepsilon + \nu_{\ell,i} + M_\nu(1 - z_{i,\ell}), \qquad i \in \mathcal{I},\ \ell \in \mathcal{T}_L, \tag{6}$$

where $M_\xi > 0$ and $M_\nu > 0$ are a sufficiently large values.
The resulting MARGOT^{REG} formulation, proposed as part of ongoing research, is the following:

$$\min_{\boldsymbol{w},\boldsymbol{b},\boldsymbol{\xi},\boldsymbol{\nu},\ \boldsymbol{z}} \quad \sum_{\ell\in\mathcal{T}_L}\left(\frac{1}{2}\|w_\ell\|^2+C\sum_{i\in\mathcal{I}}(\xi_{\ell,i}+\nu_{\ell,i})\right) \tag{2.1}$$

$$\text{s.t.} \quad y^i-(w_\ell^T x^i+b_\ell)\le\varepsilon+\xi_{\ell,i}+M_\xi(1-z_{i,\ell}), \qquad i\in\mathcal{I},\ \ell\in\mathcal{T}_L, \tag{2.2}$$

$$(w_\ell^T x^i+b_\ell)-y^i\le\varepsilon+\nu_{\ell,i}+M_\nu(1-z_{i,\ell}), \qquad i\in\mathcal{I},\ \ell\in\mathcal{T}_L, \tag{2.3}$$

$$w_t^T x^i+b_t\ge-(n+1)\left(1-\sum_{\ell\in\mathcal{S}_R^\ell(t)} z_{i,\ell}\right), \qquad i\in\mathcal{I},\ t\in\mathcal{T}_B, \tag{2.4}$$

$$w_t^T x^i+b_t+\mu\le(n+1+\mu)\left(1-\sum_{\ell\in S_L^\ell(t)} z_{i,\ell}\right), \quad i\in\mathcal{I},\ t\in\mathcal{T}_B, \tag{2.5}$$

$$\sum_{\ell\in\mathcal{T}_L} z_{i,\ell}=1, \qquad i\in\mathcal{I}, \tag{2.6}$$

$$z_{i,\ell}\in\{0,1\}, \qquad i\in\mathcal{I},\ \ell\in\mathcal{T}_L, \tag{2.7}$$

$$-1\le w_{t,j}\le 1, \qquad t\in\mathcal{T}_B,\ j=1,\dots,n, \tag{2.8}$$

$$-1\le b_t\le 1, \qquad t\in\mathcal{T}_B, \tag{2.9}$$

$$\xi_{\ell,i}\ge 0, \quad \nu_{\ell,i}\ge 0, \qquad \ell\in\mathcal{T}_L,\ i\in\mathcal{I}. \tag{2.10}$$

4 Heuristic for a starting feasible solution

4.1 Local SVR Heuristic

Nowadays Branch&Bound algorithms, thanks to their relevance and efficiency, are at the core of state-of-the-art software for solving Mixed Integer Programming (MIP) problems exactly.
As well known, this method uses bounds to discard subproblems that are guaranteed not to lead to a better solution. For this reason, providing the MIP solver a feasible solution to be used as a good-quality warm start, which serves an upper bound of the optimal one and it can be used to prune nodes of the B&B tree, improves the efficiency of the solution search, yielding eventually to shorter computational times. Thus, developing a good warm start solution is usually addressed in MIP formulations and implemented in off-the-shelf software at the root node of the branching tree. Therefore, for MARGOT regression, we developed a simple greedy heuristic algorithm, denoted as *Local SVR Heuristic*, which exploits the special structure of the MIQP models addressed and can be applied to obtain feasible solutions for MARGOT.

The Local SVR Heuristic is based on a greedy recursive top-down strategy. Starting from the root node, the narrowest tube that best approximates the continuous-valued function is computed for each node using an SVR model embedding.
The general idea is that of using a weakened SVR model for all branch nodes $t \in \mathcal{T}_B$ across all the datasets, by fixing a low value for the regularization term (e.g. $C'_t = 1$) and a relatively high value for the tube width ($\varepsilon'_t = 0.1$). At each leaf node $\ell \in \mathcal{T}_L$, instead, we adopt the same hyperparameters as in the original formulation, setting $C'_\ell = C, \varepsilon'_\ell = \varepsilon$. More in detail, for each node $t \in \mathcal{T}$, given a predefined index set $\mathcal{I}_t \subseteq \mathcal{I}$, the heuristic solves the following problem:

$$
\begin{aligned}
(\text{WS-SVR}_t) \quad \min_{w,b,\xi,\nu} \quad & \frac{1}{2}\|w\|_2^2 + C'_t \sum_{i \in \mathcal{I}_t} (\xi_i + \nu_i) & \\
\text{s.t.} \quad & y^i - (w^T x^i + b) \leq \varepsilon'_t + \xi_i, & \forall i \in \mathcal{I}_t, \\
& (w^T x^i + b) - y^i \leq \varepsilon'_t + \nu_i, & \forall i \in \mathcal{I}_t, \\
& \xi_i \geq 0, & \forall i \in \mathcal{I}_t \\
& \nu_i \geq 0, & \forall i \in \mathcal{I}_t.
\end{aligned}
$$

Given the optimal tuple $(\widehat{w}_t, \widehat{b}_t, \widehat{\xi}_t, \widehat{\nu}_t) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^{|\mathcal{I}|} \times \mathbb{R}^{|\mathcal{I}|}$ obtained at node $t \in \mathcal{T}_B$, the samples are partitioned to the left or right child node in the next level of the tree according to the routing rule defined by the hyperplane $\mathcal{H}_t = \{x \in \mathbb{R}^n : \widehat{w}_t^T x + \widehat{b}_t = 0\}$. Thus, each node t operates on a different subset of samples $\mathcal{I}_t \subseteq \mathcal{I}$, and $\mathcal{I}_{L(t)}$ and $\mathcal{I}_{R(t)}$ represent the index sets of samples assigned to the left and right child nodes of t, respectively. At the end of the procedure, for each $t \in \mathcal{T}$, the solutions $(\widehat{w}_t, \widehat{b}_t, \widehat{\xi}_t, \widehat{\nu}_t) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^{|\mathcal{I}|} \times \mathbb{R}^{|\mathcal{I}|}$, constitute a feasible solution for the original problem. The general scheme is reported in Algorithm 1.

Algorithm 1: Local SVR Heuristic

Data: $\{(x^i, y^i) \in \mathbb{R}^n \times \{-1, 1\}, i \in \mathcal{I}\}$;

Parameters: $\{C'_t > 0,\ \varepsilon'_t > 0,\ t \in \mathcal{T}\}$, D;

Input: Model MARGOT^{REG};

Initialize: $\mathcal{I}_0 = \mathcal{I}$, $\mathcal{I}_t = \emptyset,\ t \in \mathcal{T}_B \setminus \{0\}$, $\widehat{z}_{i,\ell} = 0,\ \ell \in \mathcal{T}_L,\ i \in \mathcal{I}$, $\widehat{\xi}_{t,i} = 0$, $t \in \mathcal{T},\ i \in \mathcal{I}$, $\widehat{\nu}_{t,i} = 0$, $t \in \mathcal{T},\ i \in \mathcal{I}$;

for *level* $k = 0, \ldots, D-1$ **do**
  for *node* $t = 2^k - 1, \ldots, 2^{k+1} - 2$ **do**
    Find $(\widehat{w}_t, \widehat{b}_t, \widehat{\xi}_t, \widehat{\nu}_t)$ optimal solution of WS-SVR$_t$
    Set $\mathcal{I}_{L(t)} = \{i \in \mathcal{I}_t : \widehat{w}_t^T x^i + \widehat{b}_t + \mu < 0\}$ and $\mathcal{I}_{R(t)} = \{i \in \mathcal{I}_t : \widehat{w}_t^T x^i + \widehat{b}_t \geq 0\}$
  end
end
for $\ell \in \mathcal{T}_L$ **do**
  if $i \in \mathcal{I}_\ell$ **then**
    Set $\widehat{z}_{i,\ell} = 1$
  end
end

Output: A feasible solution $(\widehat{w}, \widehat{b}, \widehat{\xi}, \widehat{\nu}, \widehat{z})$ for the input model.

4.2 Proximity Heuristic

We also developed an alternative heuristic, called *Proximity Heuristic*, to provide a feasible solution for the MARGOT regression model. This solution also serves as an upper bound to the optimal one. To explain the general idea behind this warm start, we first recall the concept of proximity matrix, which can be computed based on the structure of a Random Forest model trained on the dataset. The intuition of this measure is that "similar" observations are more likely to end up in the same leaf, compared to dissimilar ones. This intuition is grounded in the idea that when two samples fall in the same leaf of a decision tree, it indicates that those two samples share sufficiently similar characteristics to be routed through the same decision path. In particular, the (i, j) element of the proximity matrix is the fraction of trees of the Random Forest in which samples x^i and x^j fall in the same terminal node (leaf of the tree):

$$\text{Proximity}(i, j) = \frac{\text{Number of trees where } x^i \text{ and } x^j \text{ fall in the same leaf}}{\text{Total number of trees}} \tag{7}$$

Once the proximity matrix was computed from the Random Forest method, we introduced, for each dataset, a set U_π, containing the pairs of indices (i, j) of those samples whose proximity value is higher than or equal to a given *proximity threshold* denoted by π, formally defined as:

$$U_\pi := \{(i,j) \in \mathcal{I} \times \mathcal{I} : \ i < j \ \wedge \ \text{Proximity}(i,j) \geq \pi\}. \tag{8}$$

In other words, U_π includes all pairs of samples that fall into the same leaf in more than a π fraction of the trees in the Random Forest. The proposed heuristic enforces that any pair of samples belonging to the U_π set must also be assigned to the same leaf in the MARGOT model. However, the specific leaf to which these samples are assigned is left to be determined by the optimization process. Thus, the samples are constrained to remain together in some leaf, without prescribing which one.
To enforce this link between samples in the model, following the same idea proposed in [7], we introduced the following *proximity constraints* within the MARGOT^{REG} formulation used in the Proximity Heuristic:

$$z_{i,\ell} = z_{j,\ell} \qquad (i,j) \in U_\pi, \quad \ell \in \mathcal{T}_L. \tag{9}$$

5 Computational results

In this section, we present the numerical results on 23 datasets collected from OpenML and the UCI Machine Learning Repository reported in Table 1.
We construct MARGOT^{REG} model with depths set to 1 and 2 and we solve the MIQP problem using Gurobi [10]. During the branch and bound procedure carried out by Gurobi, once an initial incumbent solution is found, the solver applies heuristic methods to enhance its quality before continuing the exploration of the search tree.
We define f_0 as the objective value of the first incumbent solution, and f_1 as the value of the best incumbent solution after the root node of the branch and bound tree has been processed. We use these two values to compare the quality of the heuristics proposed versus Gurobi one.
In general, providing a high quality warm start solution can help prune parts of the branch and bound tree, thus accelerating the optimization process. If the warm start provided has a better objective value than the initial incumbent of the solver, it is accepted as the first incumbent.
In order to evaluate the quality of the warm start input solution, we run the MIQP in three scenarios:

(i) no initial solution was provided to the solver;

(ii) the solver received the warm start generated by the Local SVR Heuristic;

(iii) the solver was given the warm start computed with the Proximity Heuristic.

In Tables 2 and 3, we report the values of f_0, f_1 of the warm start solution generated by the Local SVR Heuristic and by the Proximity Heuristic against the one used by Gurobi.

For the Proximity Heuristic, we run the random forest implemented in scikit-learn with $N = 100$ estimator trees, the splitting criterion set to the squared error and the random seed fixed to zero to ensure reproducibility. The depths were set to 1 and 2 to obtain the

Dataset	P	n
airfoil	1503	5
auction	2043	7
auto-mpg	392	7
building-costs	372	107
building-sales	372	107
communities	123	122
computer	209	6
concrete-flow	103	7
concrete-mpa	103	7
concrete-slump	103	7
concrete_compr	1030	8
enb-cool	768	8
enb-heat	768	8
housing	506	13
lpga2008	157	6
lpga2009	146	11
lpga2022	158	12
real_estate	414	6
servo	167	12
winequality-red	1599	11
yatch	308	6

Table 1: Information about the datasets considered, where P is the number of samples and n is the number of features.

proximity sets to be used in the Proximity Heuristic, corresponding respectively to the depth values employed in MARGOT^{REG}.
To set the value of the parameter π, we conducted empirical experiments, evaluating the performance of the model on different candidate values: $\pi \in \{0.85,\ 0.90,\ 0.95,\ 0.98,\ 1.00\}$ for one of the problem *auction*.
We note that higher values of π result in a stricter definition of similarity, such that only the pairs of samples that are assigned most frequently to the same leaf in the Random Forest are included in the set U_π. As a consequence, the number of proximity constraints in the model decreases, providing the solver with greater flexibility during optimization. Although this may increase the computational time needed to explore the solution space effectively, it can ultimately lead to higher-quality incumbent solutions. This was reflected by the empirical results. Based on these considerations, we selected $\pi = 0.98$ as it represented a suitable compromise between performance and solution quality.
We set a time limit of 300 seconds for the Local SVR Heuristic (which is never met) and

600 seconds for the Proximity one. For the Proximity Heuristic the time limit is never reached for $d = 1$ and, instead in most cases for $d = 2$. In table 3, we put a * near the name of problems which does NOT reach the time limit for the Proximity Heuristic. For the MARGOT^{REG} model without any warm start, a time limit of 3600 seconds was set, which, however, was never reached, as the reported results refer to both the first incumbent solution and the best incumbent solution obtained after the root node.
The experimental results reported for both the initial objective value f_0 and the best incumbent solution after the root node f_1 clearly demonstrate the effectiveness of the proposed Proximity Heuristic as a warm start strategy within the MARGOT^{REG}, for both depths 1 and 2. This heuristic consistently yields superior performance across all 23 benchmark datasets when compared to both the default scenario (no warm start) and the Local SVR Heuristic.
Most notably, for $D = 1$, the Proximity Heuristic provides significantly lower initial incumbent solution f_0 than its counterparts, indicating that the solver is initialized from a more favorable starting point in the feasible space. The magnitude of improvement is especially evident on larger and more complex datasets such as auction and concrete-compressive, where the warm start produced by the Proximity Heuristic reduces the initial objective by several orders of magnitude compared to the default case. This superior initialization also translates into improved results for f_1, confirming the advantage of beginning the optimization process from a better incumbent solution.
Although the Local SVR Heuristic yields moderate improvements compared to the no warm start scenario, its performance remains consistently inferior to that of the Proximity Heuristic. This may be attributed to the nature of the greedy approach, which lacks the broader perspective captured by the Proximity Heuristic that better exploits structural similarities within the data.
As in the case for $d = 1$, also for $d = 2$ the Proximity Heuristic consistently outperforms both the No Warm Start and Local SVR Heuristic cases across all datasets, for both f_0 and f_1.
Given that none of the tested datasets, characterized by varying feature dimensions and sample sizes, resulted in worse outcomes than the initial incumbent found by Gurobi, the Proximity Heuristic can be considered a robust and generally applicable warm start strategy. Although it required more computational time than the Local SVR Heuristic, the consistent superiority of the Proximity Heuristic can be attributed to its ability to effectively leverage the local structure of the training data by exploiting sample similarity, thus providing warm start points that are potentially closer to the optimal solution.
The best result for each dataset is highlighted in bold in each table.

6 Conclusion

This work presents two heuristic strategies designed to generate high-quality warm start solutions for mixed-integer optimization problems in regression tasks. The proposed heuristics, the Local SVR Heuristic and the Proximity Heuristic, are designed to enhance the efficiency of the branch-and-bound procedure by supplying high-quality initial solutions that improve convergence speed and overall solution quality. The Local SVR Heuristic

	No Warm Start		Local SVR Heuristic		Proximity Heuristic	
Dataset	f_0	f_1	f_0	f_1	f_0	f_1
airfoil	40176.88	40099.86	40176.88	40099.86	**26192.45**	**26192.45**
auction	1591792.28	1591792.28	1481818.94	1481818.94	**794624.38**	**794624.38**
auto-mpg	7506.56	7506.56	7169.58	7169.58	**4513.23**	**4513.23**
building-costs	2.07	1.68	1.90	1.90	**1.44**	**1.44**
building-sales	195.59	195.59	183.88	183.88	**112.08**	**112.08**
communities	392.00	1.54	1.86	1.67	**0.44**	**0.44**
computer	4.91	4.62	4.70	4.62	**3.40**	**3.40**
concrete-flow	328.00	6.23	7.95	6.23	**4.39**	**4.39**
concrete-mpa	328000.00	1607.28	1607.28	1607.28	**1075.85**	**1075.85**
concrete-slump	32800.00	707.90	862.95	707.90	**370.56**	**370.56**
concrete_compr	32960000.00	273309.78	273625.27	273309.78	**84939.38**	**84939.38**
enb-cool	188.14	188.14	187.75	187.75	**169.75**	**169.75**
enb-heat	157929.92	138623.98	157929.92	138623.98	**114833.02**	**114833.02**
housing	16160.00	252.53	250.03	250.03	**192.86**	**192.86**
lpga2008	622.70	556.66	469.58	469.58	**366.71**	**366.71**
lpga2008_log	60100.20	59333.36	60100.20	59333.36	**49443.60**	**49443.60**
lpga2009	82.25	76.04	61.80	61.80	**31.82**	**31.82**
lpga2009_ln	53.58	51.20	53.58	51.20	**41.33**	**41.33**
lpga2022	5040.00	17.00	21.49	21.49	**13.95**	**13.95**
real_estate	15.39	15.39	15.30	15.30	**12.13**	**12.13**
servo	79.99	52.96	66.61	52.96	**13.50**	**13.50**
winequality-red	674.43	674.43	674.43	674.43	**627.35**	**627.35**
yacht	23451.87	22838.40	17596.37	17596.37	**3283.39**	**3283.39**

Table 2: Warm start analysis for MARGOT^{REG} with $D = 1$. Columns report the objective values f_0 and f_1 for each heuristic. For each dataset, the best value of each objective is highlighted in bold.

adopts a recursive top-down approach using weakened SVR models at branch nodes and powerful SVR at leaf nodes, while the Proximity Heuristic exploits the proximity structure extracted from a Random Forest model to enforce clustering of similar instances into common leaves. These heuristics have been developed as part of ongoing research on MARGOT Regression, a framework that formulates optimal regression trees as a Mixed Integer Quadratic Programming problem, combining the interpretability of tree-based models with the predictive power of Support Vector Regression.

	No Warm Start		Local SVR Heuristic		Proximity Heuristic	
Dataset	f_0	f_1	f_0	f_1	f_0	f_1
airfoil*	5119.85	5119.85	5119.85	5119.85	**2778.59**	**2778.59**
auction*	13696.96	13696.96	12828.21	12828.21	**1920.53**	**1920.53**
auto-mpg	178.43	178.29	169.26	169.26	**110.21**	**110.21**
building-costs*	23760.00	12.39	13.09	11.18	**4.05**	**3.99**
building-sales	237600.00	59.03	73.19	57.96	**0.97**	**0.90**
communities	7840.00	4.92	9.67	4.92	**0.09**	**0.08**
computer	36.45	24.44	33.23	23.16	**10.47**	**10.47**
concrete-flow	65563.31	60851.07	65563.31	60851.07	**1520.98**	**1520.98**
concrete-mpa	3128.44	3009.91	3128.44	3009.91	**680.34**	**680.34**
concrete-slump	862.95	809.60	862.95	809.60	**53.97**	**45.27**
concrete_comp	82469.55	82330.85	82469.55	82330.85	**48776.39**	**48776.39**
enb-cool*	919.77	919.77	919.77	840.70	**320.92**	**320.92**
enb-heat*	27.38	27.29	27.18	27.18	**11.10**	**11.10**
housing	143.91	107.61	130.89	107.61	**58.11**	**58.11**
lpga2008*	622.70	622.70	423.72	423.72	**128.82**	**128.82**
lpga2008_log	5.55	5.44	5.55	5.44	**4.23**	**3.92**
lpga2009	9280000.00	21690.20	29021.42	21690.20	**3.82**	**3.82**
lpga2009_ln	928.00	5.28	5.52	5.28	**3.30**	**3.30**
lpga2022	29.99	28.94	26.50	26.50	**0.20**	**0.20**
real_estate	18039.57	17963.33	17939.88	17939.88	**12869.21**	**12616.66**
servo*	79.99	79.47	50.83	50.83	**2.40**	**2.40**
winequality-red	102320000.00	670955.85	674206.48	670901.44	**664747.76**	**664747.76**
yacht*	90290.88	90290.88	36698.58	36698.58	**0.01**	**0.01**

Table 3: Warm start analysis for MARGOT with $D = 2$. Columns report the objective values f_0 and f_1 for each heuristic. For each dataset, the best value of each objective is highlighted in bold. The * near the name of problems means that the Proximity Heuristic did not reach the time limit.

Empirical results show that both heuristics provide good-quality initial solutions that effectively guide the solver during the B&B procedure. In particular, the Proximity Heuristic, which is based on sample similarity analysis via Random Forests, consistently outperforms the Local SVR Heuristic. It proves to be especially effective in reducing the incumbent solution provided to the solver, thereby guiding it more effectively during the optimization process.

It is worth noting that the Proximity Heuristic is not limited to the specific context of MARGOT^{REG}. Its data-driven formulation based on proximity structures makes it a general and adaptable warm start strategy that can be applied, with minimal modifications, to other optimal decision tree methods formulated as mixed-integer optimization problems. This generality highlights its potential as a broadly applicable heuristic to improve solver performance in a variety of structured regression tasks and suggests promising directions for further research on heuristic initialization in exact tree-based models.
Overall, the proposed approach demonstrates a promising strategy to generate high-quality warm start solutions that significantly enhance the efficiency of mixed-integer optimization solvers within interpretable regression frameworks. This work contributes to the growing field that bridges machine learning and exact optimization by developing heuristics that improve solver performance while maintaining model interpretability and robustness, paving the way for more practical and scalable deployment of optimization-driven machine learning models in real-world applications.

References

[1] Dimitris Bertsimas and Jack Dunn. Optimal classification trees. *Machine Learning*, 106(7):1039–1082, July 2017.

[2] Dimitris Bertsimas and Jack Dunn. *Machine Learning Under a Modern Optimization Lens*. Dynamic Ideas LLC, Belmont, MA, 2019.

[3] L. Breiman, J. Friedman, C. J. Stone, and R. Olshen. *Classification and Regression Trees*. Chapman and Hall/CRC, 1984.

[4] Leo Breiman. Random forests. *Machine Learning*, 45:5–32, 10 2001.

[5] Ilaria Ciocci, Marta Monaci, and Laura Palagi. Margin optimal regression trees. In *EUROPT 2025 - Continuous Optimization Working Group of EURO - Book of Abstracts*. Southampton, UK, 2025. Available online at `https://europt2025.org/`.

[6] IBM ILOG Cplex. V12. 1: User's manual for cplex. *International Business Machines Corporation*, 46(53):157, 2009.

[7] Giulia Di Teodoro, Marta Monaci, and Laura Palagi. Unboxing tree ensembles for interpretability: A hierarchical visualization tool and a multivariate optimal re-built tree. *EURO Journal on Computational Optimization*, 12:100084, 2024.

[8] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[9] Federico D'Onofrio, Giorgio Grani, Marta Monaci, and Laura Palagi. Margin optimal classification trees. *Computers & Operations Research*, 161:106441, 2024.

[10] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2023.

[11] Tin Kam Ho and Eugene M Kleinberg. Building projectable classifiers of arbitrary complexity. In *Proceedings of 13th International Conference on Pattern Recognition*, volume 2, pages 880–885. IEEE, 1996.

[12] Laurent Hyafil and Ronald L. Rivest. Constructing optimal binary decision trees is np-complete. *Information Processing Letters*, 5(1):15–17, 1976.

[13] J.R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[14] J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[15] A. J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14:199–222, 2004.

[16] Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. Openml: networked science in machine learning. *SIGKDD Explorations*, 15(2):49–60, 2013.