Decision Tree Model hyper-parameters Tuning Using Fractional Factorial Design and Response Surface Modeling Methodology

Gabriel Marvin[1]

[1]Department of Fluid Mechanics and Computational Engineering, University of Rijeka, Faculty of Engineering, Vukovarska 58, 51000, Croatia

Abstract

Design of experiments (DOE) is a branch of applied statistics that is used for planning, executing, analyzing and interpreting experiments. DOE gives a framework that enables the evaluation of the effects of each input factor over a target variable. It is widely used and very common in research practices but there is not much available literature in which this methods are used to understand the interactions between hyper-parameters when tuning machine learning models. This paper presents a methodology to study the interactions between hyper-parameters of a decision tree regressor. The goal is to analyze the main effects and interactions between selected hyper-parameters with factorial designs. This knowledge is usedto tune the decision tree by modeling the response surface between the experiments and the target variables. The benefits of this approach includes fewer training runs when comparing to common practices such as grid searching.

Keywords: Design of experiments, response surface modeling, hyper-parameter tuning, decision trees

1. Introduction

Hyper-parameter selection in machine learning is a crucial and non negotiable step when optimizing the performance of a model. It is generally viewed as an expensive and time consuming process Subaşı (2024). The general approach that is mostly used to tune a model consists of understanding the models hyper-parameters and use them in conjunction with domain knowledge to iteratively search for the best combination that maximizes the performance. The process consists of several iterations of grid searching procedures that increase in granularity around a local minima. This approach, often results with

a highly accurate model but does not give the researcher a knowledge about how the hyper-parameters interact with each other with respect to the performance of the model.

The authors of Probst et al. (2019) address the challenge of hyper-parameter tuning in supervised machine learning algorithms by formalizing the problem from a statistical point of view. This paper conducts an extensive benchmarking study over 38 openML Vanschoren et al. (2013) datasets to define data-based hyper-parameter default values for 6 common machine learning algorithms. The authors also propose a measure to quantify tunability and offer guidance on identifying the most crucial hyper-parameters for each algorithm. This research aims to streamline the hyper-parameter tuning process, making it more efficient and effective for machine learning practitioners.

Yang & Zhang (2021) contributes to the field of automated machine learning optimization by reformulating the problem of hyper-parameter optimization as a computer experiment, addressing the challenge of optimizing hyper-parameters with unknown response surfaces. The paper describes a novel strategy called Sequential Uniform Design (SeqUD). The proposed methodology explores the hyper-parameters space using evenly distributed design points generated in batches. This allows for parallel processing of the batches and sequential generation of new design points. The authors conduct extensive experiments on different hyper-parameter optimization tasks and demonstrate that SeqUD outperforms many benchmark methods.

The paper by Rosa et al. (2020) applies DOE to find the best combination of hyper-parameters that maximize the accuracy of an artificial neural network. The paper follows the methodology from a highly cited paper Lujan-Moreno et al. (2018) but applies it on a more complex model that is considered a black box. The interactions between the hyper-parameters ar much harder to interpret in this case, as opposed to the high interpretability of a random forest model described in Lujan-Moreno et al. (2018). The authors try to help researchers not familiar with the field of machine learning gain a deeper understanding of the hyper-parameters without expert knowledge about the models.

The work by Lujan-Moreno et al. (2018) highlights the current challenges and limitations present when tuning model hyper-parameters. It mentions often disregarded issues like infeasibility of grid searching and not optimal results in commonly used approaches such as changing parameters while keeping one as constant. The paper proposes a series of fractional factorial designs to screen out most significant hyper-parameters and proposes a response surface methodology to fine tune the model with the most significant parameters. This is all done by using a random forest classifier as a case study. This approach offers a more structured and efficient method for hyper-parameter tuning, potentially improving both the performance and interpretability of machine learning models. It addresses the limitations of current popular methods and provides a statistical framework for optimization.

In this paper, a design of experiments (DOE) methodology is proposed to screen out the most significant hyper-parameters of a simple machine learning model. Then as the

next step, to fine tune the model, a response surface modeling (RSM) technique is used to approximate the models performance with different values of hyper-parameters. A fractional factorial design is used to get an idea of the most significant factors. This helps to reduce the number of runs in the full factorial design that follows. In a realistic setting when a model has to be trained on a great amount of data, understanding what the most significant parameters are can reduce the error of the model considerably in a smaller time span. The full factorial design is done to break all the confounding effects of the fractional factorial and get a clear understanding of the existing interaction effects. As a second phase, two rounds of a RSM designs are used to approximate the interactions between the hyper-parameters in the proximity of a centerpoint. At the end, a fine tuned model is presented and compared to an optimized model with classical grid searching.

The paper is structured as follows: Section 2 compiles all the background knowledge that is needed to understand the applied methods and experimental designs. In Section 3 a discussion about the used dataset is conducted. Followed by a description of the model evaluation method and metrics. In Subsection 3.3 a brief explanation of the used models hyper-parameters is conducted. The results of the different designs are described in Section 4. At the end, a discussion of the results is given in Section 5.

The methodology applied in this paper is similar to Lujan-Moreno et al. (2018), but the experiments are conducted on a simpler model. The results show that by using this methodology on this specific dataset a close to optimal model can be achieved. The purpose of this paper is to give the reader a concise and easy to reproduce pipeline for understanding the effects of the hyper-parameters of a model.

2. Background

2.1 Design of Experiments

Design of experiments (DOE) is described as planning a detailed experimental strategy in advance of doing the experiment. Well defined experimental design maximize the amount of information obtained for a given amount of experimental effort Heckert et al. (2002). In an experiment the researcher deliberately changes one or more factors to observe the behavior of a target variable and yield valid conclusions.

2.1.1 Analysis of variance ($ANOVA$)

Analysis of variance ($ANOVA$) is a statistical method used to compare means across multiple groups by analyzing the variance. It determines whether observed differences between group means are due to randomness or reflect real distinctions. The underlying principle of $ANOVA$ is based on the law of total variance, which states that the total variance in a dataset can be broken down into components attributable to different sources. In the context of DOE, $ANOVA$ is used to decide what factor is most significant by

comparing their p-values. A p-value less than the significance level of 0.05 is considered statistically significant. Later in this paper, several $ANOVA$ table will be presented. Each factor and interaction between factors will be described using 3 metrics: Sum of squared errors (SSE), the F-value F which is the ration of between-group and within-group variance, and the p-value denoted as $PR(>F)$ which indicates the probability of the data occurs under the null hypothesis. Values of $PR(>F)$ less than 0.05 are considered valid candidates that reject the null hypothesis.

2.1.2 Full factorial design

A full factorial design (FFD) is a common experimental design where the researcher decides on two settings for each experimental factor. This settings or levels are called 'high' and 'low' and are often denoted as '+1' and '-1' respectively. A FFD is conducted by running all the possible factor combination for a fixed number of repetitions and analyzing the effects on the target variable. This type of experimental design is suitable for a small number of factors since the number of runs increases exponentially 2^k, where k is the total number of factors. After all runs are completed the main and interaction effects can be estimated.

The interaction between two factors can be visualized with the use of line plots, in this paper also referred as interaction plots. For example, The left image in Figure 1 does not show an interaction between factors $X1$ and $X2$ because the response on factor $X2$ is not affected by the levels of factor $X1$. On the other hand, the right image of Figure 1 shows a clear interaction between the two factors. The response clearly drops when moving from low to high setting of factor $X1$ but keeping factor $X2$ on a high level, and increases when keeping $X2$ on a low level.

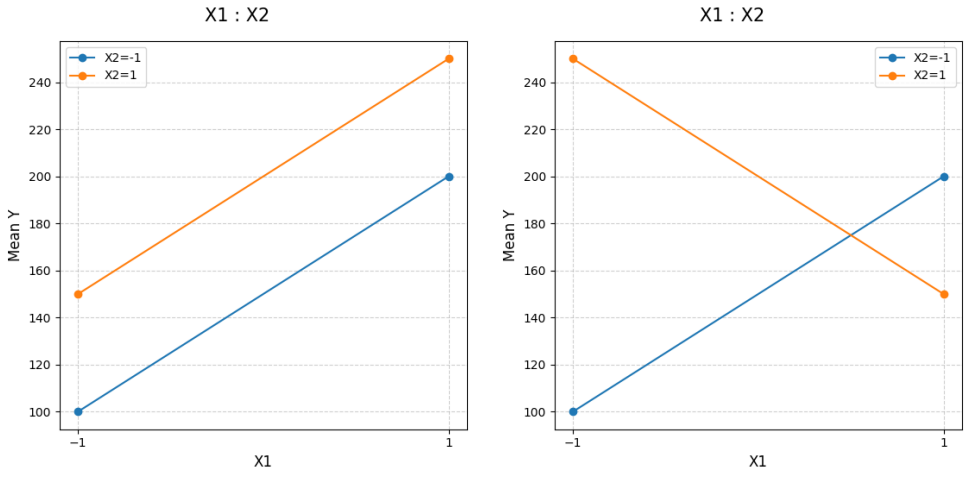


Figure 1: *Left*: experiment with no interaction, *Right*: experiment with interaction

For a full factorial design with 2 levels and 3 factors there are 3 main effects, three 2 factor interactions and, one 3 factor interaction. The complete model is shown in Equation 1.

$$Y = a_1 * X1 + a_2 * X2 + a_3 * X3 + a_{12} * X1X2 + a_{13} * X1X3 + a_{23} * X2X3 + a_{123} * X1X2X3 \quad (1)$$

Using $ANOVA$ the significance of each main and interaction effect is analyzed. This allows to reject the factors with the highest p value.

2.1.3 Fractional factorial design

As it was mentioned in the previous subsection, a FFD can become unfeasible if the number of factors becomes too large. For example, if the goal is to analyze the effects of 6 factors with two levels each on a target variable, the number of total runs that need to be performed is $2^6 = 64$. This means that the researcher needs to set up the equipment and run the experiments with different settings 64 times. To properly run a full factorial the researcher also needs to add centerpoints and conduct several repetitions for each run. This of course multiplies with the amount of time for completing one experiment. In the majority of cases this amount of total runs in unacceptable. One popular design that can be used to tackle this issue is to use fractional factorial designs (FRFD). As the name suggests the idea is to use only a fraction of the runs specified by the FFD. Using FRFD comes with a tradeoff between efficiency and accuracy. This tradeoff is a result of using aliasing or confounding of main effects with higher order interactions. In most cases interactions between 3 or more factors is considered negligible and hard to interpret. For example, when constructing a fractional factorial with 3 factors. The researcher might first construct a 2^2 FFD and use the interaction between factors $X1$ and $X2$ as an alias for the main effect of factor $X3$. Table 1 shows the setup for a 2^2 FFD and Table 2 shows the setup for a 2^{3-1} FRFD.

Table 1: Setup of a 2^2 full factorial design

Run number	X1	X2	X1X2
1	-1	-1	+1
2	+1	-1	-1
3	-1	+1	-1
4	+1	+1	+1

Table 2: Setup of a 2^{3-1} fractional factorial design

Run number	X1	X2	X3
1	-1	-1	+1
2	+1	-1	-1
3	-1	+1	-1
4	+1	+1	+1

Now, if the factor $X3$ turns out to be significant there is no way to know if the significance comes from the interaction between $X1$ and $X2$ or just from the factor $X3$. If a confounded main effect turns out to be significant, a useful property of FRFD called fold over can be used to deconfound main effects. Fold over is achieved by switching certain signs of the FRFD to isolate effects of particular interest Lujan-Moreno et al. (2018).

2.1.4 Box-Behnken design

A RSM experiment is design to allow to estimate interaction and quadratic effects, thus giving an approximation of the local response surface under investigation Heckert et al. (2002). These types of designs are commonly used to find optimal process settings, or in the context of this paper, find an optimal model. A complete description of the process behavior might require not just main and interaction effects but also quadratic (2) or even cubic terms.

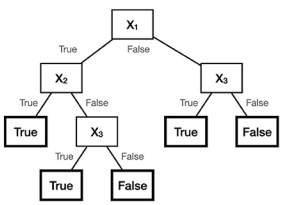
$$\begin{aligned} y = a_0 + a_1 X1 + a_2 X2 + a_3 X3 \\ + a_{12} X1X2 + a_{13} X1X3 + a_{23} X2X3 \\ + a_{11} X1^2 + a_2 X2^2 + a_3 X3^2 \end{aligned} \tag{2}$$

The Box-Behnken design (BBD) is a quadratic RSM design that requires 3 levels at each factor and results in less experimental runs in comparison with other RSM designs such as the central composite design.

2.2 Decision trees

A decision tree (DT) represents a procedure for computing the outcome of a function $f(x)$. The procedure consist of repeatedly performing tests on the input x, where the outcome of each test determines the next test, until $f(x)$ is known with certainty. Figure 2 shows a function in tabular format and two different DT's that represent it Blockeel et al. (2023). In other words, DT's are flow like structures in which each node represents a test on the inputs of that node. The node splits into child nodes called leafs based on the possible outcomes of the parent node.

X_1	X_2	X_3	Y
True	True	True	True
False	True	True	True
True	False	True	True
False	False	True	True
True	True	False	True
False	True	False	False
True	False	False	False
False	False	False	False



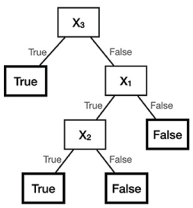


Figure 2: The Boolean function $Y = X1 \wedge X2 \vee X3$, and two decision trees representing it. Figure taken from Blockeel et al. (2023)

DT's that predict nominal or numerical variables are respectively called classification trees and regression trees. The Scikit-learn package Pedregosa et al. (2011) offers the user to choose between a DecisionTreeClassifier and DecisionTreeRegressor (DTR) based on the usecase. An important property of DT's is their interpretability. The outputs of the new samples can easily be traced back to their respective inputs by following the tree structure.

2.3 Gradient descent

Gradient descent (GD) encapsulates a family of optimization algorithms most commonly used for training neural networks in their back-propagation phase. The general purpose of this algorithms is to minimize an objective function $J(\theta)$ by updating the parameters $\theta \in \mathbb{R}^{\times}$ in the opposite direction of the gradient $\nabla J(\theta)$ of the objective function. This algorithm needs a parameter called learning rate (η) that is used to determine the length of the step at each iteration. Eventually the algorithm converges to a minimal value of the target variable. The most common GD algorithm is called vanilla GD a.k.a. batch GD where the parameters θ are evaluated on the entire training dataset before making only one step in the opposite direction of the gradient. Equation 3 describes the update for one step. A pseudo-code of GD is depicted in Algorithm 1.

$$\theta = \theta - \eta * \nabla J(\theta) \tag{3}$$

Algorithm 1 GD algorithm

Require: Initial point x_0, learning rate η, tolerance ϵ, max iterations T
Ensure: Optimized point x

```
x ⇐ x_0
t ⇐ 0
while t < T and |∇f(x)| > ε do
    Δx ⇐ −∇f(x)
    x ⇐ x + αΔx
    t ⇐ t + 1
end while
```

3. Dataset and initial experimental setup

3.1 Dataset

The dataset used for this paper is the *Concrete Compressive Strength* from the UCI ML repository I (1998). The dataset consists of 1030 and 9 attributes, which allows for quick training and experimentation. The concrete compressive strength is a highly nonlinear function of age and ingredients I (1998), this makes it a suitable regression problem for training a DTR, which is known to work well with nonlinear data. All the attributes

consist of real values and no missing data. The target is the compressive strength of concrete in MPa obtained by mixing different quantities of ingredients. The attributes include cement, blast furnace slag, fly ash, water, superplasticizer, coarse aggregate, fine aggregate and age.

3.2 Model evaluation

When conducting experiments a suitable target variable must be chosen. In this case the target variable to study the effects is MSE. Choosing a suitable target variable depends on the goals of the experiments. In the context of this paper, MSE was chosen because of its interpretability and simplicity. Another metric such as R^2 is used later in conjunction with MSE to measure the accuracy of the fine tuned model and comparisons with other baseline models. The final model is compared with a highly underfitted model and a model optimized using an extensive grid search procedure.

The model evaluation with a certain combination of hyper-parameters is conducted using a 3 fold cross validation procedure. This is done to bypass the stochastic nature of decision trees and get a precise estimate of the error. More formally, in the context of design of experiments, repetition of experiments with the same combination of factor levels is desirable Heckert et al. (2002). By using 3 fold cross validation this condition is satisfied.

3.3 Hyper-parameters selection

The chosen hyper-parameters used as factors are the following: max_depth, $max\_features$, $min\_samples\_split$, $min\_samples\_leaf$, $min\_weight\_fraction\_leaf$, max_leaf_nodes and ccp_alpha. The DTR implemented in the package scikit-learn Pedregosa et al. (2011) offers one more hyper-parameter named $min\_impurity\_decrease$, but due to the already great number of factors to be studied this factor was not included. For each factor that is being studied, a low and high setting was defined. A general guideline for the selection of factor settings can be cited from the first few sentences of chapter 5.3.2. in the NIST Engineering Statistics Handbook Heckert et al. (2002) which states that when selecting factor settings the researcher should be bold but not foolish.

This statement can be interpreted in the context of tuning machine learning models as a constraint to avoid hyper-parameter values that can cause the model to overfit or underfit to the data. For the parameter max_depth the low and high settings are chosen to be 20 and 30 respectively. Too low values of max_depth results in a shallow tree prone to underfitting and a very large value allows the tree to fit to the noise in the data resulting in a overfitted model. $min\_samples\_leaf$, $min\_weight\_fraction\_leaf$ are used to improve generalization of the model by requiring a minimum amount of data at each leaf node. $min\_samples\_leaf$ low and high setting are integer values 10 and 100 respectively and for $min\_weight\_fraction\_leaf$ the low and high settings are set to

be 0.1 and 0.4. $min\_samples\_split$ is used to control the minimum number of samples before making a split into a new node. The low and high settings for this factor are set to be 10 and 100. The hyper-parameter $max\_features$ controls the amount of features to consider when looking for the best split at each node. For the low setting the number of features to consider was set to 2 and for the high setting to 6. The number of maximum leaf nodes created before purging is set with the hyper-parameter max_leaf_nodes. Its low and high settings are set to be 10 and 100. ccp_alpha is a complexity parameter used for minimal cost complexity pruning, which is an algorithm used for pruning a tree to avoid overfitting. Low and high settings for ccp_alpha are set to 0.1 and 0.2.

The used dataset is considered small, with only 1030 rows and 9 features. This makes the experiments and model training very fast to compute, to the point that a full factorial design with centerpoints can be run on a personal computer in a short amount of time. However the purpose of this paper is to showcase the effectiveness of conducting well designed experiments to achieve an accurate model in a realistic scenario with scarce resources.

Table 3: Low and high settings for the initial screening design

factor	low	high
max_depth	20	30
$min\_samples\_split$	10	100
$min\_samples\_leaf$	10	100
$min\_weight\_fraction\_leaf$	0.1	0.4
$max\_features$	2	6
max_leaf_nodes	10	100
ccp_alpha	0.1	0.2

4. Results

4.1 Screening of effects with fractional factorial design

In order to extract the main factors that contribute the most in the minimization of the error, while also keeping the number of runs in an acceptable range, a 2^{7-2} fractional factorial experimental design is conducted. This design results in a total of 32 runs, which in comparison to a full factorial design is a substantial decrease in combinations to evaluate. The proposed design consists of 5 independent factors and 2 factors that are defined by interactions between 2 independent factors. The generators used to construct the confounding pattern are $F = AB$ and $G = CDE$, which results in a design of resolution III. Resolution III was chosen to have a less complex confounding pattern with the assumption that there will be minimal interaction between the factors. In Table 6 indicates the aliases of the factors. Table 7 shows the confounding pattern for up to two factor interactions. The $ANOVA$ results of this design are shown in Table 8, where it

is clearly visible that some significant 2 factor interactions exist. Unfortunately due to the confounding effect of the design it is not possible to distinguish one interaction effect from its confounded counterpart, as shown in Table 7. For example, the significance of the interaction $min\_samples\_leaf : min\_weight\_fraction\_leaf$ cannot be distinguished from the interaction $max\_features : ccp\_alpha$.

To resolve this issue, a 2^{7-1} with resolution VII is conducted. The new design only confounds the last factor with the interaction of all the other factors. This still introduces confounding into the design but the improvement is that the main effects are equivalent to interactions between 6 factors, and 2 factor interactions are confounded with 5 factor interactions, which is highly unlikely to have substantial significance. The significant factors are shown in Table 9. At this point, the most significant factors are $min\_samples\_leaf$, $min\_weight\_fraction\_leaf$, $max\_features$ and all of their interactions. The number of factors is successfully reduced to a number suitable to perform a full factorial design, to avoid any confounding effects.

4.2 Full factorial design with distilled factors

Now that only significant factors are left, a full factorial design is conducted. To check for curvature, between each factor settings a centerpoint value is added. This addition of centerpoints is equivalent to performing a full factorial design with 3 factors and 3 levels (3^3). Table 4 shows the significant factors and their respective levels, including the centerpoints. This design results in a total of 27 runs.

To check for significant main effects and interactions another $ANOVA$ is conducted. From the results in Table 5 the interaction between $min\_samples\_leaf$ and $max\_features$ can be considered not significant. This is also visible by looking at the plots of the main effects and the factor interaction plots. Figure 3 shows the interaction between the factors $min\_samples\_leaf$ and $max\_features$. It is clearly visible that the lines at each level are parallel to each other indicating no interaction between the factors at any level.

On the other hand the interactions $min\_samples\_leaf : min\_weight\_fraction\_leaf$ and $min\_weight\_fraction\_leaf : max\_features$ show some interesting behavior. By inspecting the left image of Figure 4, an interaction effect can be seen when the factor $min\_weight\_fraction\_leaf$ is at its centerpoint and high level while $min\_samples\_leaf$ is at its high level. On the other hand, the right image shows an interaction between $min\_weight\_fraction\_leaf$ and $max\_features$ where for all levels of the factor $max\_features$ the MSE increases at the high setting for $min\_weight\_fraction\_leaf$. All the independent factors are still significant, and their effects are shown in Figure 5. The errorbars in the plots of Figure 5 depict the minimum and maximum value of the MSE at that level, while the center line is the mean. This is done to better show the variability of the results at each specific level.

Examination of the residuals obtained from the model is a key part of deciding if the $ANOVA$ is providing trustworthy results. The model residuals need to be (roughly)

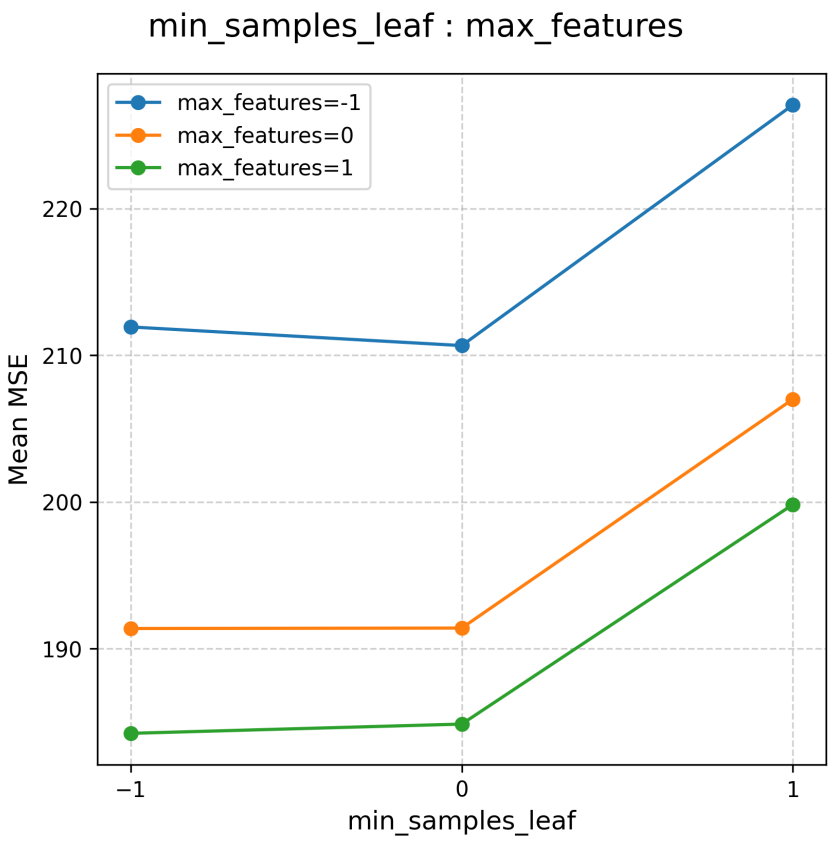


Figure 3: Interaction $min\_samples\_leaf : max\_features$

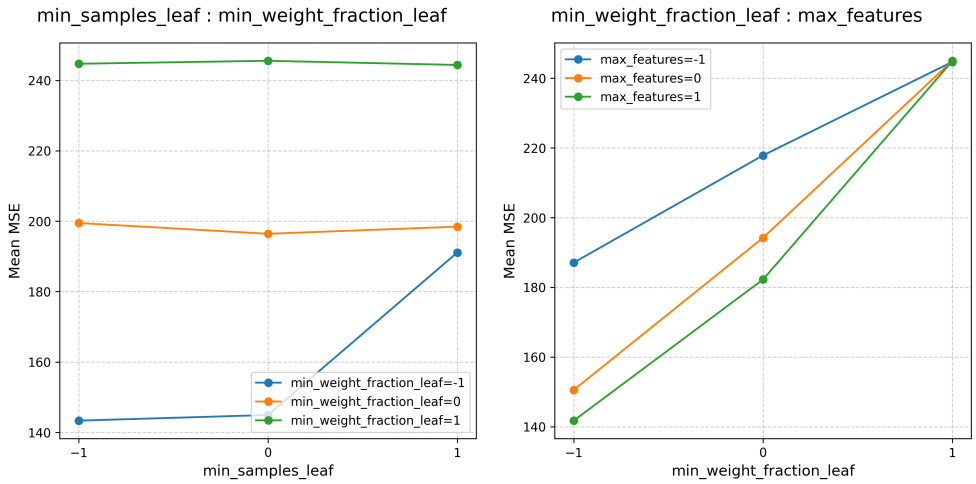


Figure 4: Interactions $min\_samples\_leaf : min\_weight\_fraction\_leaf$ and $min\_weight\_fraction\_leaf : max\_features$

normal and (approximately) independently distributed with the mean of 0 and some constant variance Heckert et al. (2002). From the Q-Q plot in the Figure 6 a strong case for the normality of the residuals can be made. The residuals mostly fall around the reference line that is fit through the quartiles. Furthermore Figure 7 drives the point forward by confirming the constant variance of the residuals. Figure 8 and Figure 9 show the independence of the residuals and their distribution around the mean of 0.

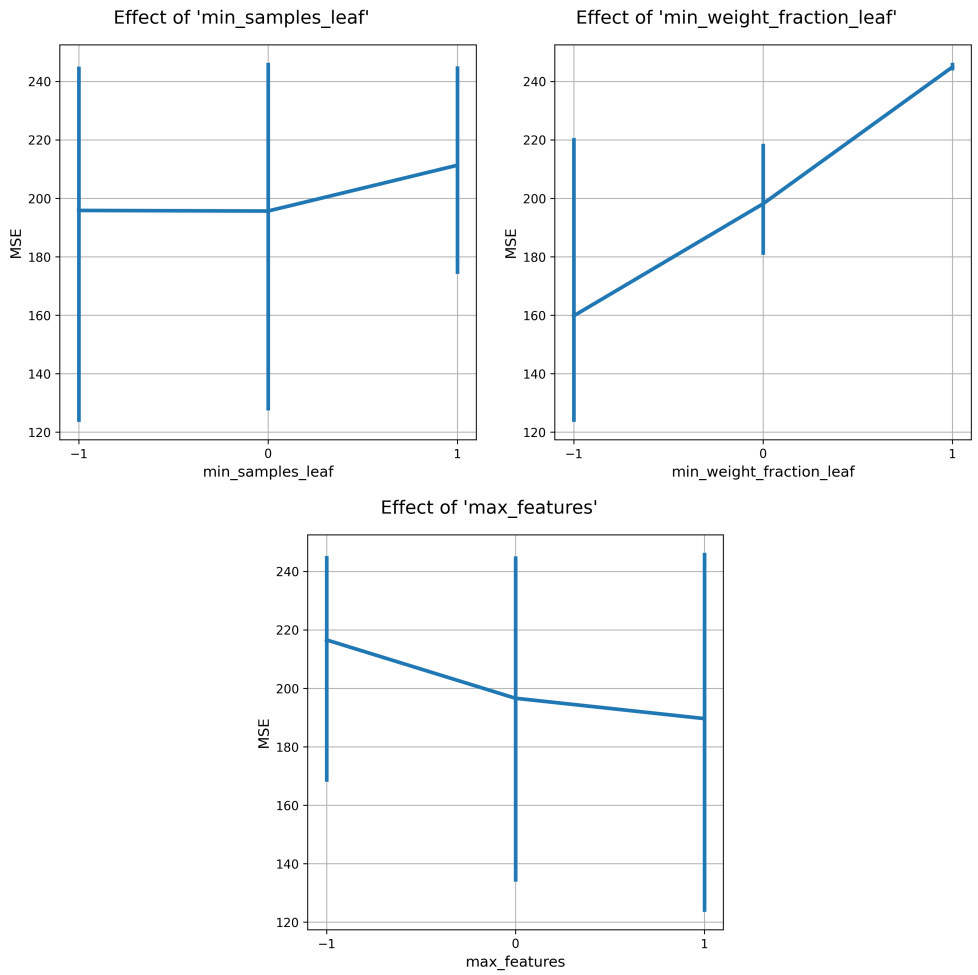


Figure 5: Independent factors effects

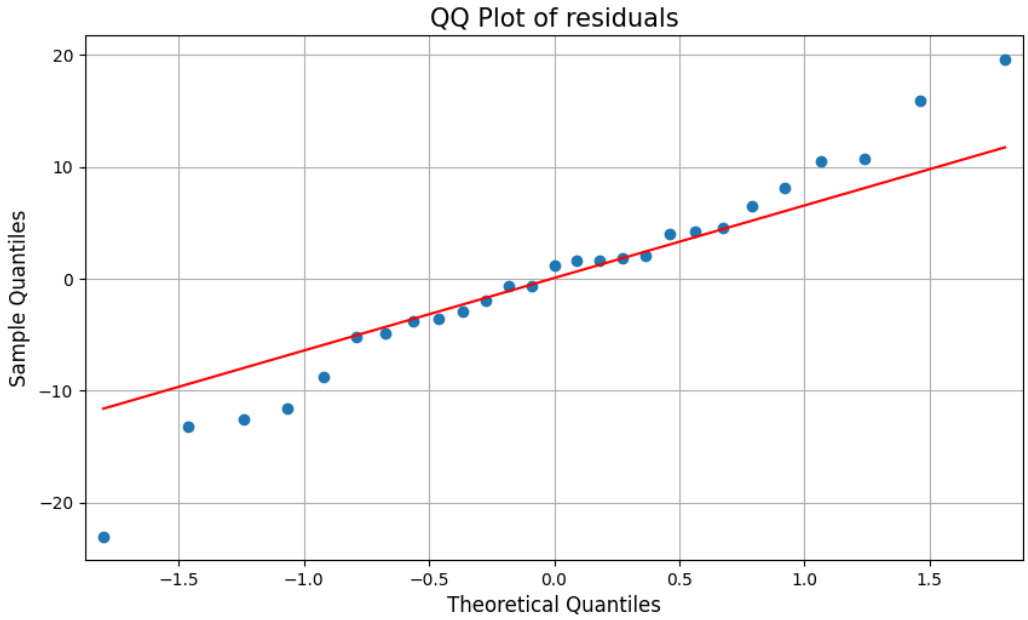


Figure 6: Q-Q plot of model residuals

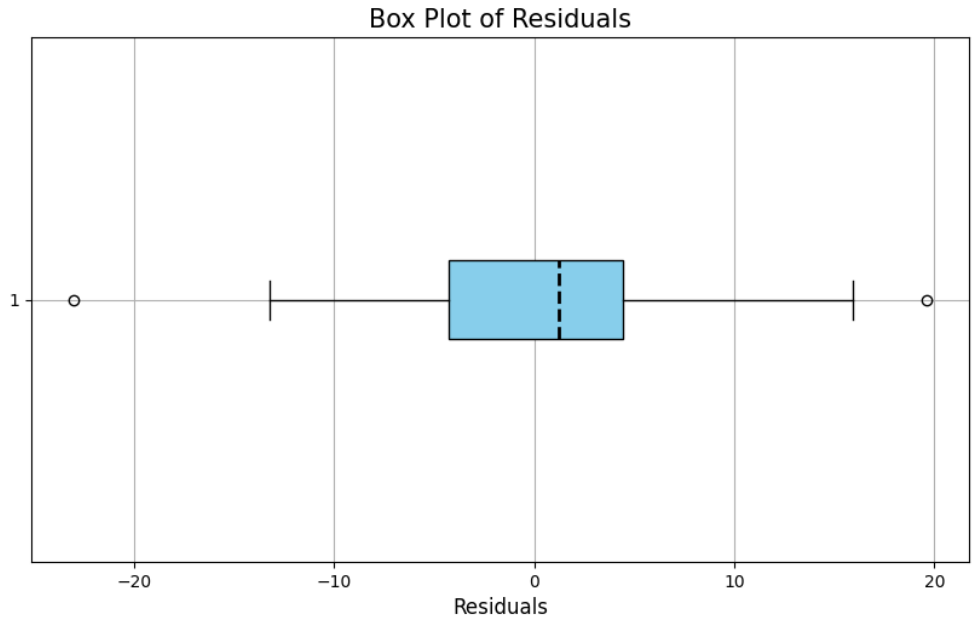


Figure 7: Box plot of model residuals

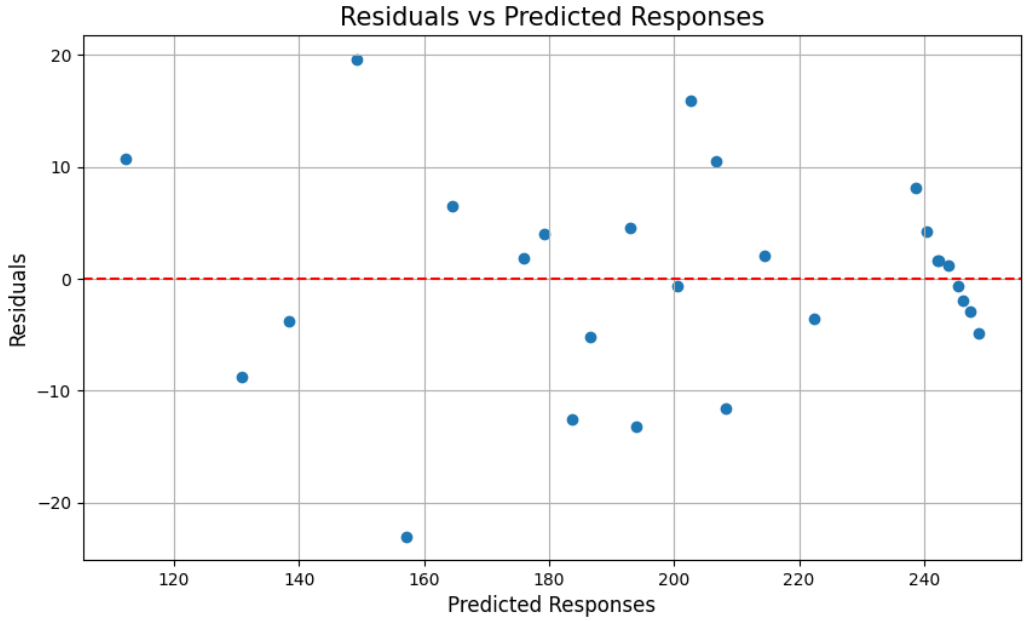


Figure 8: Independence of the residuals

Table 4: Settings for 3^3 full factorial design

factor	low	center	high
$min\_samples\_leaf$	10	**50**	100
$min\_weight\_fraction\_leaf$	0.1	**0.25**	0.4
$max\_features$	2	**4**	6

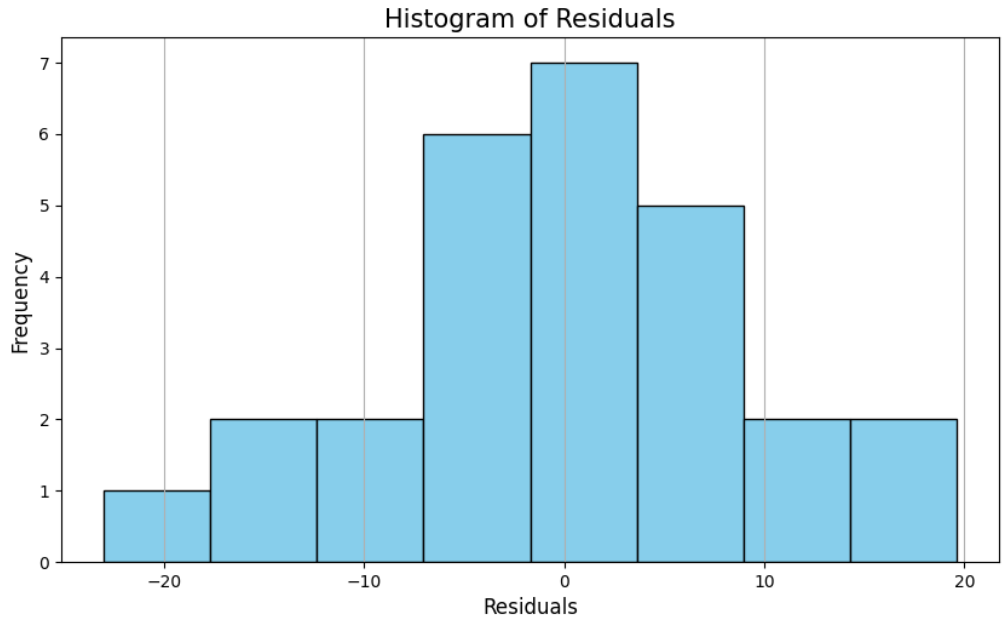


Figure 9: Residual spread around the mean

Table 5: *ANOVA* result of 3^3 full factorial design

Factors	SSE	F	PR(>F)
$min\_samples\_leaf$	1045.3	9.464	0.006
$min\_weight\_fraction\_leaf$	33820.9	306.206	1.4E-13
$max\_features$	3509.6	31.775	1.6E-05
$min\_samples\_leaf : min\_weight\_fraction\_leaf$	1494.2	13.528	0.001
$min\_samples\_leaf : max\_features$	0.917	0.008	**0.928**
$min\_weight\_fraction\_leaf : max\_features$	1859.0	16.831	0.0005
Residual	2209.032		

Table 6: Factor aliases

Factor	Alias	Confounding alias
max_depth	A	BF
$min\_samples\_split$	B	AF
$min\_samples\_leaf$	C	DEG
$min\_weight\_fraction\_leaf$	D	CEG
$max\_features$	E	CDG
max_leaf_nodes	F	AB
ccp_alpha	G	CDE

4.3 Box-Behnken design for response surface modeling

Finally a BBD is used to analyze the hyper-parameter space in more detail. The procedure is conducted in two iterations. For the first iteration the new centerpoints of the factors are assigned by taking the settings of the best run from the full factorial design. If the new centerpoint of a factor is a low or high level for that same factor in the full factorial

Table 7: Second order interactions

Factor interactions aliases	Confounded aliases
CD	EG
CE	DG
CG	DE
AC	BCF
AD	BDF
AE	BEF
AG	BFG
BC	ACF
BD	ADF
BE	AEF
BG	AFG
CF	ABC
DF	ABD
EF	ABE
FG	ABG

design, new values for the high and low levels are calculated. The difference between each level is constant. After setting up the new factors, a quadratic model is fitted to the results of the experimental design.

A GD procedure is employed to move in the direction of the steepest descending gradient in the hyper-parameter space. This procedure will locate the best factor combination in a close area around the centerpoint. The local minima will be used for the second iteration in the same manner.

The new values for the factors are shown in Table 10. Table 12 shows all the run combinations and their respective target values. To analyze the search space a quadratic model is fitted to the Table 12 data. The quadratic model serves as an approximation of the real search space of the DTR hyper-parameter combinations in a small region. This approximation allows stepping through the search space with simple calculations of gradients of the quadratic model. The quadratic model is shown in Equation 4, where $min\_samples\_leaf$ is aliased with A, $min\_weight\_fraction\_leaf$ with B and $max\_features$ with C. The parameters a_0 to a_{33} are the optimized model parameters after fitting to the data.

$$MSE = a_0 + a_1A + a_2B + a_3C + a_{12}AB + a_{13}AC + a_{23}BC + a_{11}A^2 + a_{22}B^2 + a_{33}C^2 \quad (4)$$

The combination of hyper-parameters shown in the first row of Table 12 is taken as the starting point of the GD procedure. Table 14 shows the results of five GD steps. Figure 10 visualizes the response surface from the first round. The ridge plot clearly shows that the MSE value decreases when both hyper-parameters have decreasing trends. However it must be taken into account that the quadratic model is only an approximation of the

Table 8: ANOVA on 2^{7-2} design

Factors	SSE	F	PR(>F)
max_depth	0.825	0.263	0.62
$min\_samples\_split$	4.692	1.498	0.252
$min\_samples\_leaf$	4524.8	1444.1	**3E-11**
$min\_weight\_fraction\_leaf$	44046.3	14057.6	**1.1E-15**
$max\_features$	5105.8	1629.5	**1.75E-11**
max_leaf_nodes	0.012	0.004	0.951
ccp_alpha	75.857	24.21	**0.0008**
$max\_depth : min\_samples\_split$	0.012	0.004	0.951
$max\_depth : min\_samples\_leaf$	1.13	0.361	0.563
$max\_depth : min\_weight\_fraction\_leaf$	1.661	0.53	0.485
$max\_depth : max\_features$	5.378	1.716	0.223
$max\_depth : max\_leaf\_nodes$	4.692	1.498	0.252
$max\_depth : ccp\_alpha$	7.265	2.319	0.162
$min\_samples\_split : min\_samples\_leaf$	0.714	0.228	0.644
$min\_samples\_split : min\_weight\_fraction\_leaf$	4.481	1.43	0.262
$min\_samples\_split : max\_features$	5.59	1.784	0.214
$min\_samples\_split : max\_leaf\_nodes$	0.825	0.263	0.62
$min\_samples\_split : ccp\_alpha$	4.69	1.497	0.252
$min\_samples\_leaf : min\_weight\_fraction\_leaf$	4526.4	1444.6	**3E-11**
$min\_samples\_leaf : max\_features$	54.08	17.26	**0.0024**
$min\_samples\_leaf : max\_leaf\_nodes$	6.529	2.084	0.183
$min\_samples\_leaf : ccp\_alpha$	5025.0	1603.7	**1.87E-11**
$min\_weight\_fraction\_leaf : max\_features$	5025.0	1603.7	**1.87E-11**
$min\_weight\_fraction\_leaf : max\_leaf\_nodes$	0.435	0.139	0.718
$min\_weight\_fraction\_leaf : ccp\_alpha$	54.08	17.26	**0.0025**
$max\_features : max\_leaf\_nodes$	0.11	0.035	0.856
$max\_features : ccp\_alpha$	4526.3	1444.6	**3E-11**
$max\_leaf\_nodes : ccp\_alpha$	2.057	0.656	0.439
Residual	28.199		

Table 9: Significant factors from ANOVA for 2^{7-1} design

Factors	SSE	F	PR(>F)
$min\_samples\_leaf$	9400.8	1625.8	6.2E-31
$min\_weight\_fraction\_leaf$	88531.0	15310.35	8E-48
$max\_features$	9481.1	1639.6	5.4E-31
$min\_samples\_leaf : min\_weight\_fraction\_leaf$	9433.0	1631.3	6E-31
$min\_samples\_leaf : max\_features$	84.257	14.571	0.0005
$min\_weight\_fraction\_leaf : max\_features$	9765.5	1688.8	3.2E-31

real search space, thus deviating too much from the origin point can produce unrealistic results.

For the second round, the same procedure is repeated. The new centerpoints for the BBD are taken from the step with the minimal value of MSE from Table 14. All the runs

are shown in Table 13. The best run in Table 13 has a value of 18 for $min\_samples\_leaf$ when the factor $max\_features$ is 7, as opposed to 28 when $max\_features$ is 6, which was calculated as a result of stepping through the search space in the previous round. This indicates that the search space is divided into planes for each $max\_features$ setting. The results of the previous GD procedure only found the local minima for one plane of the search space. Finally, a second GD procedure is conducted to explore the vicinity of the response surface. Table 15 shows the results of GD for the second round. Inspecting the results from Table 15 and Figure 11, a decreasing trend in both $min\_samples\_leaf$ and $min\_weight\_fraction\_leaf$ can be observed. The further decrease in $min\_samples\_leaf$ could lead to an overfitted model, since a new leaf is created based on small number of samples, so the procedure can be concluded. The hyper-parameter settings of the final model after the second round are shown in Table 16.

Table 10: Round 1 settings for 3 factor BBD with centerpoints design

factor	low (-1)	center (0)	high (+1)
$min\_samples\_leaf$	25	50	75
$min\_weight\_fraction\_leaf$	0.05	0.1	0.15
$max\_features$	5	6	7

Table 11: Round 2 settings for 3 factor BBD with centerpoints design

factor	low (-1)	center (0)	high (+1)
$min\_samples\_leaf$	18	28	38
$min\_weight\_fraction\_leaf$	0.005	0.025	0.03
$max\_features$	5	6	7

4.4 Comparison with baseline models

To showcase the performance of the best model that resulted from this analysis, a comparison is done against a severely underfitted model and a model with optimized hyper-parameters using an extensive grid search procedure. Grid searching is a time consuming procedure that trains a model with all possible combinations of provided values of hyper-parameters over a k number of folds for each combination. This results in a highly optimized model, however the procedure can take a long time to complete, especially with large amounts of data Subaşı, 2024. The hyper-parameter values used to conduct the grid search procedure are given in Table 17. In Table 16 all the model configurations used for comparison are shown.

This grid search procedure results in a total of 72576 fits, 24192 fits per cross validation fold over 3 folds. While the procedure described in this paper consists of only 150 fits of the DTR per fold over 3 folds. This is a considerable decrease in number of fits, which

Table 12: Round 1 BBD runs with target values

$min\_samples\_leaf$	$min\_weight\_fraction\_leaf$	$max\_features$	MSE
-1	-1	0	**91.55**
-1	1	0	151.34
1	-1	0	144.79
1	1	0	151.59
-1	0	-1	134.78
-1	0	1	118.89
1	0	-1	144.72
1	0	1	141.45
0	-1	-1	134.74
0	-1	1	113.57
0	1	-1	160.49
0	1	1	158.27
0	0	0	122.59

Table 13: Round 2 BBD runs with target values

$min\_samples\_leaf$	$min\_weight\_fraction\_leaf$	$max\_features$	MSE
-1	-1	0	82.64
-1	1	0	80.89
1	-1	0	104.84
1	1	0	106.06
-1	0	-1	82.03
-1	0	1	**79.45**
1	0	-1	103.35
1	0	1	111.21
0	-1	-1	89.48
0	-1	1	93.07
0	1	-1	90.77
0	1	1	93.27
0	0	0	92.91

Table 14: Result of first round GD

$min\_samples\_leaf$	$min\_weight\_fraction\_leaf$	$max\_features$	MSE
25	0.05	6	91.55
25.88	0.045	5.95	91.52
26.54	0.039	5.91	90.85
27.08	0.034	5.90	90.17
27.55	0.029	5.90	89.52
28	0.024	5.90	88.95

clearly shows the efficiency aspect of this method. The method produces a slightly worse model in both metrics than the one optimized by extensive grid searching. A comparison of the models can be seen in Figure 12.

Table 15: Result of second round GD

$min\_samples\_leaf$	$min\_weight\_fraction\_leaf$	$max\_features$	MSE
18	0.005	7	80.3
17.25	0.0048	7.0009	79.37
16.49	0.0047	7.0022	78.48
15.74	0.0045	7.004	77.61
14.97	0.0044	7.006	76.74
14.19	0.0043	7.008	75.88

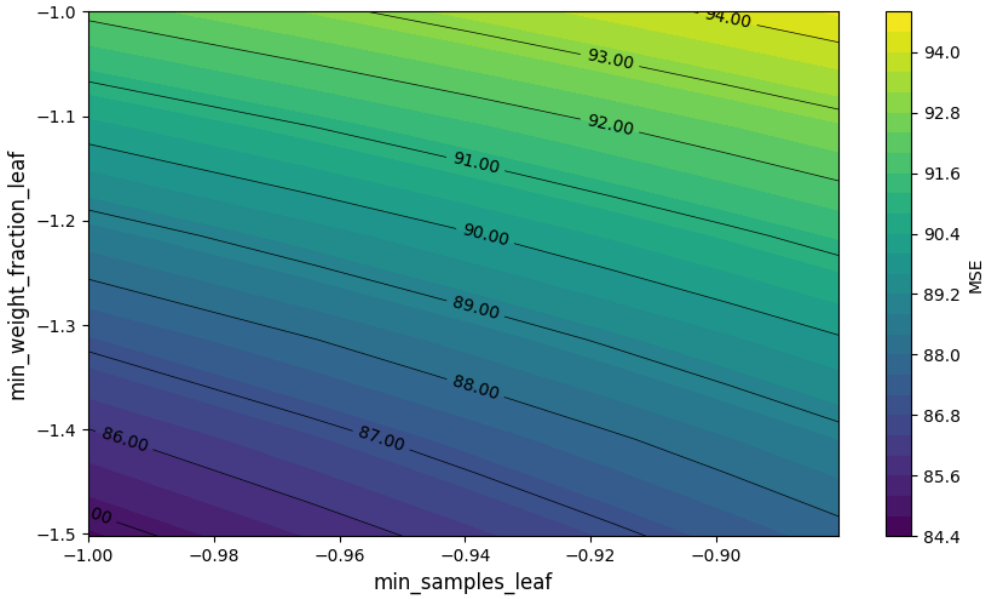


Figure 10: First round ridge plot

Table 16: Settings of best model, grid search model and underfitted model

factor	**best DTR**	grid search DTR	underfitted DTR
max_depth	None	10	1
$min\_samples\_split$	None	5	824
$min\_samples\_leaf$	14	5	412
$min\_weight\_fraction\_leaf$	0.0043	0	0.5
$max\_features$	7	None	1
max_leaf_nodes	None	100	2
ccp_alpha	0.0	0.02	1

A heavily underfitted model is trained to show the absolute range of MSE and R^2 values that can be achieved by using DTR on this dataset. Figure 13 shows the median values and percentiles around the median after evaluating each model with 10 cross validation splits.

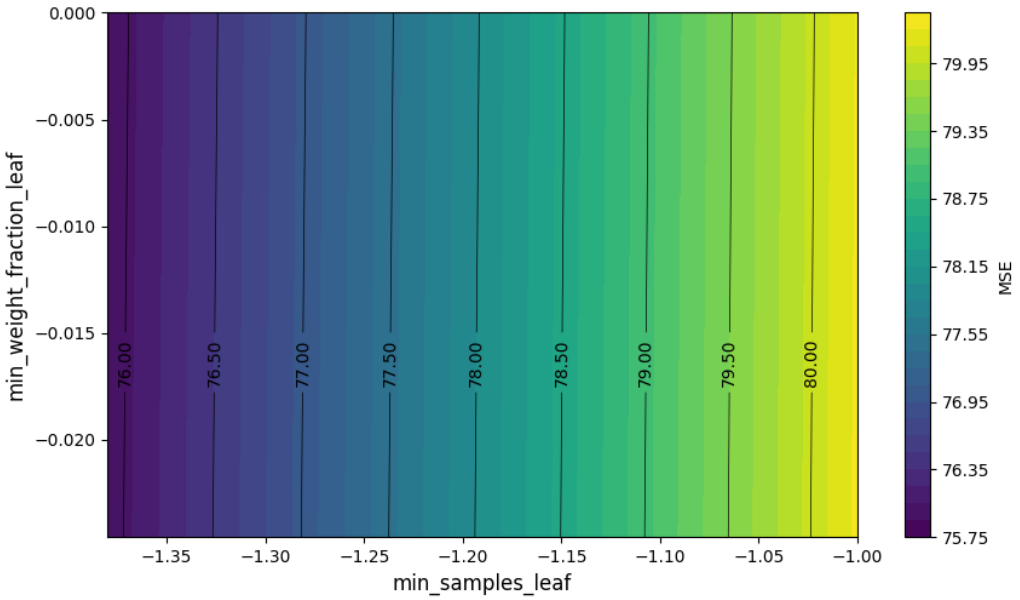


Figure 11: Second round ridge plot

Table 17: Grid search hyper-parameter values

Parameter name	values
max_depth	5, 10, 50, 100, 150, 200, None
$min\_samples\_split$	5, 15, 25, 35
$min\_samples\_leaf$	5, 15, 25, 35
$min\_weight\_fraction\_leaf$	0, 0.001, 0.2
$max\_features$	5, 6, None
max_leaf_nodes	2, 10, 100, None
ccp_alpha	0, 0.001, 0.02

5. Conclusions and Discussion

Tuning machine learning models often requires domain knowledge about the data, familiarity with the different hyper-parameters and their effects. On top of this requirements, training complex models on large datasets with classical approaches is very time consuming and often infeasible. This paper can be divided into two main parts. The first part of the paper focuses on understanding the effects of changing the factors and their interactions on the target variable. The second part however uses a response surface modeling technique to optimize the model by approximating the local search space and finding a local minima. In the first part of the paper, after conducting the 2^{7-2} design, several significant factors and interactions can be observed. After conducting $ANOVA$, high significance can be observed for 6 interaction between factors and 4 main factors. However due to the low resolution of the design it was not possible to distinguish the true significance of this interactions due to the confounding effect. To resolve this issue a 2^{7-1}

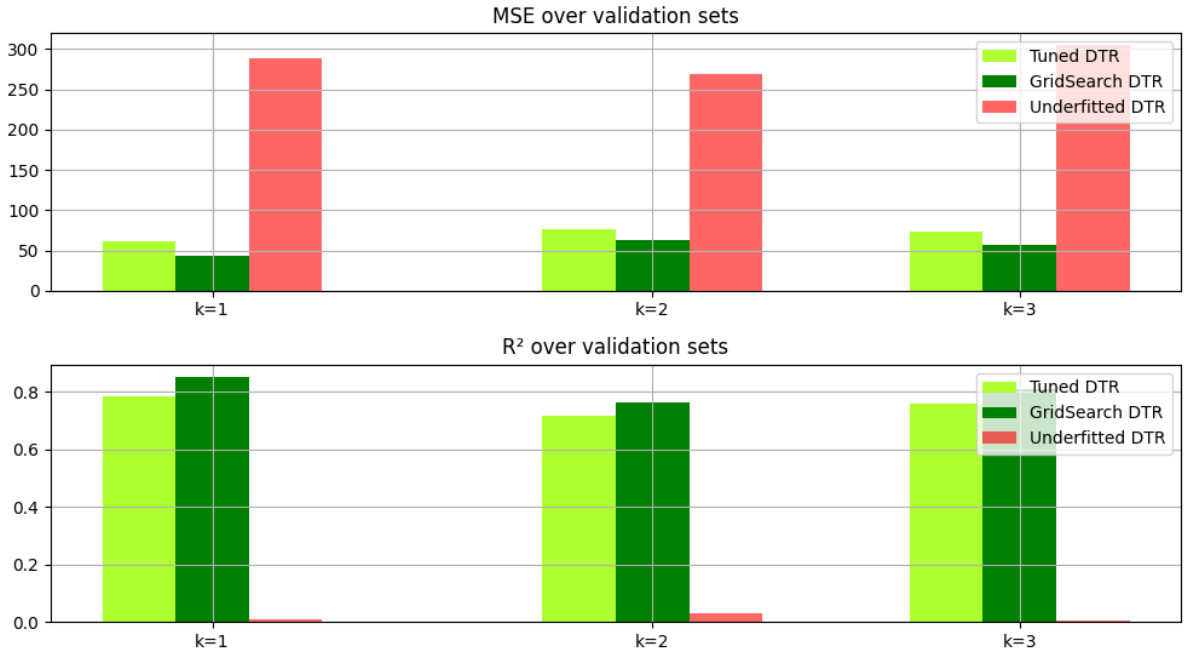


Figure 12: MSE and R^2 values over cross validation sets

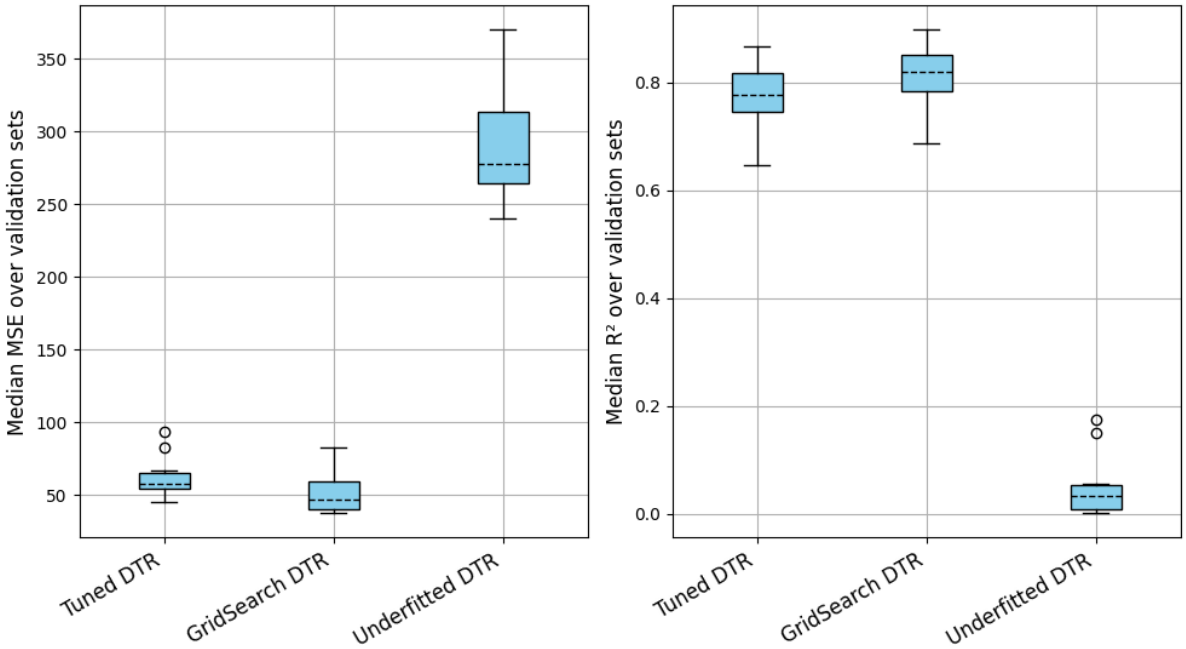


Figure 13: Median MSE and R^2 values over 3 cross validation sets

fractional factorial experimental design was conducted. After conducting $ANOVA$ on the results of this new design the significance of the factor ccp_alpha was not observed anymore. This clearly shows how the significance of some interactions can be wrongly interpreted due to confounding effects. One of the most interesting findings is that the factor max_depth was not significant at all in the initial range of 20 to 30. This might be due to bad initial setting assignment, which is confirmed by observing the value of max_depth in the grid search model. As a final step in the first part a full factorial design was conducted on only 3 significant factors: $min\_samples\_leaf$, $min\_weight\_fraction\_leaf$ and

$max\_features$. The settings for the full factorial design included all possible combinations of 3 factors with centerpoints. By adding centerpoints this design becomes a 3^3 full factorial design. The main effect plots show clear changes in the target values between different settings. However, the interaction plot $min\_samples\_leaf$: $max\_features$ shows no interaction between the factors. As a final step a series of regression analysis plots are created to visually check if the regression assumptions are satisfied. From the regression analysis plots it can be clearly seen that normality and constant variance of the $ANOVA$ model residuals is satisfied. The RSM procedure was conducted by using a Box-Behnken design. This design was applied on only 3 factors that showed high significance in the previous experiments. The fine tuning procedure using RSM was conducted in two iterations. Each iteration fits a quadratic model to the BBD experiment and includes stepping trough the approximated space using a GD algorithm. In the first iteration the origin point for the GD algorithm was fixed to be best run from the BBD. By using GD clear trends for each factor can be observed. $min\_samples\_leaf$ increases as the value of $min\_weight\_fraction\_leaf$ decreases. $max\_features$ is not considered to be changing since for each step its relative value only drops by a small amount. In the next iteration, new settings for the BBD are calculated around the settings of the last step of the GD of the previous round. After conducting GD in the second round an interesting behavior can be noticed. The starting point for the factor $min\_samples\_leaf$ is much lower than the value in the last step of previous GD procedure. Also the trend for this factor now decreases. This might indicate that the response surface is ridged and has local minima for each setting of the $max\_features$ factor. By decreasing the $min\_samples\_leaf$ factor the model might start to overfit to the data since it requires a small number of samples to make a leaf at each node. Factor $min\_weight\_fraction\_leaf$ still has a decreasing trend and most likely converges to zero. As in the previous round the factor $max\_features$ keeps a constant value.

The final model is compared to a highly optimized DTR model with a grid search procedure and a severely underfitted model. The tuned model with this methodology results in a median MSE value of 56.81 and a R^2 value of 0.78 over 10 cross validation folds. Both values show that our methodology resulted in a well generalizing model that performs well on not seen inputs. Comparing the tuned model to the grid search model a small difference in performance can be observed and a high improvement when comparing with the underfitted model. The real advantage of this methodology is in the reduced number of runs that are needed to get an accurate model, and to model the underlying system that governs the behavior of individual hyper-parameters and their interactions. Clearly, it must be noted that for two different datasets the best combination for a decision tree or any model will not be the same. Thus the same procedure and analysis must be repeated on a different dataset. The total number of runs that have been conducted to achieve these results is only 150 per fold over 3 folds, resulting in a total of 450 fits. In comparison, a grid search procedure with the hyper-parameter ranges shown in Table 17

requires a total of 72576 fits which might be unfeasible for a large dataset.

References

Blockeel, H., Devos, L., Frénay, B., Nanfack, G., & Nijssen, S. (2023). Decision trees: From efficient prediction to responsible ai. *Frontiers in Artificial Intelligence*, 6. https://doi.org/10.3389/frai.2023.1124553

Heckert, N., Filliben, J., Croarkin, C., Hembree, B., Guthrie, W., Tobias, P., & Prinz, J. (2002, November). *Handbook 151: Nist/sematech e-handbook of statistical methods*. NIST Interagency/Internal Report (NISTIR), National Institute of Standards; Technology, Gaithersburg, MD.

I, Y. (1998). Concrete compressive strength [dataset]. *UCI Machine Learning Repository*. https://doi.org/https://doi.org/10.24432/C5PK67

Lujan-Moreno, G. A., Howard, P. R., Rojas, O. G., & Montgomery, D. C. (2018). Design of experiments and response surface methodology to tune machine learning hyperparameters, with a random forest case-study. *Expert Systems with Applications*, 109, 195–205. https://doi.org/https://doi.org/10.1016/j.eswa.2018.05.024

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

Probst, P., Boulesteix, A., & Bischl, B. (2019). Tunability: Importance of hyperparameters of machine learning algorithms. *Journal of Machine Learning Research*, 20.

Rosa, A., Riccardo, C., Luca, P., & Luigi, S. (2020). Design of experiment-based configuration of hyperparameters of an artificial neural network. *JSM 2020 - Section on Statistical Learning and Data Science*.

Subaşı, N. (2024). Comprehensive analysis of grid and randomized search on dataset performance. *European Journal of Engineering and Applied Sciences*, 7, 77–83.

Vanschoren, J., van Rijn, J. N., Bischl, B., & Torgo, L. (2013). Openml: Networked science in machine learning. *SIGKDD Explorations*, 15(2), 49–60. https://doi.org/10.1145/2641190.2641198

Yang, Z., & Zhang, A. (2021). Hyperparameter optimization via sequential uniform designs. *Journal of Machine Learning Research*, 22(149), 1–47.