Chair of Physical Metallurgy

Master's Thesis

# Impact of measurement errors on machine learning models in steel production

Andreas Rechberger, BSc

October 2024

## EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich diese Arbeit selbstständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt, den Einsatz von generativen Methoden und Modellen der künstlichen Intelligenz vollständig und wahrheitsgetreu ausgewiesen habe, und mich auch sonst keiner unerlaubten Hilfsmittel bedient habe.

Ich erkläre, dass ich den Satzungsteil „Integrität im wissenschaftlichen Studien-, Lehr- und Forschungsbetrieb“ der Montanuniversität Leoben gelesen, verstanden und befolgt habe.

Weiters erkläre ich, dass die elektronische und gedruckte Version der eingereichten wissenschaftlichen Abschlussarbeit formal und inhaltlich identisch sind.

Datum 08.Okt.2024

Reichberger A

(Die Originalsignatur ist an der Universität hinterlegt)

# ACKNOWLEDGEMENTS

I would like to thank the many people who have provided valuable comments and discussions during the writing of this thesis. My thanks go to Gerfried Millner, Manfred Mücke, Christoph Kickinger, Daniel Scheiber, Michael Schmid, and David Wurm for their insights and support.

Special thanks go to my supervisor, Lorenz Romaner, for providing the topic of this Master's thesis and for offering guidance and feedback throughout this process.

Lastly, I would like to thank my friends and family who have made my time in Leoben a wonderful and pleasant experience supporting me both academically and personally.

# ABSTRACT

The use of machine learning (ML) models to describe material properties has become an important research direction in materials science and many relevant datasets are becoming available. However, the systematic treatment of uncertainties associated with measurement errors remains a challenging topic. This thesis provides an investigation of the impact of measurement errors on the prediction accuracy of ML algorithms for the mechanical properties of steels.

In the first part, this topic is addressed using artificially generated linear and non-linear datasets. It is shown that the errors in the target variable $\mathbf{y}$, although affecting prediction accuracy more strongly than the errors in the features $\mathbf{X}$, average out if $\mathbf{y}$ is normally distributed. Errors in the features $\mathbf{X}$, on the other hand, lead to bias in relation to the true correlation that systematically offsets the prediction with respect to the true value.

In the second part, this thesis aims to provide a deeper understanding of the nature of the prediction error via the bias-variance decomposition. It analyses how separate determination of the measurement error and investigation of Learning Curves can be used to quantify bias or variance of the ML model. This is demonstrated on a martensite starting temperature ($\mathrm{M}_s$) data set and a r-value data set. For the $\mathrm{M}_s$ dataset, it is found that for underparametrized models such as linear regression, the training error and validation error converge well above the measurement error indicative of bias. For overparametrized models such as XGBoost or Random Forest, the training error is smaller than the measurement error, while the validation error is sizably higher. Therefore, these models exhibit variance and would benefit from more data points. The performance on the validation set of XGBoost, Random Forest, and Gaussian Process regression is comparable. For the r-value dataset, XGBoost reveals behavior similar to that of the $\mathrm{M}_s$ dataset. Finally, an analysis of ML models available from the literature to predict the mechanical properties obtained from tensile testing reveals that the model's validation error is close to the measurement error. Therefore, the bias and variance of these models are small and, therefore, the prediction accuracy is higher than the validation error (or the associated coefficient of determination $R^2$) suggests.

# Contents

# List of Figures

## List of Tables

# NOTATION

In this thesis, we adopt different notations to denote different mathematical entities that are commonly used in machine learning [1]:

- Scalars are denoted by italic letters such as $x = 1$
- Vectors are denoted by bold lowercase letters, like **x**.

$$\mathbf{x} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

- Matrices are denoted by bold capital letters. For example, **X**.

$$\mathbf{X} = \begin{bmatrix} 1 & -2 & 2 \\ 3 & 1 & 0 \end{bmatrix}$$

# 1 Introduction

The use of ML models to describe material properties has become a popular method since they can cope with the large number of factors that can influence the production of materials. In the steel industry, many machine learning models have been created to model the mechanical properties of steel from the industrial production route [2–11]. Another representative example of the suitability of ML methods is the prediction of the martensite start temperature ($M_s$) of steels [12–14] via the chemical composition.

However, when performing an experiment such as a tensile test, there will always be some noise due to factors that cannot be 100 % controlled. Some of the measurement noise comes from the geometry and machining of the test piece, or the characteristics of the test machine (stiffness, drive, and control mode), to name a few factors [15]. Bhadeshia gives an overview of the applications and implications of neural networks in materials science. He also mentions the importance of accounting for uncertainties in material science in his paper [16]. Heidl and colleagues thoroughly examine the impact of noise, model bias, and model variance on chemical property predictions. By conducting controlled experiments on datasets of molecular properties, they reveal significant patterns in model performance linked to the degree of noise present in the dataset [17]. Relevant to this topic is also an early work that has pointed out the conceptual difference between the error in the target variable and the input variable [18]. More recently, a detailed description of uncertainty in deep learning has also been provided. [19].

The aim of this thesis is to gain a better understanding of the impact of measurement uncertainties on prediction accuracy. Gaining insight into these uncertainties can significantly improve the decision-making process of what steps to take next in a machine learning project. In this thesis, the influence of measurement errors is first investigated on synthetic data sets where the noise is applied to the input and output variables at different levels. In this case the ground truth is known. This will be followed by an examination of real-world data sets and considerations on how a known measurement error can be used to evaluate the machine learning model. In addition, a brief comparison of the existing literature on machine learning models used to predict mechanical properties from tensile testing will be provided.

# 2 Fundamentals

This section describes the concepts of measurement error and how it can be divided into random and systematic errors. The focus then shifts to the uncertainties in machine learning and the additional complexities encountered when applying machine learning models in real-world scenarios. A brief summary of key machine learning models and performance metrics is presented along with a short analysis of the bias-variance trade-off.

## 2.1 Measurement Error

A measurement error or observational error is the difference between a measured value of a quantity and its unknown true value. Generally, measurement errors can be divided into random errors and systematic errors [20].

Statistical fluctuations in measured data due to precision limitations of the measurement device are known as **random errors**. These errors can be evaluated by statistical analysis and reduced by averaging over a large number of observations. **Systematic errors** are inaccuracies that occur consistently in the same direction. They are difficult to detect and cannot be analyzed statistically. If a systematic error is identified during calibration against a standard, a correction or correction factor should be applied to compensate for the effect. Unlike random errors, systematic errors cannot be detected or reduced by increasing the number of observations [21].

The total measurement error is the sum of systematic and random errors [22]. Figure 1 illustrates the concept of measurement error. The x-axis represents the measured value and the y-axis shows the frequency of each measured value.The curve itself represents the distribution of measured values around the measured value. The spread of the curve indicates the variability or precision of the measurements. A narrower curve would suggest more precise measurements, while a wider curve would indicate less precision [23].

## 2.2 Classification of Uncertainty in Machine Learning

There are two main categories of uncertainty that affect machine learning algorithms **epistemic and aleatoric** (Figure 2). The word aleatory derives from the Latin *alea*, which

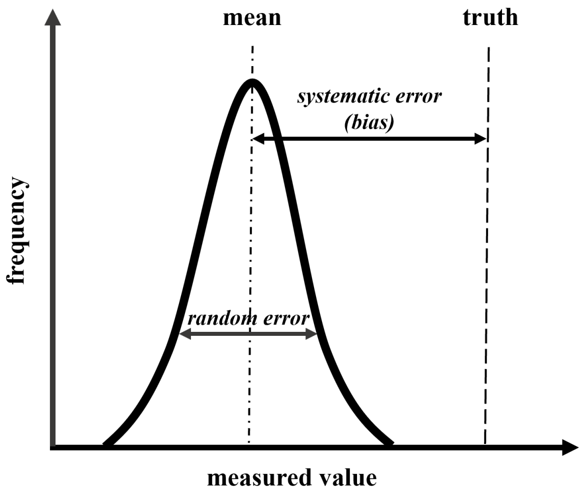


**Figure 1:** *Illustration of the cumulative effect of systematic and random error.*

means the rolling of dice, and the word epistemic derives from the Greek *episteme*, which means knowledge [17, 24, 25].

Aleatoric uncertainty is also known as stochastic uncertainty and is representative of unknowns that differ each time we run the same experiment. For example, the measurement of martensite start ($M_s$) temperature with dilatometric data. This inconsistency can be attributed to uneven temperatures across the small samples placed inside the dilatometer or other factors, as explained by Bhadeshia [16].

Epistemic uncertainty, also known as systematic uncertainty, arises from information that could be known in principle, but is not known in practice. This can occur when the model ignores certain effects. An example of this would be the increase in the $M_s$ temperature as a result of the formation of carbides. This depletes the matrix of carbon and the $M_s$ temperature increases to higher temperatures [26]. The fraction of carbides could be measured and incorporated into the model to reduce the resulting uncertainty. However, as this measurement is difficult and costly, this is not done in most models, and therefore not all relevant variables are included in the model.

Even so in a practical application of a machine learning model, some additional uncertainty is to be expected. This can be understood by looking at the onion layer model (Figure 3) of the prediction uncertainty described by Kläs [27]. The three uncertainties, **scope compliance, data quality** and **model fit**, are stacked on top of each other.

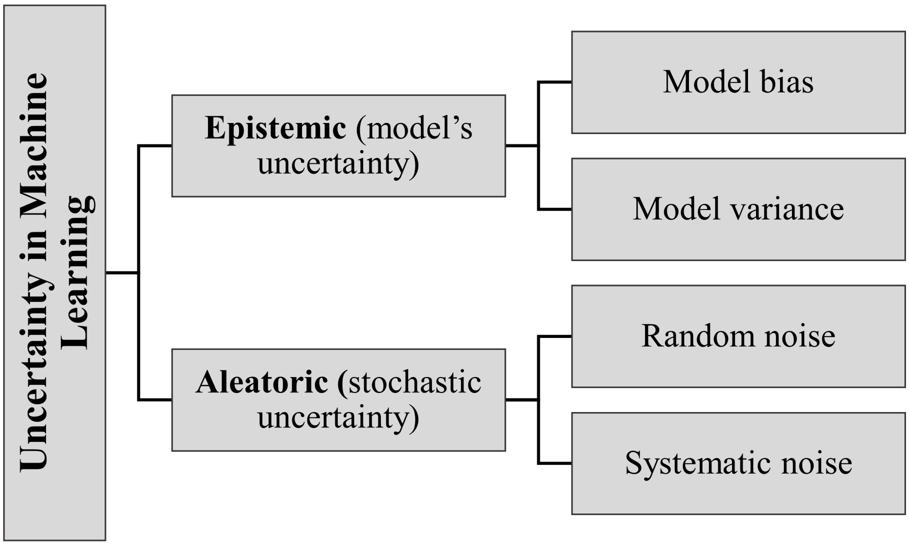

Uncertainty in Machine Learning
Epistemic (model's uncertainty)
Aleatoric (stochastic uncertainty)
Model bias
Model variance
Random noise
Systematic noise

**Figure 2:** *Division of the uncertainty in machine learning (based on [17].)*

Uncertainty in model fit is caused by the fact that machine learning techniques provide empirical models that are only an approximation of the real (functional) relationship between the model input and its output. The accuracy of this approximation, which is limited, has been discussed above in the text.

In a real environment, all types of collected data (e.g. based on sensors but also human input) are limited in their accuracy and potentially affected by various types of quality issues. Thus, the actual level of uncertainty in the output of a machine learning model is affected by the quality (especially the accuracy) of the data on which it is currently being applied. Therefore, additional uncertainty resulting from a delta between the quality of the cleaned data and the data on which the model is currently being applied can be defined as data quality (induced) uncertainty. An example would be that the chemical analysis of the steel is less accurate than it was in training. This would lead to a less precise prediction in the $M_s$ temperature.

Scope compliance addresses the issue of how machine learning models are built and tested in a specific context. If these models are used outside of that context (scope compliance), their results may become unreliable (because the model has to extrapolate). An example would be that the model was trained on lean C-Mn steels, but now we want to use it on tool steels.

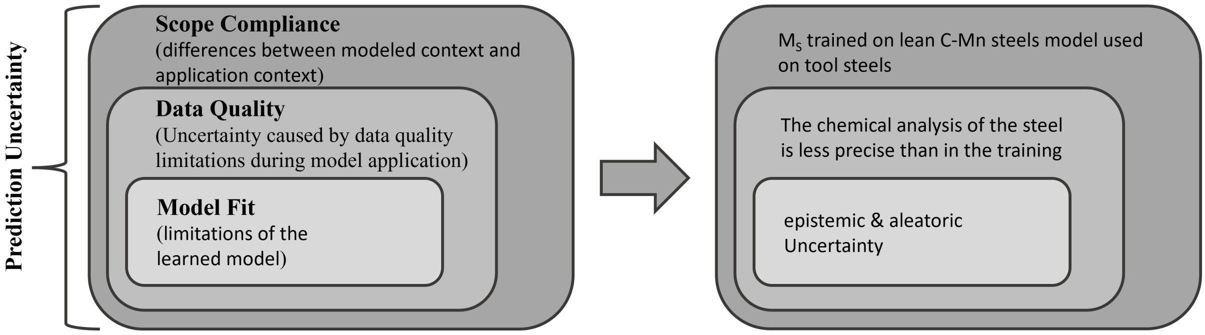

Prediction Uncertainty
Scope Compliance
(differences between modeled context and application context)
Data Quality
(Uncertainty caused by data quality limitations during model application)
Model Fit
(limitations of the learned model)
$M_S$ trained on lean C-Mn steels model used on tool steels
The chemical analysis of the steel is less precise than in the training
epistemic & aleatoric Uncertainty

**Figure 3:** *Onion layer model of the prediction uncertainty.*

## 2.3 The Bias-Variance Decomposition

In general, when our machine learning model wants to predict a target point $y$, we use our features $\mathbf{x}$. We assume that there is a relationship that relates to the other such as $y = f(\mathbf{x}) + \epsilon$ where the error term $\epsilon$ is normally distributed with a mean of zero. This leads to the formulation of a machine learning model $\hat{f}(\mathbf{x})$ for the given point $\mathbf{x}$, as an approximation of the true function $f(\mathbf{x})$. This results in the expected squared prediction error:

$$\text{Err}(x) = \mathbb{E}\left[(y - \hat{f}(x))^2\right] \tag{1}$$

This error can then be decomposed into bias and variance components[28]:

$$\text{Err}(x) = \left(\mathbb{E}\left[\hat{f}(x)\right] - f(x)\right)^2 + \mathbb{E}\left[\left(\hat{f}(x) - \mathbb{E}\left[\hat{f}(x)\right]\right)^2\right] + \sigma_\epsilon^2 \tag{2}$$

$$\text{Err}(x) = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error} \tag{3}$$

The decomposition into bias variance and irreducible error as shown above is described in more detail in the book by Hastie et al. [29]. Then the accuracy of a machine learning model's predictions can essentially be broken down into two key elements: the error rooting from "bias" and the error resulting from "variance". Just like many situations in life, there is a balancing act involved in a model's capacity to reduce bias and variance simultaneously. This principle is called the Bias-Variance trade-off. When the model is underfitting (Figure 4), it means it has high bias, as the model has not learnt enough information from the training set and does not capture the relationship between the features **X** and the target variable **y**. When the model is overfitting, it means that it has high variance and the model is too closely related to the examples in the training set and does not generalise well to other examples. Besides these errors, there is also something called an Irreducible Error, which is basically the noise in the data. Noise is random and cannot be predicted, so we cannot lower this type of error.

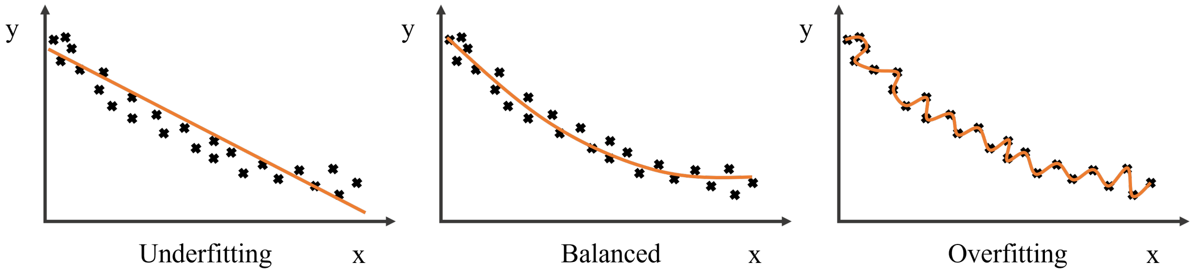


**Figure 4:** *Demonstration of overfitting and underfitting.*

## 2.4 Machine learning evaluation metrics

This subsection describes some important regression metrics: the mean squared error MSE, the root mean squared error RMSE, and the coefficient of determination $R^2$.

### 2.4.1 Mean Squared Error MSE

The mean squared error (MSE) measures the amount of error in statistical models. It is the average squared difference between the observed and predicted values. When a model has no error, the MSE is equal to zero. As the model error increases, its value increases. Mathematically, it's represented as:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{4}$$

where $y_i$ are the observed values, $\hat{y}_i$ are the predicted values, and $n$ is the number of observations. A graphical representation can be seen in Figure 5.

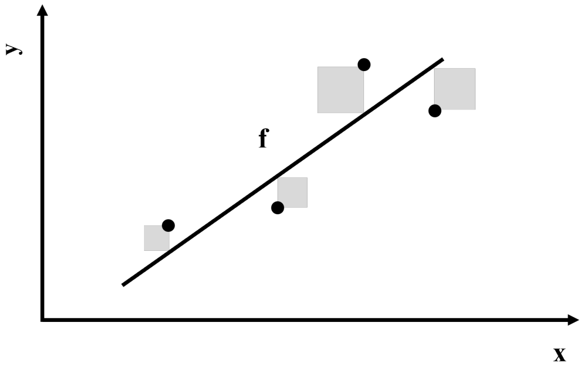


**Figure 5:** *Graphical representation of the calculation of MSE.*

### 2.4.2 Root Mean Squared Error RMSE

The root mean square error (RMSE) is a standard method for measuring the error of a model to predict quantitative data. It is defined as the square root of the mean squared error (MSE):

$$\text{RMSE} = \sqrt{\text{MSE}} \tag{5}$$

The RMSE has the benefit of being in the same units as the original data. In addition, it serves as a measure of the dispersion of residuals, similar to standard deviation (SD) in its role and application. This similarity to the univariate case allows us to understand that the RMSE represents the typical deviation of residuals from the regression line, essentially indicating the vertical scatter of data points around this line. Furthermore, RMSE benefits from an empirical rule similar to SD, where approximately 68 % of observations are expected to lie within ±1 RMSE. Moreover, the principle that about 95% of the data values are found within ±2 RMSE holds true for many datasets, although there are exceptions where these approximations may not apply [30].

### 2.4.3 Coefficient of Determination R$^2$

R$^2$ is a statistical measure that represents the proportion of the variance for the dependent variable that's explained by the independent variables in a regression model. It is calculated as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{6}$$

where $y_i$ is the actual observed values of the dependent variable in the data, $\hat{y}_i$ the predicted values of the dependent variable obtained from the regression model, $\bar{y}$ the mean (average) of the actual observed values of the dependent variable in the data and $n$ the number of data points in the data set. If $R^2$ is 1, this means that the predictions of the model are perfectly in line with the real results. If $R^2$ is 0.7, that means that the model explains 70 % of the variance. The remaining 30 % of the variance is not explained by the model, suggesting that other factors not included in the model might account for this portion of the variance.

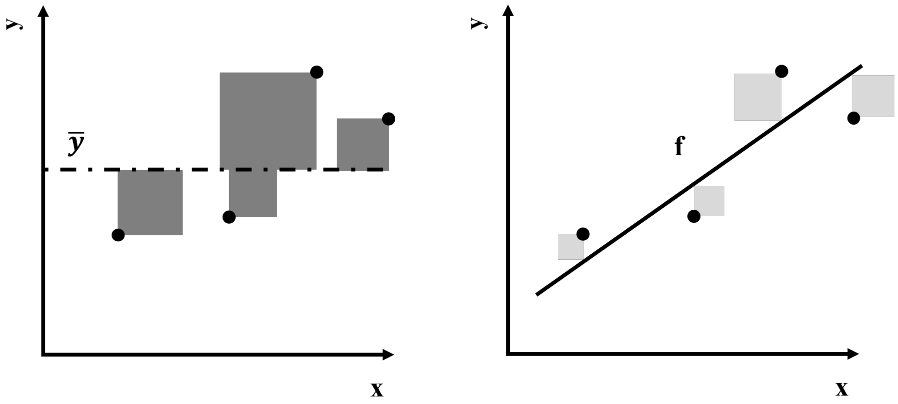


**Figure 6:** *Graphical representation of the calculation of $R^2$. The total sum of squares is represented in the left plot while the residual sum of squares is shown to the right.*

## 2.5 Machine Learning Models

This section provides a brief overview of several machine learning models, focussing on tree ensemble learning algorithms, Gaussian process regression (GPR), and neural networks. It is important to note that these are just a few examples from a wide range of models used in the field. Other commonly used models are, for example, Support Vector Machines SVMs[31].

### 2.5.1 Tree Ensemble Learning Algorithms

Tree ensemble learning algorithms combine the outputs of multiple trees to produce more accurate results. Two ensemble techniques are **Bagging** and **Boosting**. Bagging and Boosting are both ensemble techniques, but they differ in their approach to combining models. Bagging aims to introduce diversity by training models simultaneously on different subsets of the data.(Figure 7). Boosting, on the other hand, focuses on sequential training and adaptive learning, where each model attempts to correct the errors of the previous ones[32] (Figure 8). A popular example of bagging is the **Random Forest (RF)** algorithm [33], which builds a collection of decision trees. Each tree is trained on a different bootstrap sample and the final prediction is determined by aggregating the predictions of individual trees. A popular example of boosting is the **eXtreme Gradient Boosting (XGBoost)** algorithm [34]. It is an advanced implementation of gradient boosting algorithms designed for speed and performance.

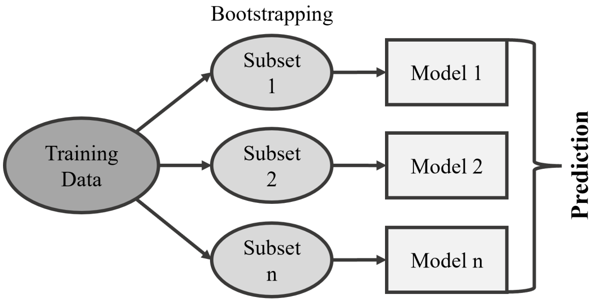


**Figure 7:** *Illustration of Bagging.*

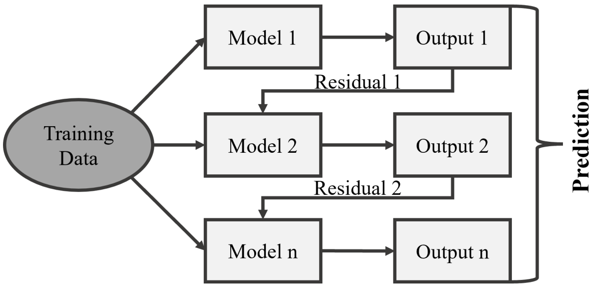


**Figure 8:** *Illustration of Gradient Boosting.*

### 2.5.2 Gaussian Process Regression (GPR)

GPR are widely used for non-linear, non-parametric regression modelling. They predict outcomes using prior knowledge encapsulated in what are known as kernels. Because different kernels can be specified, the GPR model can become very versatile. However, this is also a drawback, as it raises the question of which kernel to use for a given problem [35]. Its strength lies in the use of Bayesian statistics to produce mean predictions with confidence intervals. Traditional GP models assume that any noise affects only the output values, while the input values are assumed to be perfectly accurate. However, there is some research that considers the noise in features **X** as well [36, 37]. A significant limitation of these models is their inefficient performance as the size of the dataset increases. Because an inverse matrix has to be calculated, the GPR is slow for more than a few thousand data points[38, 39].

### 2.5.3 Neural Network

A vanilla neural network is constructed of artificial neurons (Figure 10) arranged in layers, including an input layer, a hidden layer, and an output layer (Figure 9). Each neuron applies an activation function to its input, introducing non-linearity to the model and enabling the network to capture complex patterns. The standard architecture of neural networks is feedforward, meaning that the information flows forward from the input to the output layer, with each layer's output serving as the next layer's input. When a neural network contains more than one hidden layer, it is referred to as a multilayer neural network or a deep neural network. Adding more hidden layers can enhance the network's ability to fit complex functions, although it increases the computational cost and the risk of overfitting.

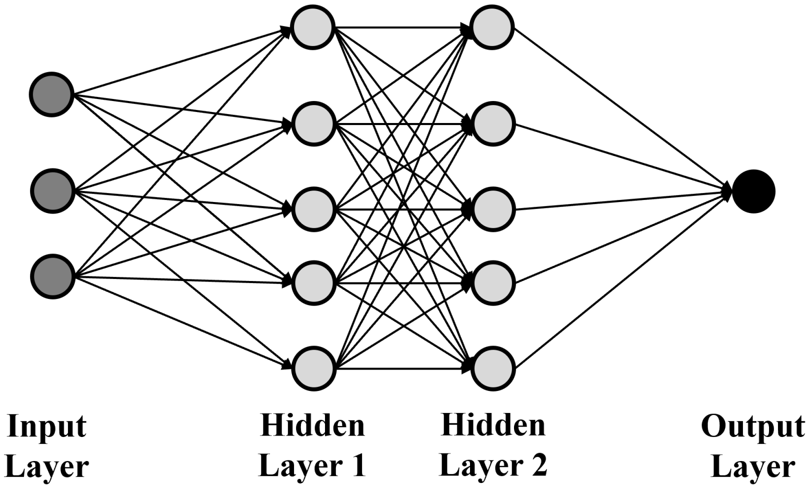


**Figure 9:** *The structure of a multilayer neural network.*

Neural networks learn by adjusting the weights of connections between neurons, primarily using a method called backpropagation. This learning process is enhanced by the use of activation functions, which introduce non-linearity to the network, allowing it to model complex relationships in the data. The choice of activation function is crucial as it affects the network's ability to converge during training and its overall performance. Common activation functions include the sigmoid, tanh, and ReLU (Rectified Linear Unit) function. Each of these functions has its own characteristics and is chosen based on the specific requirements of the neural network and the nature of the problem being solved. Together with the network architecture, these elements form the foundation of a neural network's ability to tackle a wide array of tasks [40, 41].

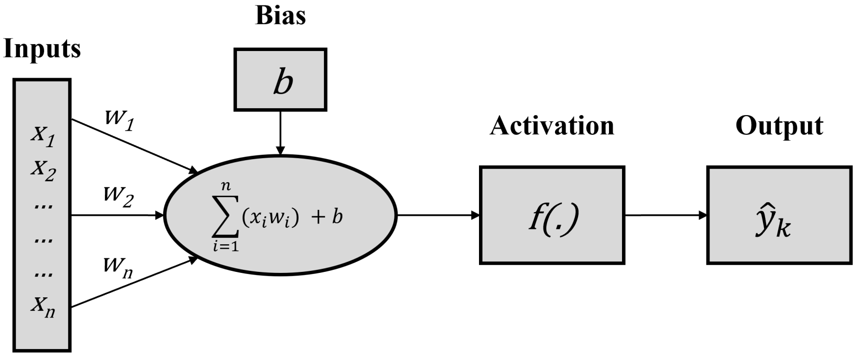


**Figure 10:** *The structure of an artificial neuron.*

# 3 Methods

The code was implemented in Jupyter Notebooks using Python 3.11. For data manipulation and visualisation, the Pandas [42], NumPy[43], Seaborn [44] and Matplotlib [45] libraries were used. Machine learning models were constructed and evaluated using the XGBoost [34] and Scikit-learn [46] libraries. The methods used to investigate errors on artificial datasets and to investigate datasets from the real world Learning Curve are described separately.

# 4 Investigation of errors on artificial data sets

This section of the examination focuses on artificial datasets. One advantage of this approach is that the ground truth is known. The following section will examine three datasets.

## 4.1 Procedure of Investigation on artificial data sets

Figure 11 shows the general workflow of noise analysis, where the first step is to create artificial data sets. This involves defining a mathematical function that simulates the relationship between the features (independent variables) and the target (dependent variable). Three functions were used to generate synthetic data sets: a simple linear function (equation 7), a Renard Series R10 linear function (equation 8) and the Friedman 1 nonlinear function (equation 9).

$$y = x + d \tag{7}$$

$$\begin{aligned} y = {} & x_0 + 1.25x_1 + 1.6x_2 + 2x_3 + 2.5x_4 + 3.15x_5 \\ & + 4x_6 + 5x_7 + 6.3x_8 + 8x_9 + 10x_{10} \end{aligned} \tag{8}$$

$$y = 10\sin(\pi x_0 x_1) + 20(x_2 - c)^2 + 10x_3 + 5x_4 \tag{9}$$

The sample size $n$ for all data sets includes 5000 data points. For the linear function, the input feature values $x_i$ are uniformly distributed on the interval $\mathcal{U}[0.5, 2]$ and the intercept $d = 1$. The input feature for the Renard Series R10 $x_{0,i}...x_{10,i}$ and for the Friedman 1 $x_{0,i}...x_{4,i}$ are uniformly distributed on the interval $\mathcal{U}[0, 1]$. The constant variable was chosen as c = 0.5 in the Friedman 1 function.

After the artificial data were generated, the next step was to introduce Gaussian noise to both the features and the target to simulate measurement errors. The noise in the feature was defined as a product of the mean value of the feature $\overline{x}$ and a noise factor $\sigma$ that is normally distributed and varied between 0 and 0.5 The same was done for the noise in the y-value and $\overline{y}$ represents the mean value of the target value:

$$x_{\text{noise}} = \overline{x}\mathcal{N}(0, \sigma) \tag{10}$$

$$y_{\text{noise}} = \overline{y}\mathcal{N}(0, \sigma) \tag{11}$$

This leads to the following corrupted relationship for the data sets:

$$[y + y_{\text{noise}}] \approx [x + x_{\text{noise}}] + d \tag{12}$$

$$\begin{aligned}[y + y_{\text{noise}}] \approx {} & [x_0 + x_{0,\text{noise}}] + [1.25x_1 + x_{1,\text{noise}}] + [1.6x_2 + x_{2,\text{noise}}] \\ & + [2x_3 + x_{3,\text{noise}}] + [2.5x_4 + x_{4,\text{noise}}] + [3.15x_5 + x_{5,\text{noise}}] \\ & + [4x_6 + x_{6,\text{noise}}] + [5x_7 + x_{7,\text{noise}}] + [6.3x_8 + x_{8,\text{noise}}] \\ & + [8x_9 + x_{9,\text{noise}}] + [10x_{10} + x_{10,\text{noise}}] \end{aligned} \tag{13}$$

$$\begin{aligned}[y + y_{\text{noise}}] \approx {} & 10\sin(\pi[x_0 + x_{0,\text{noise}}][x_1 + x_{1,\text{noise}}]) \\ & + 20([x_2 + x_{2,\text{noise}}] - c)^2 \\ & + 10[x_3 + x_{3,\text{noise}}] + 5[x_4 + x_{4,\text{noise}}] \end{aligned} \tag{14}$$

The data sets were then divided into a training set and a test set. The training set is used to fit the model and for cross-validation, while the test set was used to evaluate performance on unseen data and to measure accuracy against the ground truth function in this particular case. This should mimic the normal workflow of a machine learning project. The split ratio is 80 % for training and 20 % for testing.

The next step is to train the machine learning models using the training set. For each noise variation, a model was trained. For the corrupted linear dataset (equation 12) and the noisy Renard Series R10 dataset (equation 13) a linear ordinary least squares regression model was used and for the corrupted Friedman 1 nonlinear dataset a XGBoost (equation 14) model has been trained.

To evaluate the performance of the model, a 10-fold cross-validation is used. This involves splitting the training data set into ten parts, using nine parts for training and one part for validation. This process is repeated ten times, each part being used once as the validation set. This method provides a more reliable estimate of the performance of the model. The performance measures chosen are $R^2$ and RMSE. The performance of the model was also evaluated against the ground truth of the noise-free function defined with the test set. This evaluation provides insight into how well the model can capture the underlying relationship defined by the function.

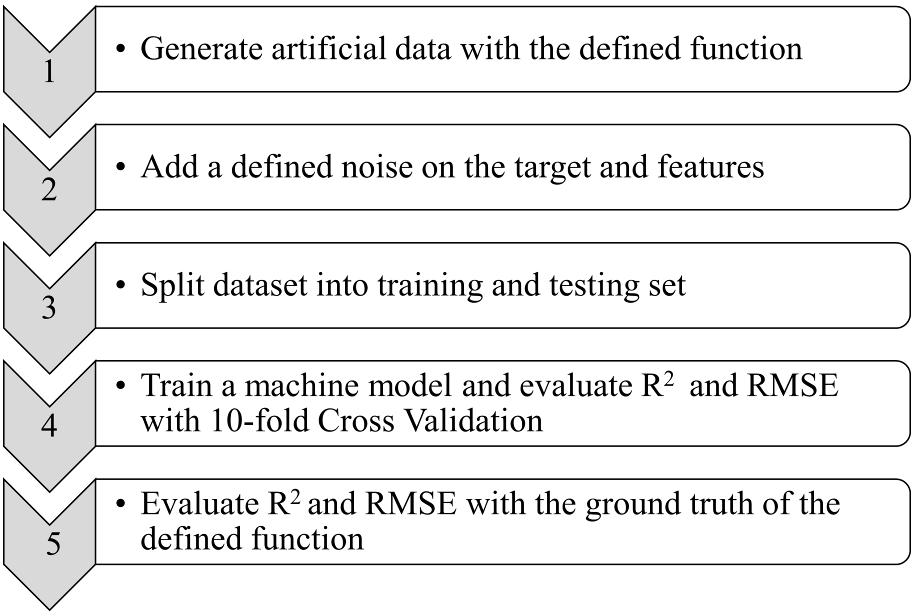

1
• Generate artificial data with the defined function
2
• Add a defined noise on the target and features
3
• Split dataset into training and testing set
4
• Train a machine model and evaluate R$^2$ and RMSE with 10-fold Cross Validation
5
• Evaluate R$^2$ and RMSE with the ground truth of the defined function

**Figure 11:** *Workflow of investigation of errors on artificial data sets.*

## 4.2 Results of errors on artificial data sets

Figure 12 shows the results of the linear regression with a noise factor of $\sigma = 0.15$ on feature and the target variable. The black line is the ground truth function, and the orange dots show the predicted values. The blue dots are the 20 % test data. Because this graphical representation is only possible for the linear function, a different presentation was developed. This can be seen in Figure 13. This 3D surface plot illustrates the relationship between all applied noise levels and $R^2$. In principle, Figure 12 corresponds to one point on the black surface and one point on the coloured surface. The x and y axes represent the noise factor, while the z axis represents the value of $R^2$. The black surface represents $R^2$ calculated using the true function (equation 7) and the trained models. The coloured surface below represents $R^2$ from cross-validation calculated using observed data (equation 12) and trained models. The colour gradient indicates the magnitude of $R^2$. Red or grey represents higher values, closer to 1, indicating a better fit, while blue or black represents lower values, closer to 0, indicating a poorer fit. The same graphical representation was done for the linear dataset Renard Series R10 (Figure (14) and the nonlinear dataset Friedman 1 (Figure (15).

The 3D surface plot was also created for the RMSE. Figure 16 shows this for the Renard Series R10 dataset and Figure 17 for the Friedman 1 dataset. The y-axis represents the standard deviation of the added noise, while the x-axis shows the average standard deviation of the noise applied to each feature. The black surface shows the RMSE calculated using the true function and the trained models. The coloured surface shows the RMSE calculated from cross-validation using observed data and trained models. The colour gradient indicates the magnitude of the RMSE, with red or grey representing higher values and blue or black representing lower values. When only noise is applied on **y**, the RMSE matches the standard deviation of the noise applied to the target variable.

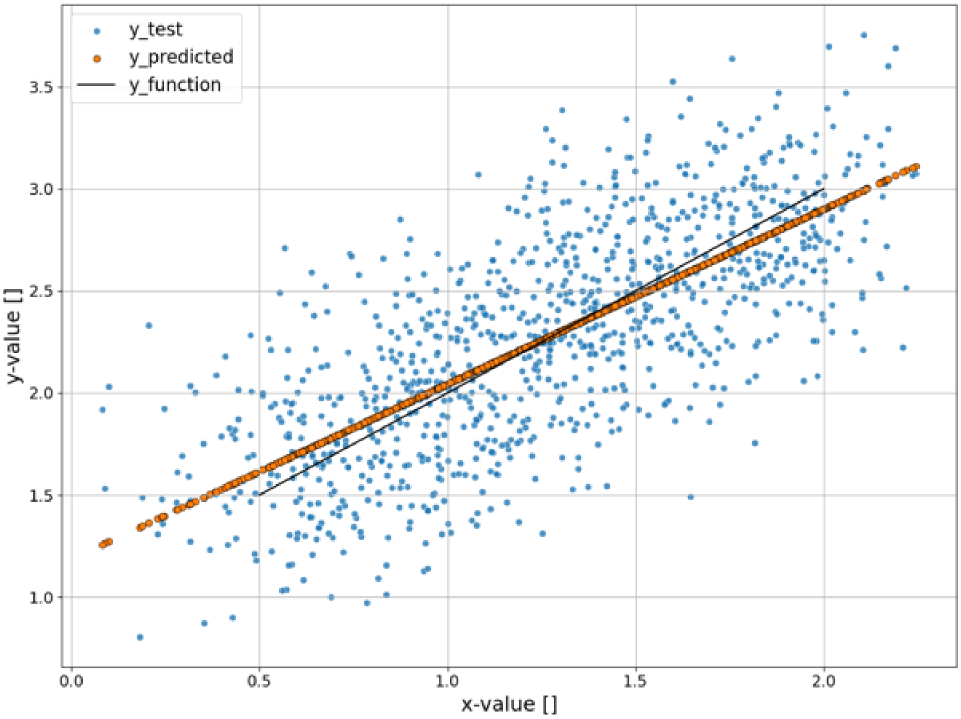


**Figure 12:** *Results of the linear regression. The black line shows the ground truth function, orange dots show the predicted values and blue dots show the testing data.*

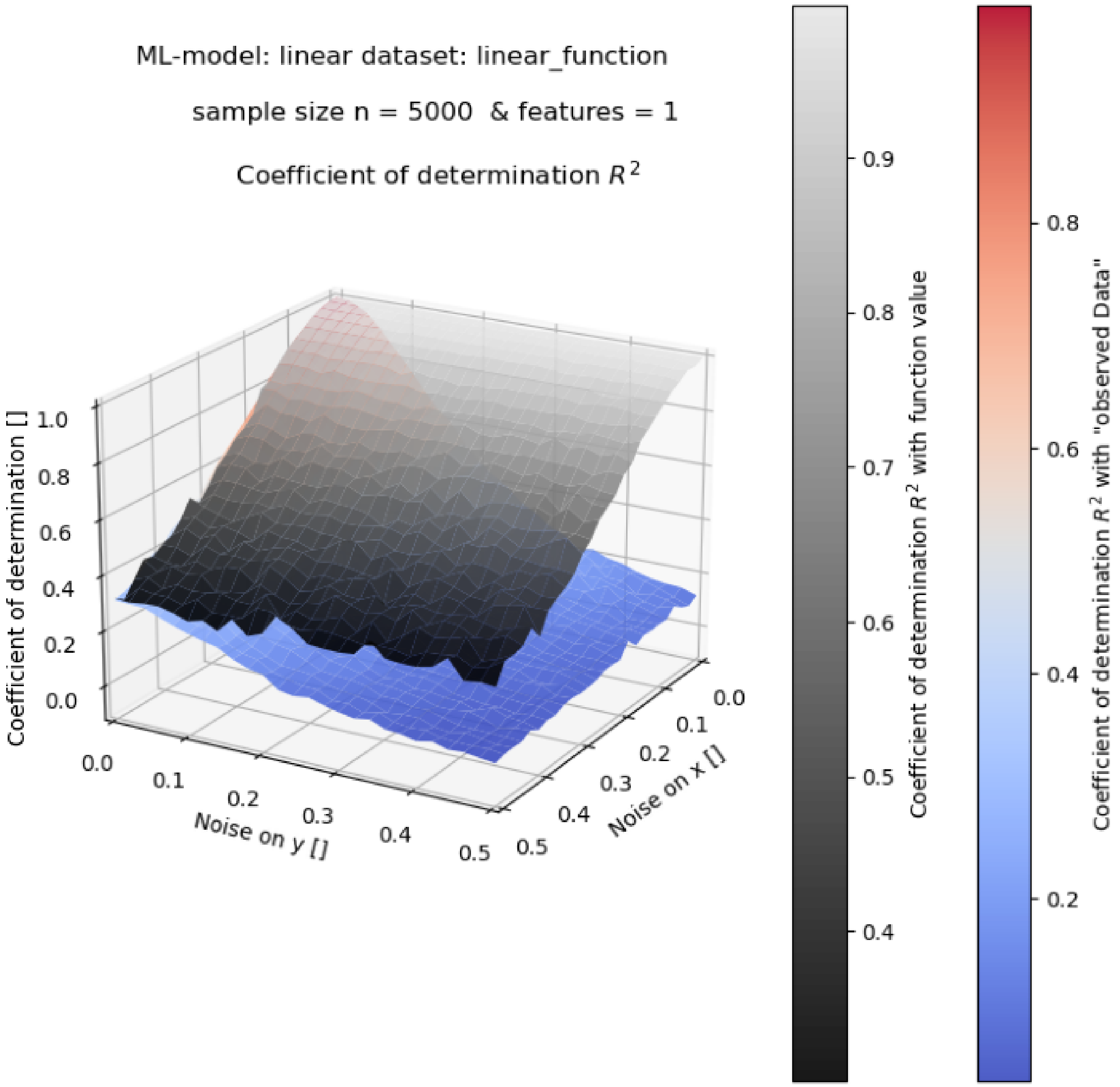


**Figure 13:** *Results of the linear regression. Black surface shows $R^2$ calculated with the true function value while the colored surface shows $R^2$ calculated for the observed data.*

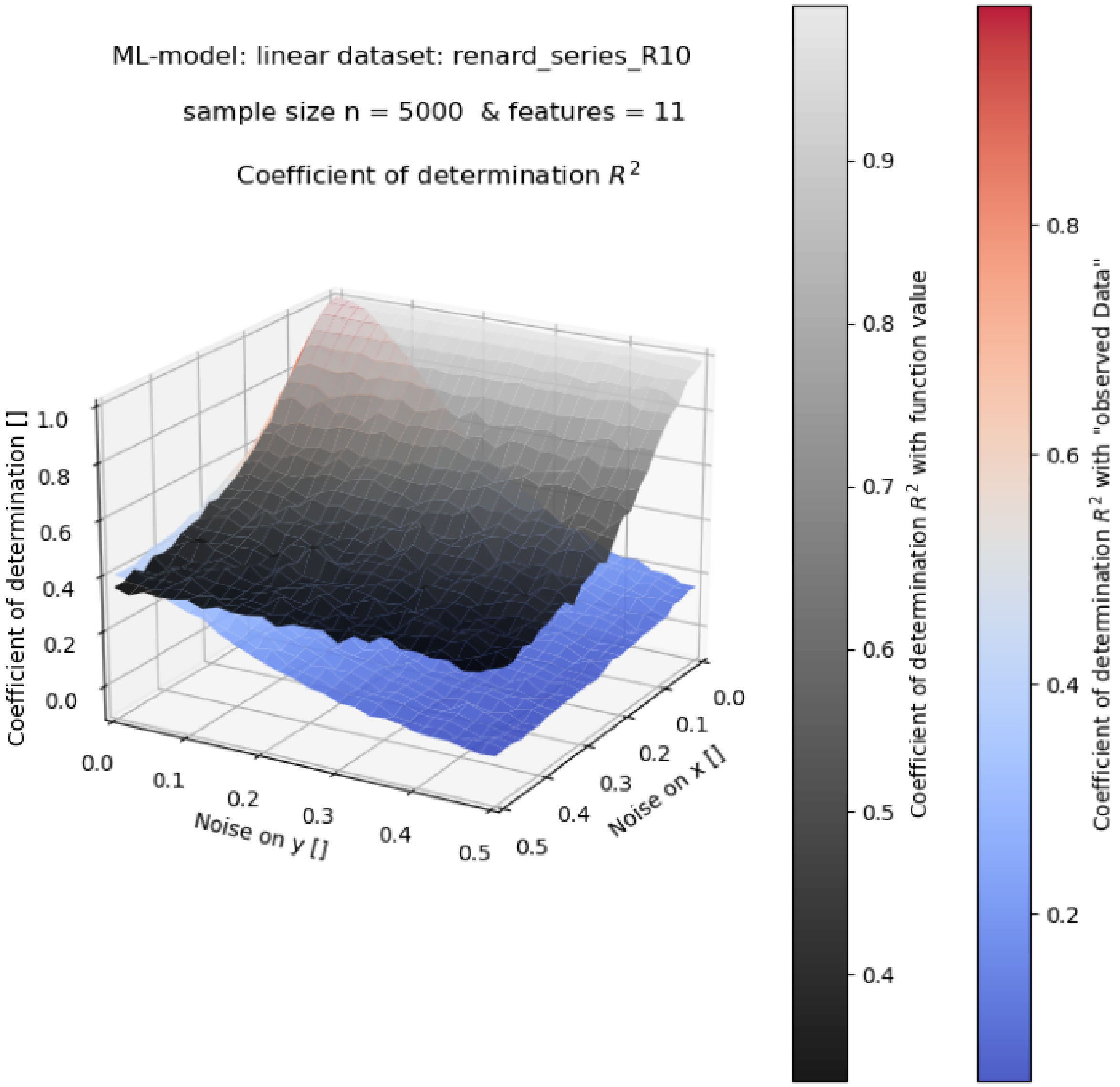


**Figure 14:** *Results of the linear regression. Black surface shows $R^2$ calculated with the true function value while the colored surface shows $R^2$ calculated for the observed data.*

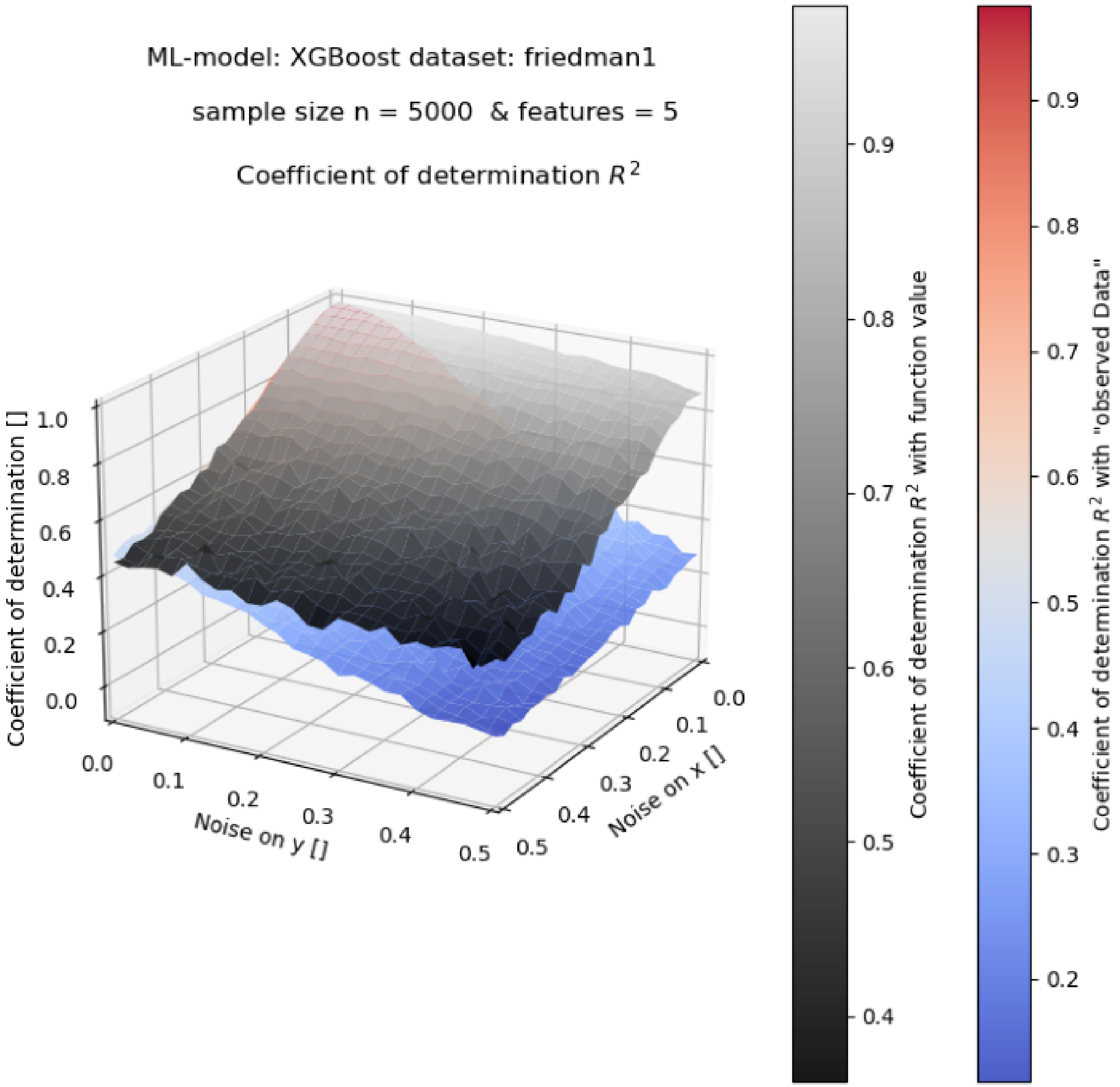


**Figure 15:** *Results of the XGBoost Model. Black surface shows $R^2$ calculated with the true function value while the colored surface shows $R^2$ calculated for the observed data.*

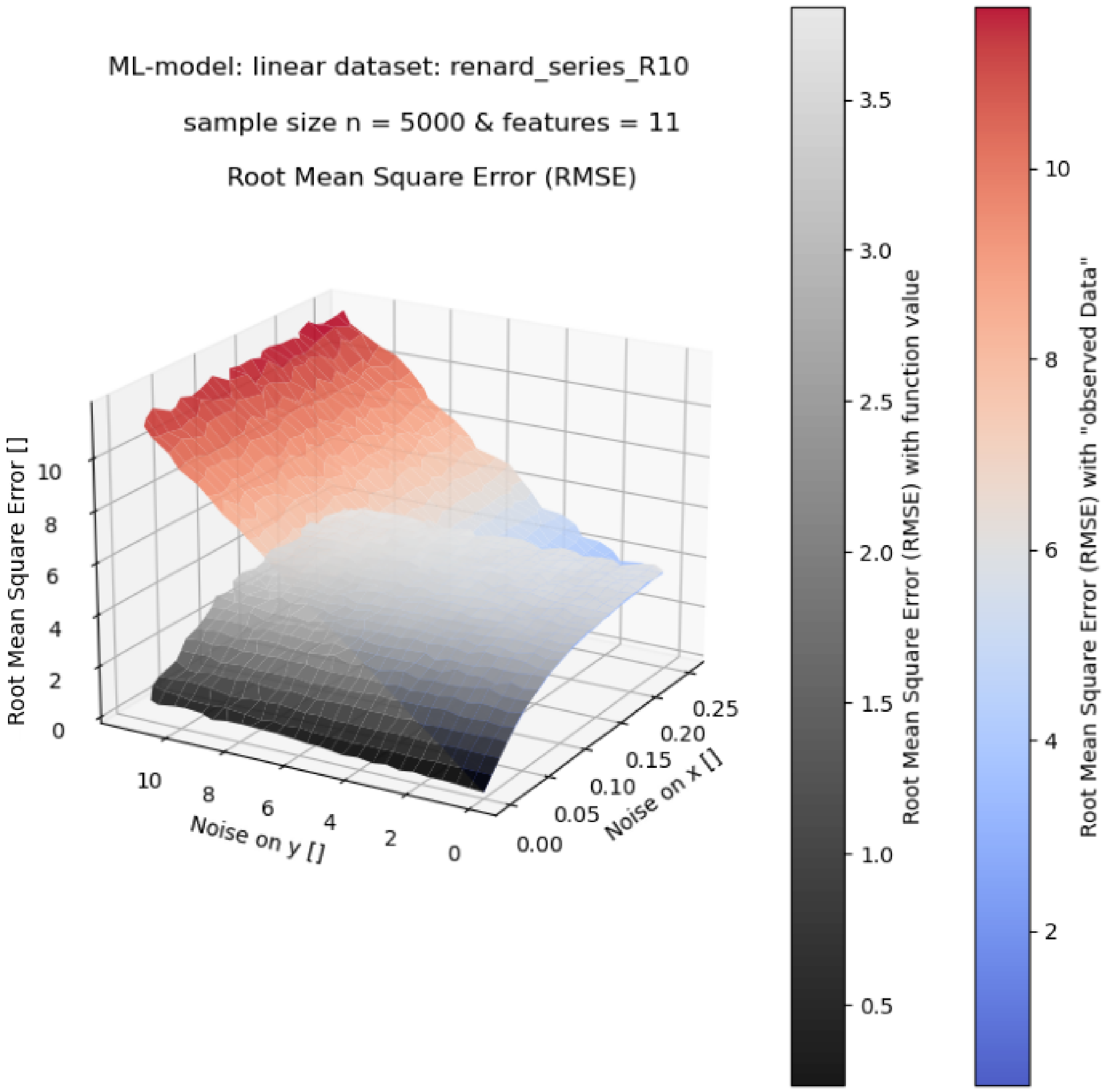


**Figure 16:** *Results of the linear regression. Black surface shows RMSE calculated with the true function value while the colored surface shows RMSE calculated for the observed data.*

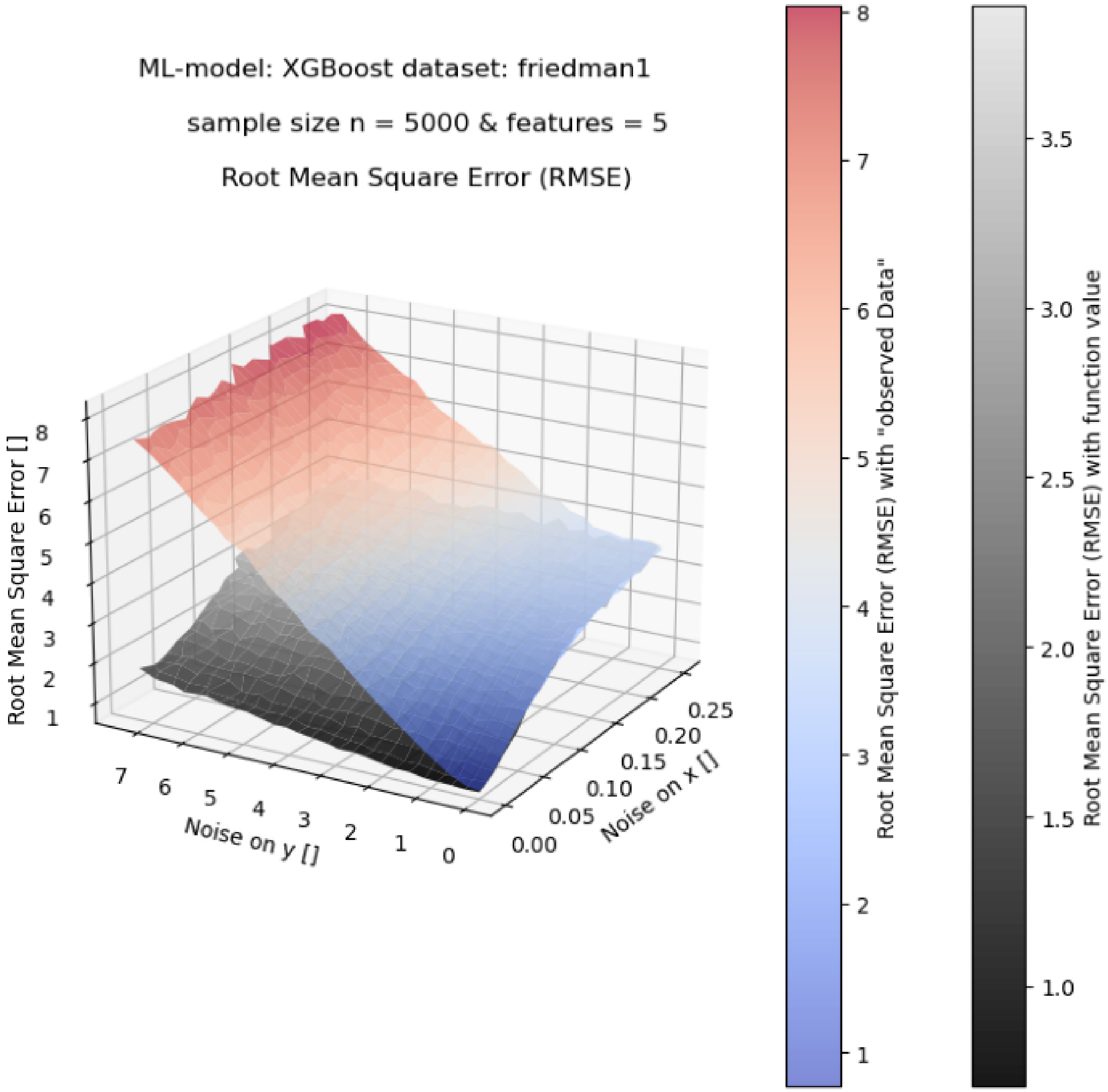


**Figure 17:** *Results of the XGBoost Model. Black surface shows RMSE calculated with the true function value while the colored surface shows RMSE calculated for the observed data.*

## 4.3 Bridging the gap between synthetic data analysis and real-world data analysis - The Learning Curves

Real-world machine learning models trained on noisy datasets correspond to a point on the coloured surface on the graps from the previous section. The ground truth function (black surface) is not known. Expanding on the findings from the previous analysis, we have determined that under conditions where the measurement error follows a normal distribution, the RMSE aligns with the standard deviation of the measurement error in an ideal model scenario. So in many material science problems, the optimal error rate would be the measurement error of the predicted properties. However, there is still an epistemic error to consider. One heuristic way to evaluate a model that suffers from bias or variance is to plot a Learning Curve and compare it to the optimal error rate, as described by Ng [47].

In Figure 18 an example Learning Curve is shown. The x-axis represents the number of training examples used to train the model, and the y-axis measures the error of the model. There are two main lines showing the training error and the mean validation error. The black dashed horizontal line represents the expected measurement error. In many material science problems, a good approximation of the optimal error rate would be the measurement error of the predicted properties. This allows us to estimate the bias and variance of the model:

$$\text{Variance} \approx \text{Validation error - Training error} \tag{15}$$

$$\text{Bias} \approx \text{Training error - Optimal error} \tag{16}$$

If the training error curve deviates significantly from the measurement error line, the model is considered to have high bias. When the validation error curve is far from the training error curve, it signifies that the model exhibits variance. An interesting point to note is that if the model would have negative bias, this implies that the model's accuracy on the training data is higher than the optimal error rate, indicating overfitting due to the memorization of the training data, which means that the model suffers from variance.

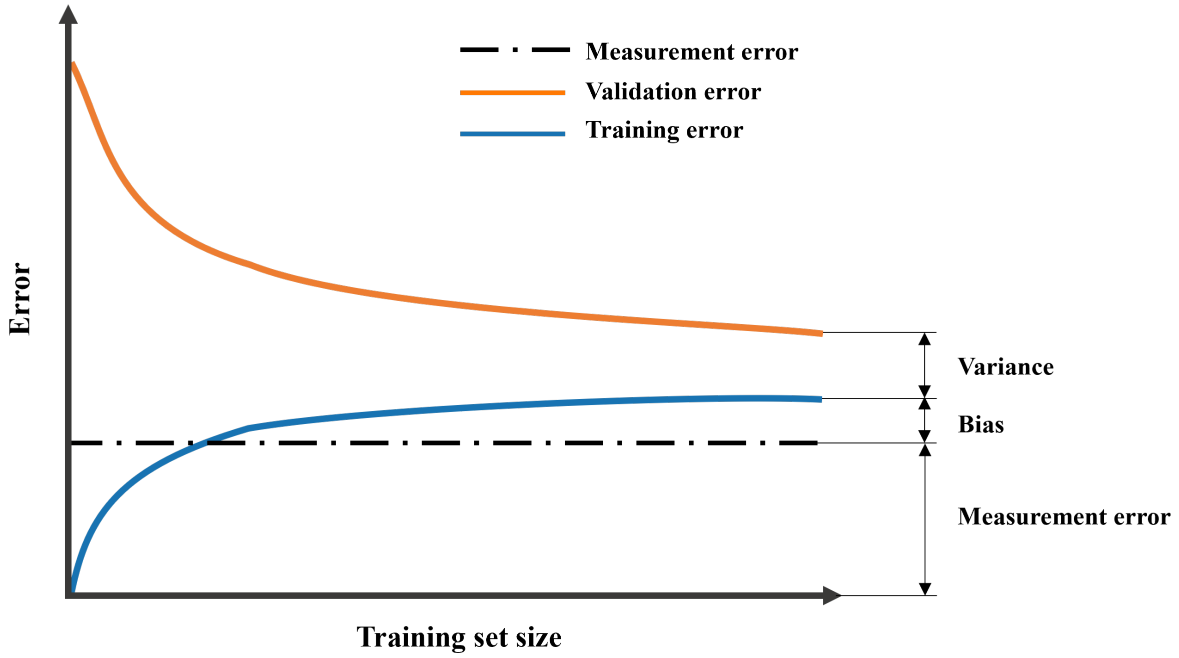


**Figure 18:** *Illustration of an ideal Learning Curve.*

In the next subsection, the method is explained and analysed with respect to the Renard Series R10 & Friedman 1 datasets.

### 4.3.1 Applying Learning Curves to Renard Series R10 & Friedman 1 dataset

The generation of the datasets and the training of the model are the same as described in the previous section. Three noise variations were investigated: no noise, only noise in $\mathbf{y}$ and noise on $\mathbf{X}$ and $\mathbf{y}$. The data sets were then divided into a 80 % training set and a 20 % test set. The Learning Curve function systematically varies the number of training samples used to fit the model and calculates the training and validation errors for each subset. The root mean squared error (RMSE) was chosen as the performance metric and a 10-fold cross-validation was applied to ensure comprehensive and reliable error estimates. The validation error, the training error, and the standard deviation for each model are shown in Table 1. The expected measurement error ME was calculated from the applied noise in the target variable. This estimation allows for the bias and variance to be estimated. The bias and variance were calculated with the formulas (15, 16) and also shown in Table 1.

For the R10 dataset, employing a linear model results in validation and training errors of

0 when no noise is present, resulting in an ideal fit without bias or variance. When noise is added to **y**, the validation error increases to 3.44, showing the impact of noise and is the same as the applied error (Figure 19). Adding noise to both **X** and **y** further increases the validation error to 4.2 (Figure 20).

When applying a XGBoost model to the Friedman 1 dataset without noise, the validation error is 0.71, while the training error is 0.24 (Figure 21). Adding noise to **y** increases the validation error to 2.38, and the bias becomes negative (Figure 22). Adding noise to both **X** and **y** further increases the validation error (Figure 23).

| Dataset | Model | Noise | ME | Val. Er. | Train. Er. | Bias | Var. |
|---|---|---|---|---|---|---|---|
| R10 | linear | no noise | $0$ | $0$ | $0$ | $0$ | $0$ |
| R10 | linear | $\sigma_x = 0; \sigma_y = 0.15$ | $3.43$ | $3.44 \pm 0.07$ | $3.43 \pm 0.01$ | $0.03$ | $0.01$ |
| R10 | linear | $\sigma_x = \sigma_y = 0.15$ | $3.38$ | $4.2 \pm 0.12$ | $4.18 \pm 0.01$ | $0.8$ | $0.02$ |
| Fried.1 | XGBoost | no noise | $0$ | $0.71 \pm 0.05$ | $0.24 \pm 0.01$ | $0.24$ | $0.46$ |
| Fried.1 | XGBoost | $\sigma_x = 0; \sigma_y = 0.15$ | $2.17$ | $2.38 \pm 0.1$ | $1.25 \pm 0.01$ | $-0.92$ | $1.13$ |
| Fried.1 | XGBoost | $\sigma_x = \sigma_y = 0.15$ | $2.16$ | $3.1 \pm 0.1$ | $1.72 \pm 0.02$ | $-0.44$ | $1.38$ |

**Table 1:** *Results of the effect of noise on the Bias and Variance for the Renard Series R10 & Friedman 1 dataset. For a more detailed explanation, please refer to the text.*

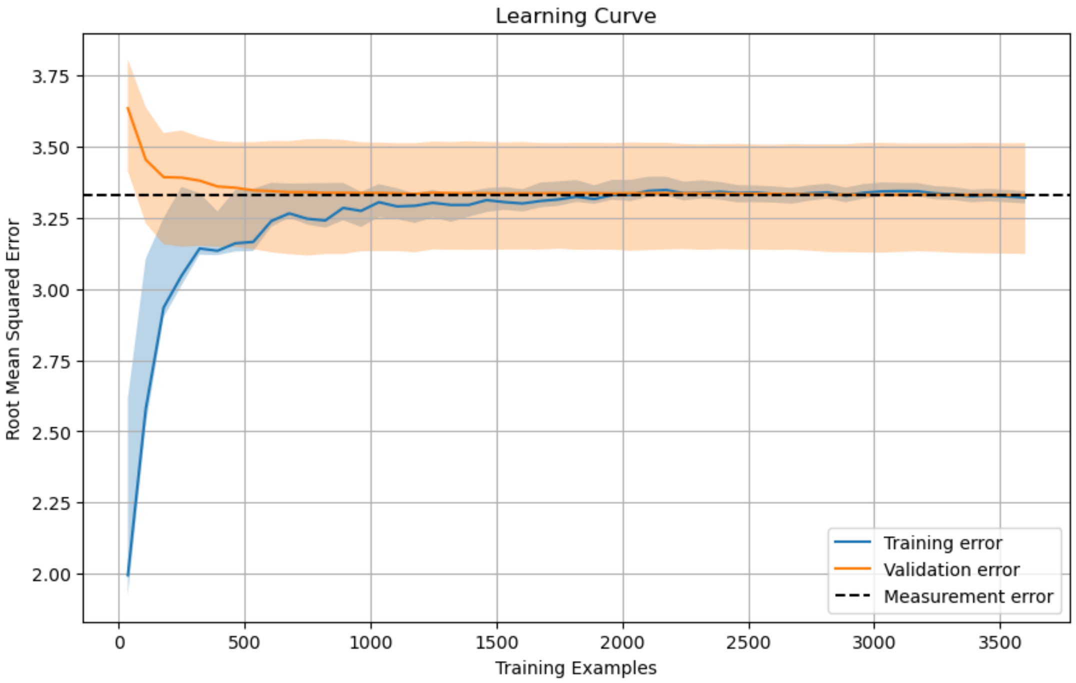


**Figure 19:** *Learning Curve for the Renard Series R10 dataset with the noise level :*$\sigma_y = 0.15$.

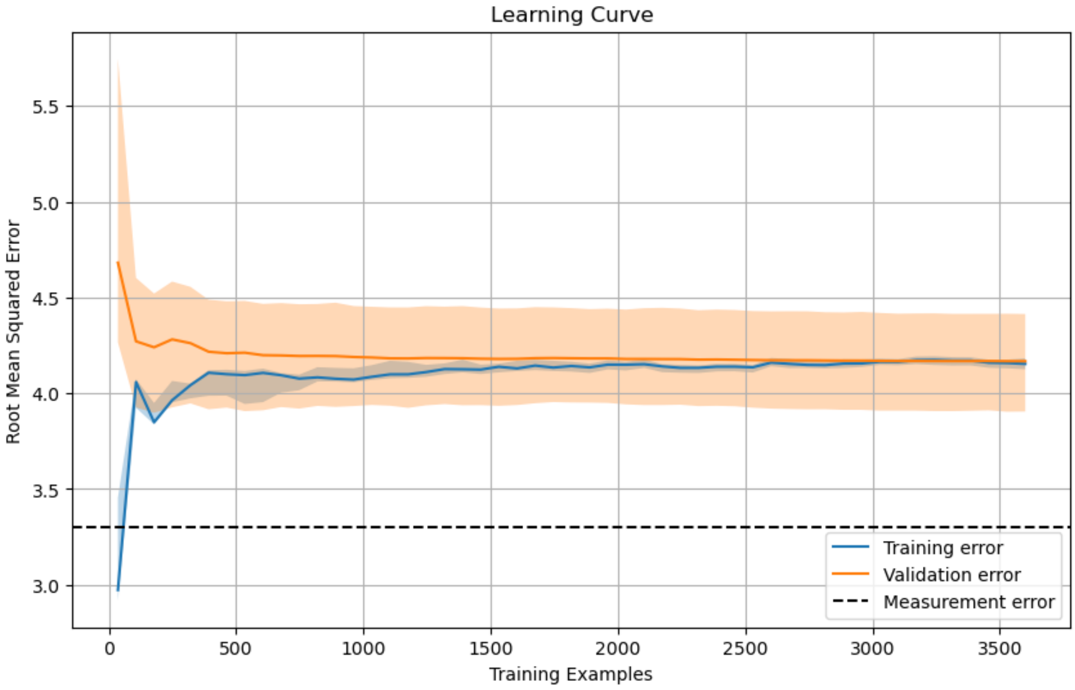


**Figure 20:** *Learning Curve for the Renard Series R10 dataset with the noise level :*$\sigma_x = \sigma_y = 0.15$.

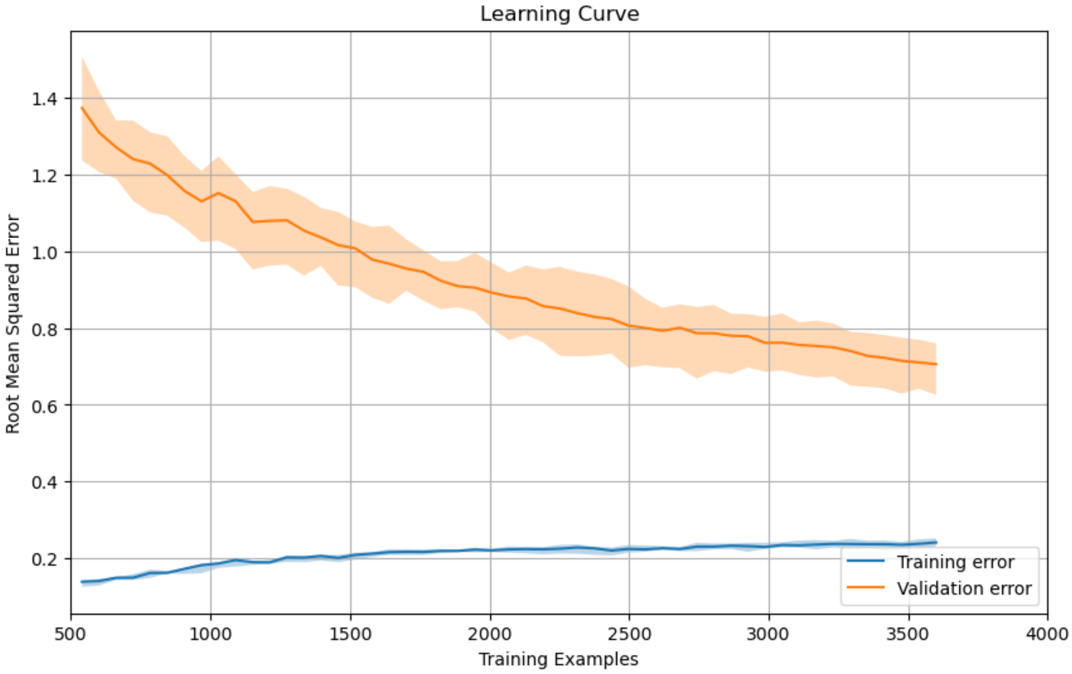


**Figure 21:** *Learning Curve for the Friedman 1 dataset with no noise.*

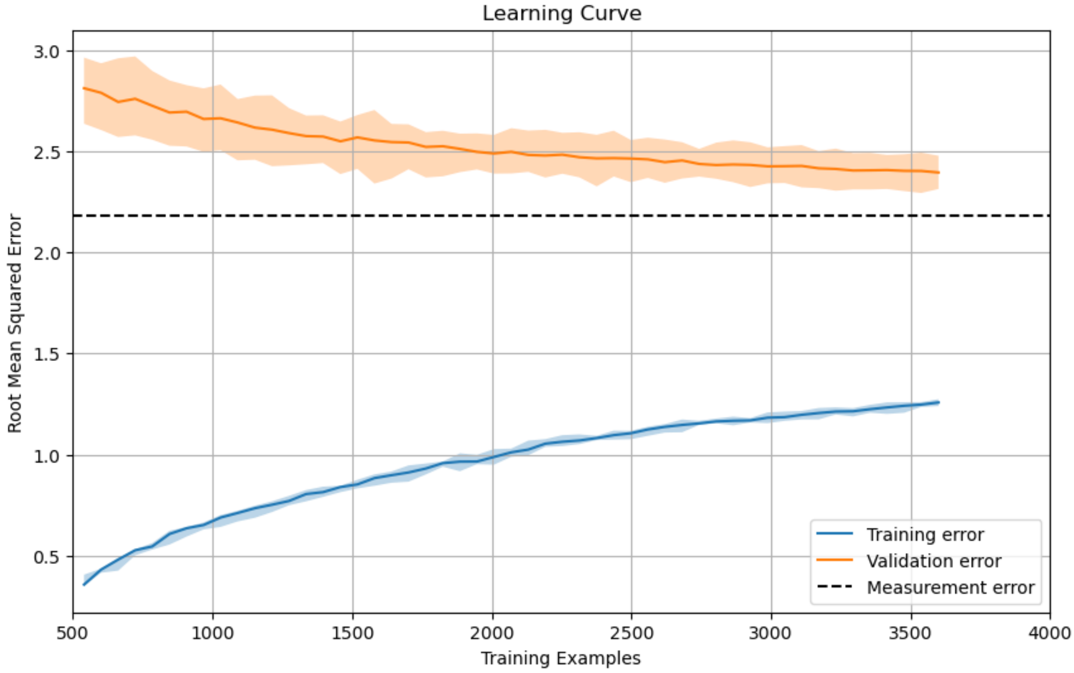


**Figure 22:** *Learning Curve for the Friedman 1 dataset with the noise level* $\sigma_y = 0.15$.

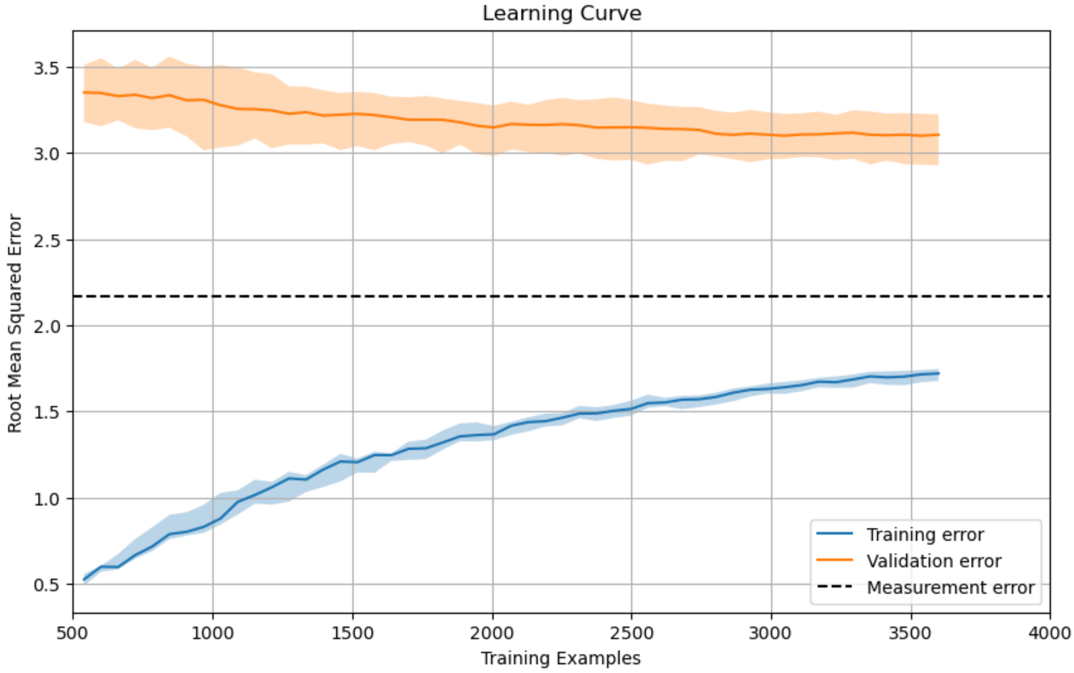


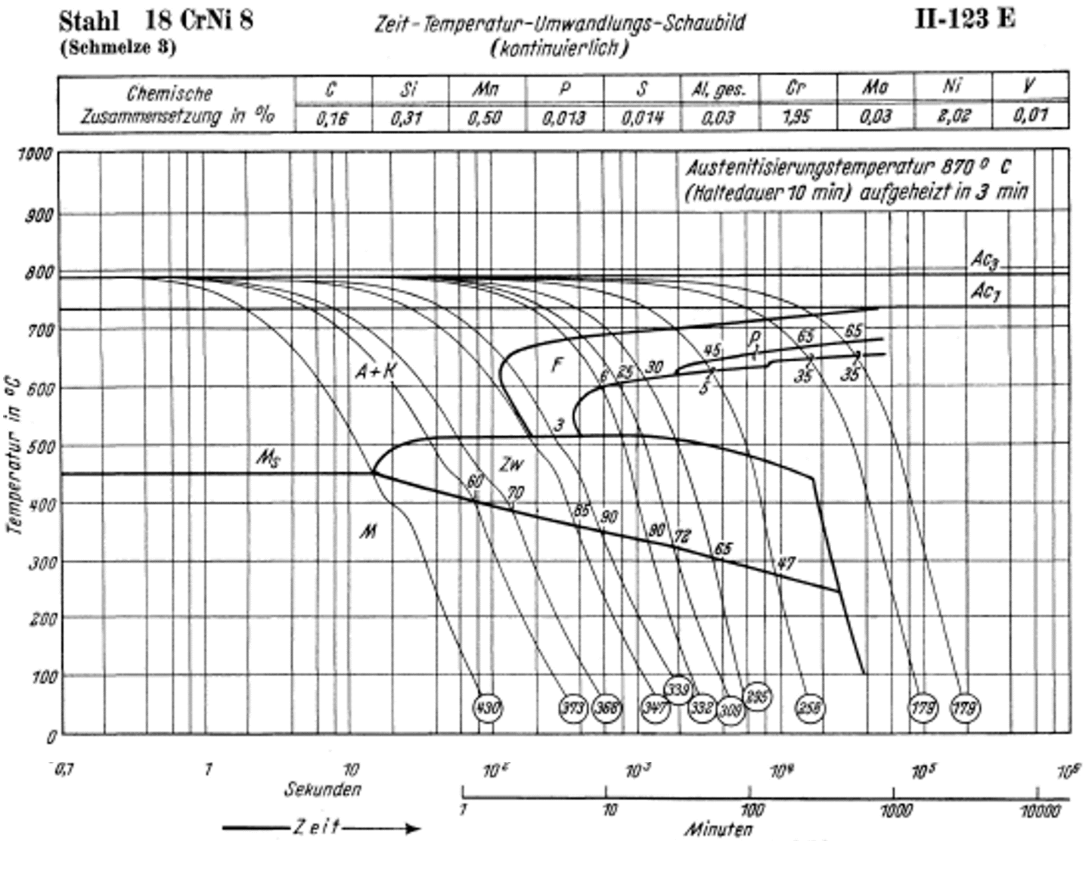
**Figure 23:** *Learning Curve for the Friedman 1 dataset with the noise level :*$\sigma_x = \sigma_y = 0.15$.

## 4.4 Discussion on the analysis of synthetic data

Measurement errors have different effects depending on whether they occur in features or in the target variable. If there is a random measurement error in the outcome variable $\mathbf{y}$, it will increase the standard error of the estimates. However, if the sample size is big enough, the ground truth will be found due to the fact that the noise averages out. However, if there is an error in the features $\mathbf{X}$, larger sample sizes will still lead to an inaccurate estimate regarding the true function. This is known as regression dilution [48]. If there is a measurement error in the features, the estimates of the regression slope coefficients will be biased toward the null [49]. Appendix A.1 provides a detailed examination of this effect based on the linear function that was utilized. This principle seems to be true not just to linear regression, but also to more complex models like XGBoost when they are used for regression tasks, like for the Friedman 1 function problem.

If the data set is incomplete or contains errors that are not normally distributed, the findings will be affected. Adding additional independent features can lead to the "kitchen sink regression" phenomenon. In the context of kitchen sink regression, the researcher includes a wide array of variables in the regression analysis with the aim of identifying a statistical pattern. This kind of regression frequently leads to inaccurately implying connections between variables and the target outcome in the dataset, potentially causing overfitting. This occurs because the higher the number of independent variables incorporated in a regression analysis, the higher the probability that one or more of them will show statistical significance, even if they do not have a causal impact on the dependent variable.[30, 50].In advanced machine learning models, this is often addressed through techniques such as regularisation. Appendix A.2 provides some extra figures for better understanding.

Missing relevant features can lead to the omitted-variable bias. The omitted variable bias occurs when a statistical model does not include one or more relevant variables [30]. An effective way to comprehend this concept is through an illustration, such as a model designed to predict the $M_s$ temperature of steels. With that model, we want to predict this for steel type 18CrNi8. The continuous cooling transformation (CCT) diagram can be seen in Figure 24 which was taken from the "Atlas zur Wärmebehandlung der Stähle" [51]. But our model does not include the cooling rate. If the cooling rate is too slow, bainite will form before the martensite transformation. This leads to carbon enrichment in the austenitic phase and a lower martensite start temperature. If we focus only on the chemical composition and ignore the cooling rate in our analysis, we might conclude that the chemical composition is the only or most important factor for martensite transformation. However, this conclusion would be biased because we have omitted the cooling rate in our study. The cooling rate is just as important as the chemical composition (at least for the low alloyed 18CrNi8). This can be seen for slow cooling times, where 60 % ferrite and 40 % pearlite and no martensite is formed. To avoid this kind of bias, it is crucial to identify and include all significant factors that could influence the outcome of interest or to specify the model application.



| Chemische Zusammensetzung in % | C | Si | Mn | P | S | Al, ges. | Cr | Mo | Ni | V |
|---|---|---|---|---|---|---|---|---|---|---|
| | 0,16 | 0,31 | 0,50 | 0,013 | 0,014 | 0,03 | 1,95 | 0,03 | 2,02 | 0,01 |

**Figure 24:** *CTT diagram for steel grade 18CrNi8, the chemical composition can be seen at the top austenitisation temperature of 870 °C and a heating time of 3 min and a holding time of 10 min [51].*

### 4.4.1 Learning Curves

The ordinary least squares regression model, when trained on the Renard Series R10 dataset has zero bias and variance (Table 1), when there is any error present in the dataset. This idealised scenario results in a perfect fit to the data. Conversely, training the XGBoost model on the nonlinear Friedman 1 dataset introduces a different challenge: epistemic uncertainty. This type of uncertainty comes not from the randomness inherent in the data, but from the model's own limitations: its inability to perfectly capture the underlying data distribution. The model's performance flaw underscores the inherent complexity of non-linear data and the limitations of predictive modeling.

The Learning Curves offer a graphical representation of how the model's performance improves as more training data is provided. It also helps to identify whether the model is affected by bias or variance, which is helpful to set the next crucial steps in the machine learning project.

### 4.4.2 Preliminary summary

- Measurement errors in features versus target variables affect models differently.
- Noise in the target variable primarily increases the standard error, whereas noise in features leads to biased estimates regarding to the ground truth function, a phenomenon known as regression dilution.
- Large sample sizes help mitigate noise in target variables but not in features, emphasising the need for noise-free high-quality data in machine learning applications.
- Learning Curves in combination with a known measurement error are a useful tool for estimate bias and variance in model performance.

# 5 Investigation of real world datasets

This section uses the Learning Curve on two material science related datasets to evaluate bias and variance of some machine learning models. It also compares neural network models to predict tensile test results to round-robin test results.

## 5.1 Martensite Start Temperature from steels

This data set was provided by Lu et al. [52], where most of the data from CTT diagrams from various sources was taken [13, 51, 53–58]. The data were cleaned and a Principal Component Analysis (PCA) performed to select the relevant features. In the end, 15 features, including the chemical composition and austenitisation temperature for 1142 steels were included in the dataset. When the element content was not reported, the corresponding values were set to zero. A summary statistic can be seen in Table 2.

| | Minimum | Maximum | Mean | Standard deviation |
|---|---|---|---|---|
| TAUST (K) | 1023 | 1673 | 1257 | 177.64 |
| C (wt%) | 0.01 | 2.25 | 0.37 | 0.29 |
| Si (wt%) | 0.00 | 3.80 | 0.33 | 0.36 |
| Mn (wt%) | 0.00 | 3.50 | 0.80 | 0.47 |
| Ni (wt%) | 0.00 | 10.00 | 0.84 | 1.46 |
| Cr (wt%) | 0.00 | 14.55 | 1.24 | 2.37 |
| Mo (wt%) | 0.00 | 5.75 | 0.26 | 0.54 |
| V (wt%) | 0.00 | 5.05 | 0.11 | 0.39 |
| Cu (wt%) | 0.00 | 1.49 | 0.05 | 0.12 |
| W (wt%) | 0.00 | 19.20 | 0.47 | 2.42 |
| Al (wt%) | 0.00 | 1.26 | 0.013 | 0.090 |
| Ti (wt%) | 0.00 | 0.20 | 0.002 | 0.014 |
| Nb (wt%) | 0.00 | 0.17 | 0.002 | 0.011 |
| N (wt%) | 0.00 | 0.29 | 0.002 | 0.017 |
| Co (wt%) | 0.00 | 11.35 | 0.08 | 0.75 |
| $M_s$ (K) | 335 | 819 | 613.3 | 86.61 |

**Table 2:** *Descriptive statistics of the $M_s$ temperature dataset [52].*

### 5.1.1 Applying Learning Curves to $M_s$ temperature dataset

The dataset was then split into an 80 % training set and a 20 % test set.The training set is used to fit the model and for cross-validation. The test set was used to evaluate the performance on the unseen data. For three advanced machine learning regression models, a Learning Curve was created. These three models are Extreme Gradient Boosting (Fig.25) XGBoost, Random Forest Regression RF (Fig.26) and Gaussian Process Regression (GPR) with a Matérn kernel (Fig.27). The hyperparameters for the models, which were optimised using a grid search, are listed in the appendix A.3. As a reference, a simple Ordinary Least Squares Regression model was trained. The performance metric selected was the root mean squared error (RMSE), and to obtain comprehensive and dependable error estimates, a 10-fold cross-validation was conducted. Table 3 displays the validation error, training error, and standard deviation for each model. The expected measurement error was $\sigma = \pm 12\text{K}$ using the offset method according to Yang's paper [59]. This estimate makes it possible to calculate bias and variance. The values for bias and variance were determined using the equations (15, 16) and are also presented in Table 3.

| **Model** | **Val. Error [K]** | **Train. Error [K]** | **Bias** | **Variance** |
|---|---|---|---|---|
| XGBoost | $22.77 \pm 2.16$ | $6.67 \pm 0.33$ | $-5.33$ | $16.09$ |
| RF | $23.38 \pm 2.3$ | $9.08 \pm 0.1$ | $-2.92$ | $14.31$ |
| GPR | $22.89 \pm 2.84$ | $13.65 \pm 0.37$ | $1.65$ | $9.25$ |
| linear | $39.72 \pm 3.98$ | $38.49 \pm 0.46$ | $26.49$ | $1.24$ |

**Table 3:** *Comparison of the different models. For a more detailed explanation, please refer to the text.*

Figure 29 illustrates the performance of the GPR model with the Matérn kernel, showing a scatter plot of the test data against the predicted $\text{M}_s$ temperatures, together with error bars indicating the uncertainty of the prediction. The model has a $R^2$ value of 0.93 and a RMSE of 23.86 K. In Figure 30 the prediction uncertainty of the model is visualised using a histogram. The minimum standard deviation is 16.87 K and the maximum is 63.2 K. The median of the standard deviation is 18.91 K.

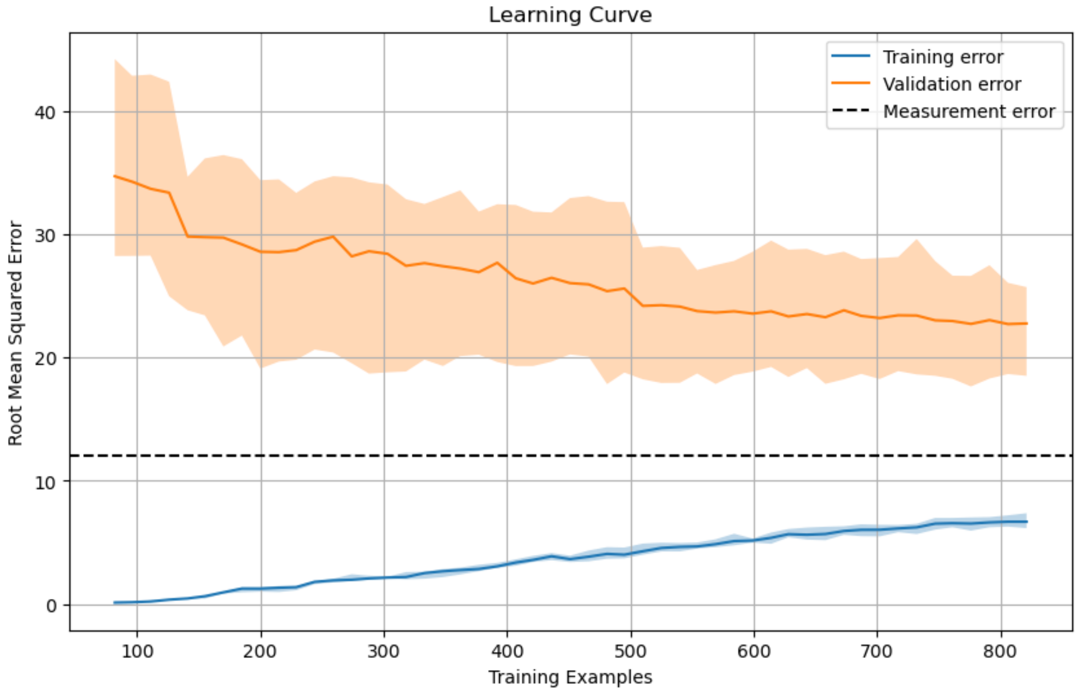


**Figure 25:** *The Learning Curve for XGBoost.*

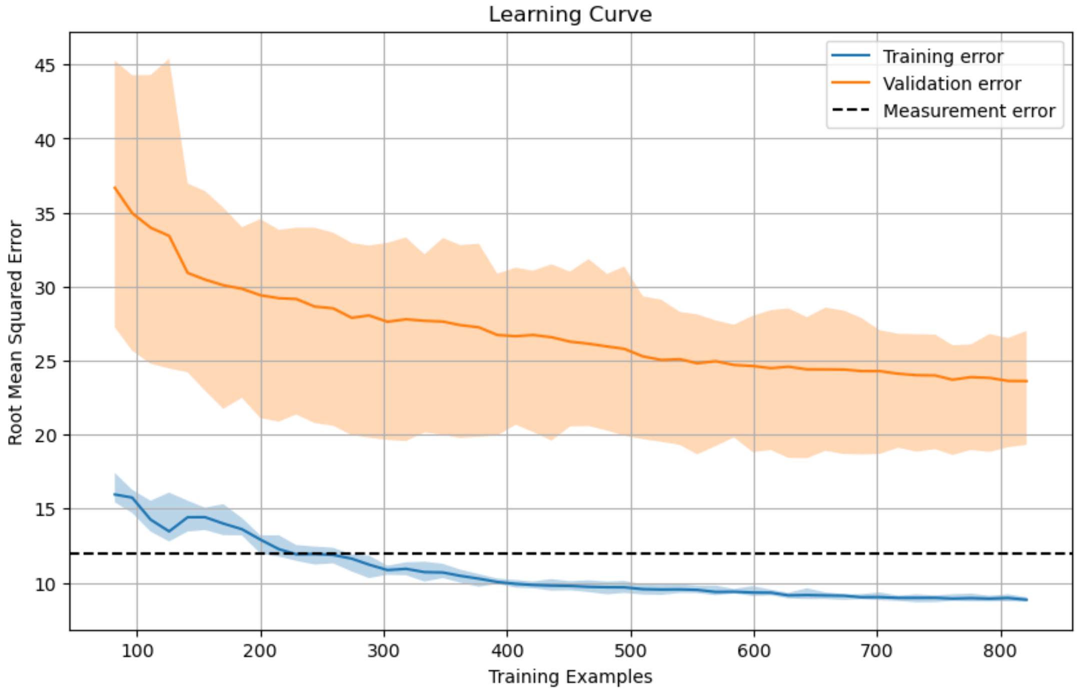


**Figure 26:** *The Learning Curve for RF.*

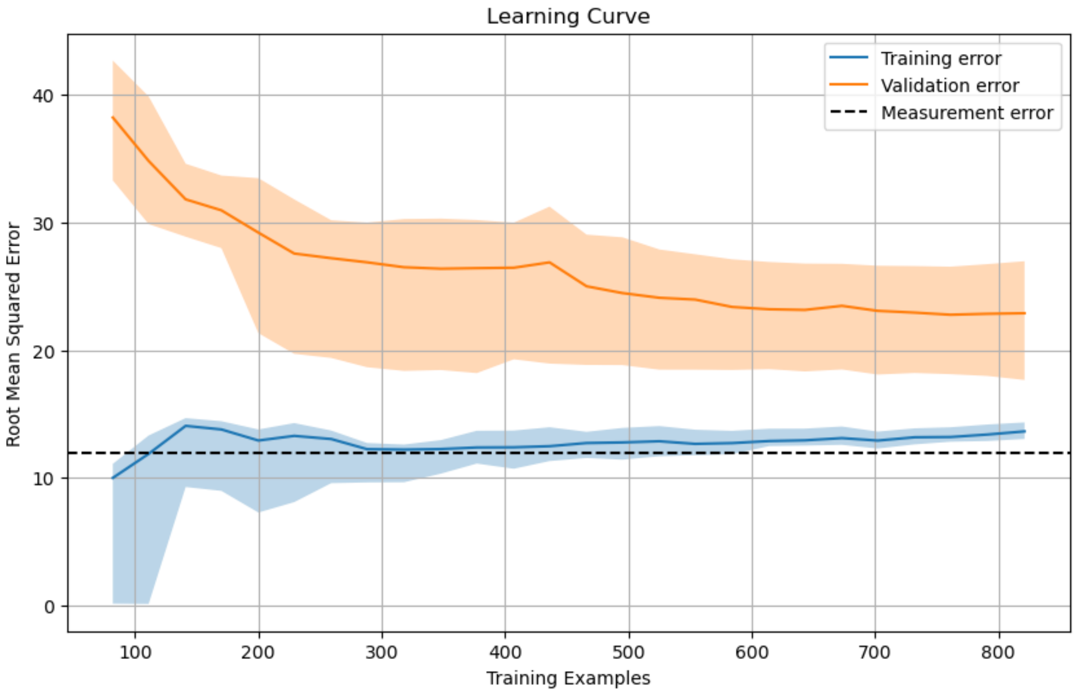


**Figure 27:** *The Learning Curve for GPR.*

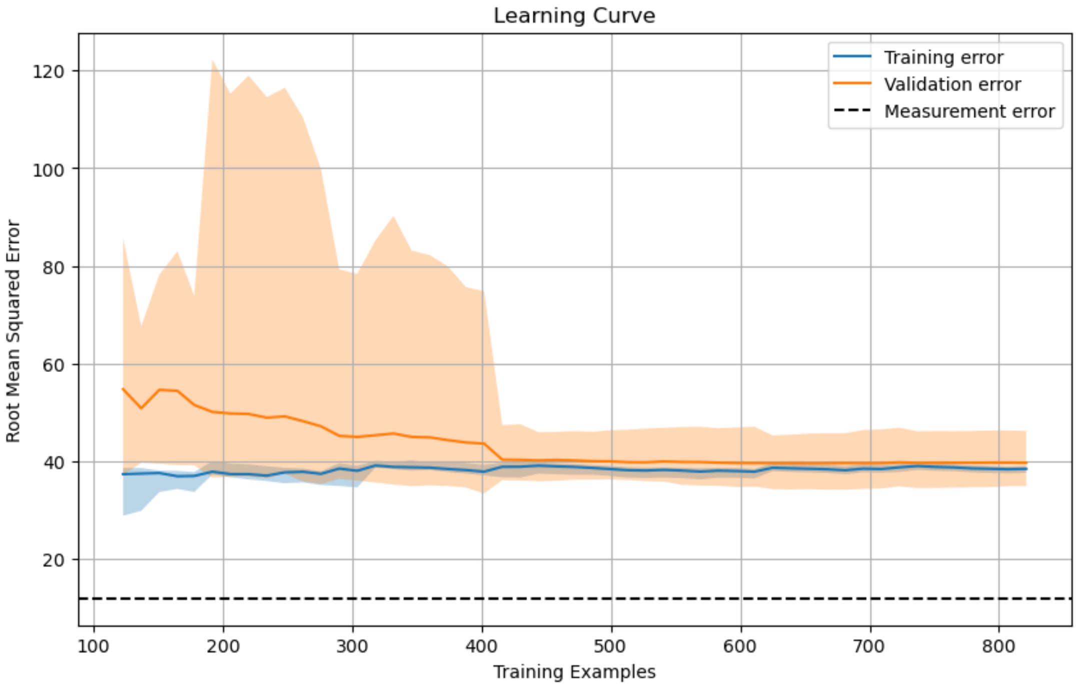


**Figure 28:** *The Learning Curve for Ordinary Least Squares Regression.*

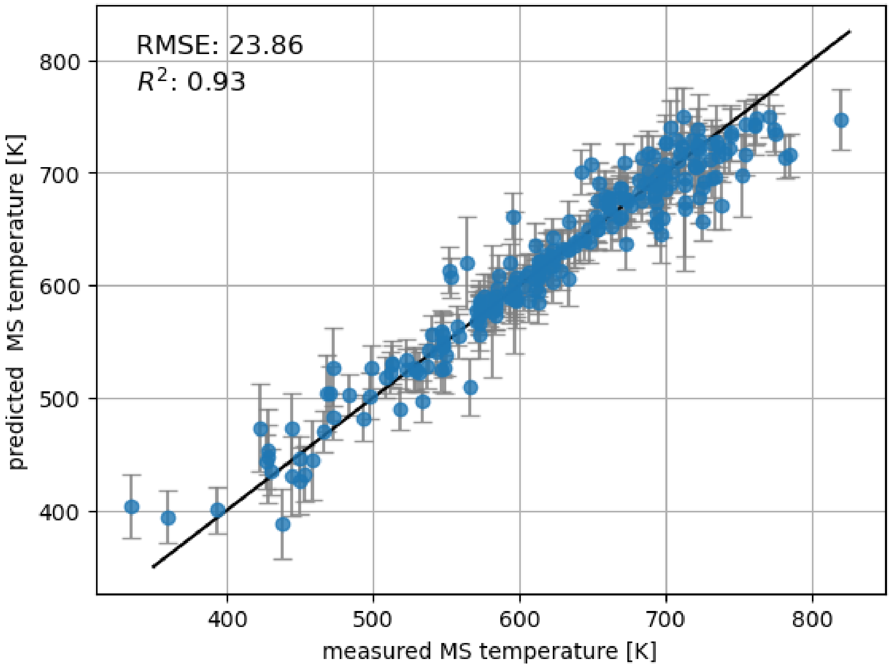


**Figure 29:** *Prediction of $M_s$ temperature using GPR model.*

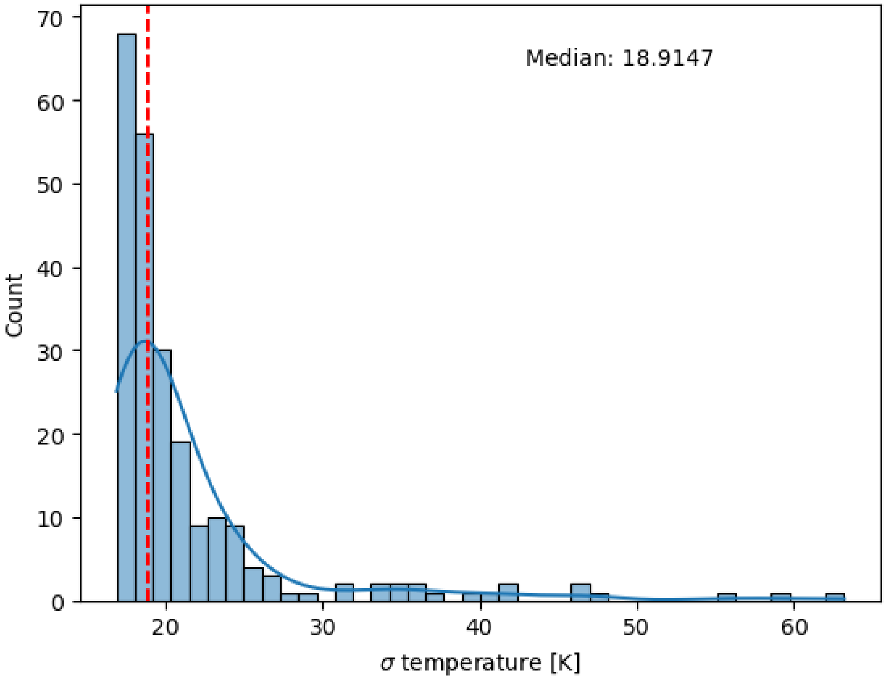


**Figure 30:** *Standard deviation of MS temperature predictions.*

### 5.1.2 Discussion on the analysis of the $M_s$ temperature dataset

When examining the validation errors of the four models presented in Table 3, it is observed that the three advanced models exhibit comparable performance metrics. In contrast, the linear model performs significantly worse than the other models. This outcome aligns with expectations, as the linear model demonstrates a pronounced bias, indicating its inability to capture complex relationships within the data. Both the RF and XGBoost models are characterized by high variance, suggesting a tendency to overfit. Similarly, the GPR model encounters issues with variance and a slight degree of bias. The XGBoost model showed high variance. Attempts to address this issue, such as increasing regularisation parameters and reducing tree depth, led to worse scores on the validation set.

The training error curve of the RF model decreases with more data (Figure 26). RF is a bagging algorithm. This means that the RF algorithm makes predictions by averaging the predictions of all individual trees. Because the spread of the chemical composition is very large in this dataset, the predictions are less accurate for the small sample size. This results in an improvement in performance with more training data, as there is more information for the model to learn from. However, if the sample size becomes larger than the actual size, the training error should increase again.

When comparing our results with those of Rahaman et al. [14], who used a database with 2277 unique entries of $M_s$ versus alloy composition, we find some differences. Their approach, employing an AdaBoost model, achieved a RMSE of 18 K. As is known, a common way to reduce variance is to add more data, this could explain why the performance of the model is better. However, it is also important to mention that their study excluded alloys containing strong carbide-forming elements such as W, V, Nb, and Ti. These elements were omitted because the carbides may not dissolve completely in the austenite matrix at austenitization temperatures. This was not done in our current database.

A key advantage of GPR is its ability to provide a confidence intervals for predictions, as illustrated in Figure 30. The median standard deviation of these predictions is 18.91 K, closely aligned with the measurement error reported of 12 K. It is important to note that the reported values for the $M_s$ temperature are derived from various sources,[13, 51, 53–58] some of which date back to the 1950s. Another benefit of the GPR model is its ability to indicate uncertainty. For example, the reported maximum standard deviation is 63.2 K for steel with a Cu concentration of 1.49 wt%. In Figure 31, a histogram shows the distribution of Cu concentrations. For most steels, the Cu concentration was assumed to be zero, indicating that the GPR model has limited information on how Cu affects $M_s$ temperature, resulting in high uncertainty in prediction.

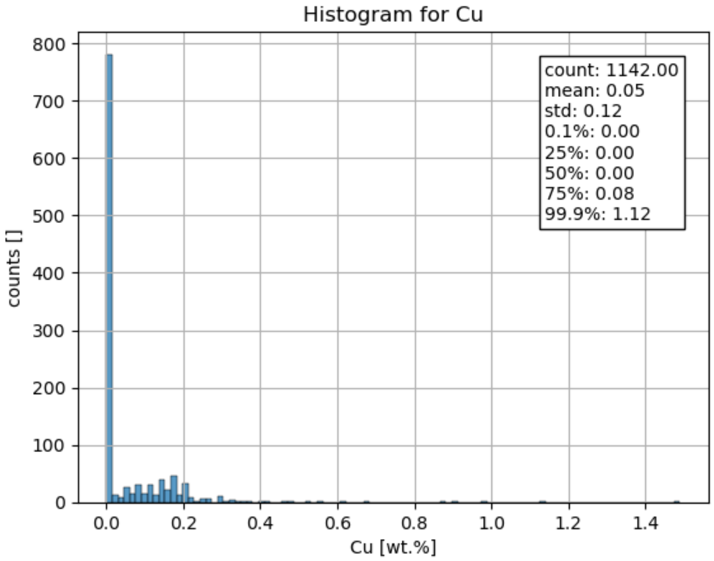


**Figure 31:** *This histogram shows the copper concentration in wt.% for the MS temperature dataset.*

## 5.2 Investigation of the r-value (plastic strain ratio) dataset

This data set was provided by Millner et al.[9]. It contains 14430 observations of the r-value (plastic strain ratio) and 594 features of a cold-rolled batch-annealed low carbon steel coil. The r-value is a measure of the resistance of a sheet to thinning during deep drawing. The r-value was taken from a tensile test and calculated according to ISO 10113 using an 80 mm sample at 18 % elongation. The properties are taken from an industrial steel production route that includes chemistry, cutting, hot rolling, pickling, cold rolling, annealing, skin-pass rolling, and sampling.

### 5.2.1 Using the theoretically measurement error in the Learning Curve

The expected measurement error for the measurement of the r value was estimated using the propagation of uncertainty from the ASTM E 517-00 Appendix X1 [60] for the median value of 2.1. This leads to an expected standard deviation of 0.09 (calculation can be found in Appendix A.4). With this information, we can now apply the same analysis we performed for the $M_s$ temperature dataset. In Figure 32 the Learning Curve of a XGBoost model with the optimised hyperparameter which where reported from Millner et al.[9] can be seen. The bias and variance were again calculated with the equations (15, 16) and can be seen in Table 4.

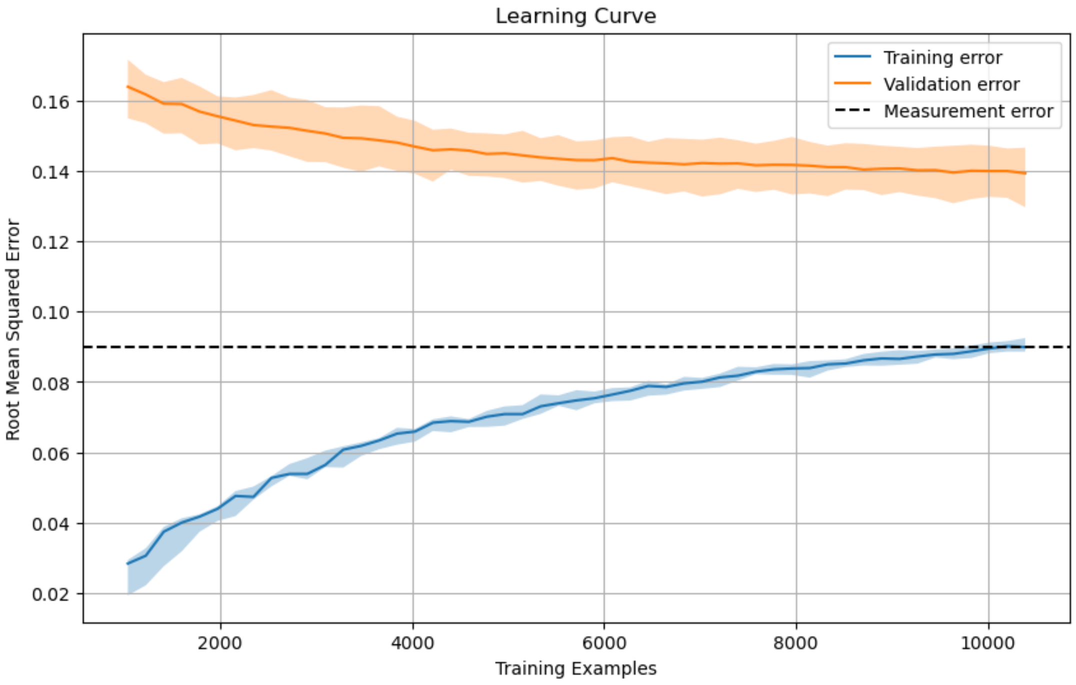


**Figure 32:** *The XGBoost Learning Curve for predicting r-value.*

| Model | Val. Error | Train. Error | Bias | Variance |
|---|---|---|---|---|
| XGBoost | $0.14 \pm 0.01$ | $0.09 \pm 0.00$ | 0.0 | 0.05 |

**Table 4:** *Validation and Training Error, Bias, and Variance for the XGBoost Model. For a more detailed explanation, please refer to the text.*

### 5.2.2 Discussion on the analysis of the r-value dataset

The XGBoost model for the r-value data set shows variance. Common ways to deal with this are to add more training data, add regularisation, reduce the number / type of input features, or reduce the model size [47]. However, 14430 observations are already a very large data set in the context of materials science. L2 regularisation is already applied in the optimised hyperparameter. With the current 594 features, feature selection could be beneficial.

An important consideration is that we have no information about the error in the **X** features. If we look at the Learning Curve of the Friedman 1 data set23), where noise has been applied to **X** and **y**, we can see that there is a greater deviation from validation error to the measurement error. Another very important point is that the error also depends on the r-value as well. This calculated standard deviation as a function of the r-value can be seen in Figure 33. Due to these effects, it is very difficult to determine an optimum error rate, and therefore it should not be regarded as a hard limit but as a benchmark.

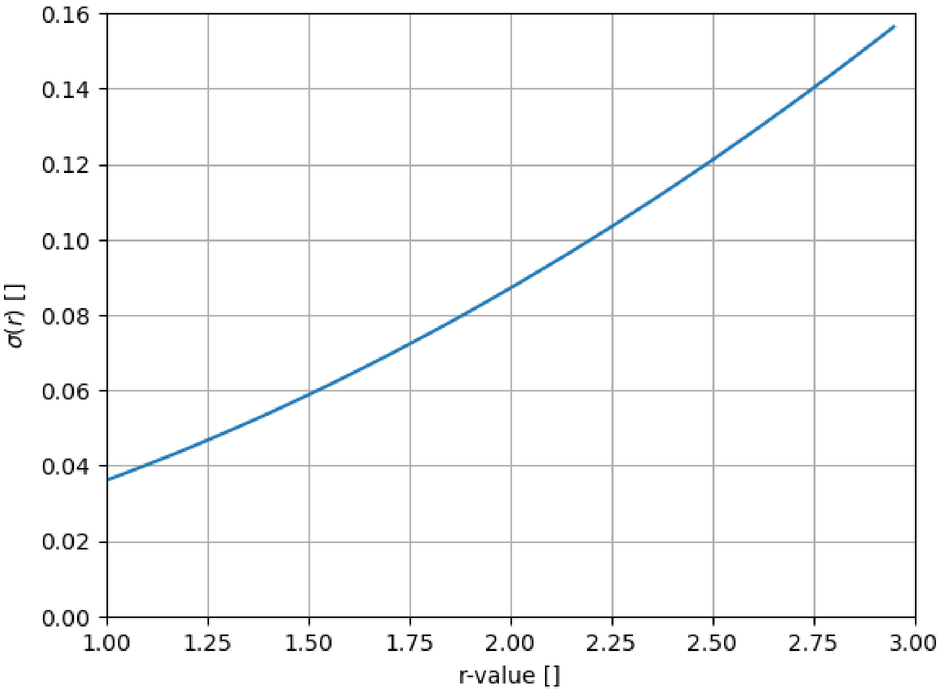


**Figure 33:** *Calculated standard deviation as a function of r-value for a relative length and width measurement error of 0.001 at an axial strain of 0.18.*

## 5.3 Comparison of the performance of neural networks from the literature in the production of steel for the prediction of material properties from tensile tests

Tensile testing is a key experiment in materials science and engineering, providing critical insight into the mechanical properties of materials, such as their strength, ductility, and elasticity. This importance is reflected in the development of many neural networks aimed at predicting tensile properties directly from production data. In Table 6 is a summary of the performance of different machine learning models from the literature for different types of steel. This table includes the yield strength (YS), the ultimate tensile strength (UTS), the elongation at the fracture (A) and their corresponding RMSE and $R^2$ from the predictions of the model versus the test data.

Despite technological advances, no test is completely free of errors. Factors such as machine calibration, specimen preparation and test conditions can all influence the results. Table 5 summarises the results of an interlaboratory comparison of the results of tensile tests for different types of steel. It includes yield strength (YS), ultimate tensile strength (UTS), and elongation at fracture (A). The 95% confidence interval has been reported in ISO 6892-1 Annex K [61]. The standard deviation was calculated from the confidence interval reported. The interlaboratory scatter provides a baseline of variability. If the predictive model has a RMSE close to the measurement error, it would suggest that the machine learning model's predictions are within the natural variability range of different laboratory measurements, which is an indication of good model performance, and therefore all variables required for the prediction are included in the model.

For example, let us compare the DX56 (deep-drawing steel) with the model from Lalam et al. [4] that describes a commercial quality & drawing quality steel. For the yield strength (YS), the round robin test shows a standard deviation of 3.8 MPa, while the model has a higher RMSE of 6 MPa. This suggests that the model predictions for YS are less accurate than the variability seen in laboratory tests. The ultimate tensile strength (UTS) shows a closer alignment between the round robin test standard deviation of 7.7 MPa and the model's RMSE of 5.52 MPa, indicating that the model is quite effective in predicting UTS within the range of natural variability observed in the tests. This comparison is useful in understanding how well the predictive model performs relative to the natural variability observed in material testing.

Another important point is that when evaluating and comparing the performance of different prediction models, relying only on $R^2$, can be misleading [62]. Although a high $R^2$ indicates a good fit of the model to the data, it does not provide any information on the absolute size of the errors. To gain a complete understanding, it is important to consider additional metrics, such as RMSE. A good example is the work of Tamminen et al. [7] (Figure 34). An excellent value of $R^2$ of 0.99 was reported for the prediction of the ultimate tensile strength. However, the RMSE for the ultimate tensile strength is 13.3 MPa, which may not be expected when considering the value of $R^2$ alone.

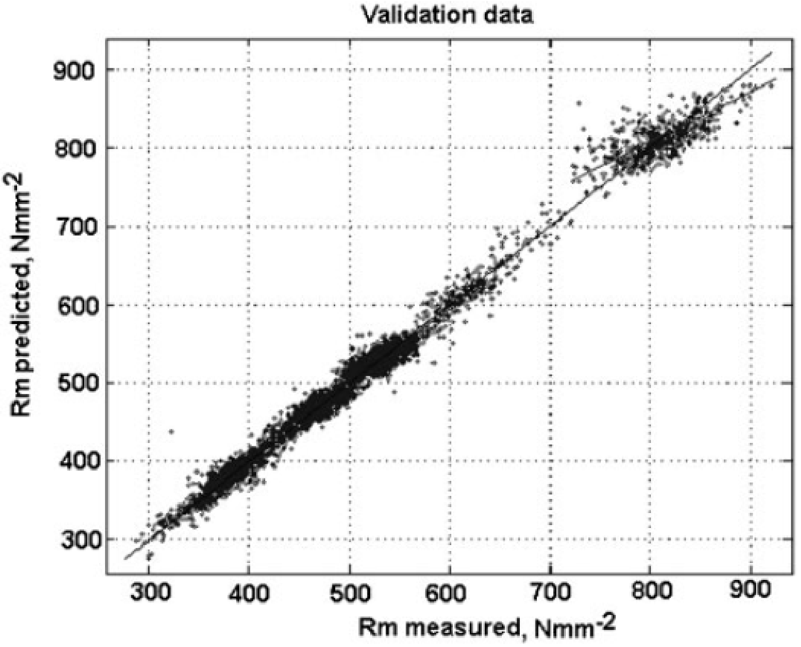


**Figure 34:** *Graphical representation of ultimate tensile strength predictions by Tamminen et al. [7].*

| Code | YS [MPa] | ± [%] | $\sigma$ [MPa] | UTS [MPa] | ± [%] | $\sigma$ [MPa] | A [%] | ± [%] | $\sigma$ [%] |
|---|---|---|---|---|---|---|---|---|---|
| DX56 (deep-drawing steel) | 162 | 4.6 | 3.8 | 301.1 | 5 | 7.7 | 45.2 | 12.4 | 2.9 |
| HR3 (hot-roiled carbon steel sheet) | 228.6 | 8.2 | 9.6 | 335.2 | 5 | 8.6 | 38.4 | 13.8 | 2.7 |
| ZStE 180 (bake hardening steel) | 267.1 | 9.9 | 13.5 | 315.3 | 4.2 | 6.8 | 40.5 | 12.7 | 2.6 |
| P245GH | 367.4 | 5 | 9.4 | 552.4 | 2 | 5.6 | 31.4 | 14 | 2.2 |
| C22 | 402.4 | 4.9 | 10.1 | 596.9 | 2.8 | 8.5 | 25.6 | 10.1 | 1.3 |
| S355 | 427.6 | 6.1 | 13.3 | 564.9 | 2.4 | 6.9 | 28.5 | 17.7 | 2.6 |
| 30NiCrMo16 | 1 039.90 | 2 | 10.6 | 1167.8 | 1.5 | 8.9 | 16.7 | 13.3 | 1.1 |

**Table 5:** *Reproducibility from laboratory intercomparison exercises [61]*

| Steel type | YS RMSE [MPa] | YS $R^2$ | UTS RMSE [MPa] | UTS $R^2$ | A RMSE [%] | A $R^2$ | Total Datasize | Ref. |
|---|---|---|---|---|---|---|---|---|
| aluminium killed steel | 21.4 | — | 11.05 | — | 2.37 | — | 4196 | [2] |
| bake hardening steel | 10.42 | — | 8.58 | — | 2.03 | — | 6584 | [2] |
| dual phase steel | 14.48 | — | 18.56 | — | 1.67 | — | 3538 | [2] |
| high-strength low-alloy | 15.55 | — | 11.39 | — | 2.41 | — | 14 332 | [2] |
| interstitial-free steel | 7.06 | — | 5.99 | — | 1.81 | — | 20 362 | [2] |
| rephosphorized steel | 11.1 | — | 8.11 | — | 1.94 | — | 2495 | [2] |
| hot rolled steel (S355,Q345,AH36,X80,12Mn,Q550) | 21.9 | 0.92 | 16.73 | 0.93 | 2.37 | 0.94 | 11 101 | [3] |
| interstitial-free steel | 6.001 | — | 5.36 | — | — | — | — | [4] |
| commercial quality & drawing quality | 6.009 | — | 5.52 | — | — | — | — | [4] |
| tempered steel | — | — | 34.6 | — | — | — | — | [5] |
| non-micro-alloyed thermo-mechanical steel | — | 0.9321 | — | 0.9799 | — | 0.8569 | 35 000 | [6] |
| micro-alloyed thermo-mechanical steel | — | 0.9479 | — | 0.9667 | — | 0.8479 | 35 000 | [6] |
| hot-rolled C–Mn and micro-alloyed steel strip | 18 | 0.99 | 13.3 | 0.99 | 1.8 | 0.94 | 56 436 | [7] |
| hot-rolled C–Mn and micro-alloyed steel strip $>$ 720 MPa | 27.7 | 0.75 | 21.5 | 0.62 | 1.2 | 0.55 | — | [7] |
| hot-rolled IF steel | — | 0.97 | — | 0.985 | — | 0.967 | 1557 | [8] |
| mild steel grade | — | 0.77 | — | 0.85 | — | 0.71 | 6986 | [9] |
| low carbon strip steel UTS $>$ 320 MPa | — | 0.76 | — | 0.76 | — | 0.51 | 1209 | [10] |
| carbon & HLSA steel | 20.2 | — | 12.5 | — | 3.2 | — | 12 197 | [11] |

**Table 6:** *Summary of Machine Learning Models from the literature and their performance metrics*

# 6 Conclusion & Future Work

The discussions prior to this section have already addressed the investigation of both synthetic and real-world datasets. The success of a machine learning project is highly dependent on the difficulty of the problem, the cost of the data, and the quality of the performance of the needed model. If high accuracy is required, the precision of the measurement becomes decisive. The bias-variance decomposition also highlights this issue, with an irreducible noise term related to measurement error, further emphasizing the importance of verifying the measurement error in advance. For a successful machine learning model,**high-quality data is the key factor** [25, 63].

The thesis investigated the influence of measurement uncertainties on the prediction accuracy of machine learning. Through the analysis of both synthetic and real-world datasets, it was shown that measurement uncertainties significantly affect model performance. In particular, errors in features **X** were found to introduce bias in predictions, while errors in target variable **y** tended to average out over large datasets. It was shown that the measurement error can be used as an optimal error which offers a heuristic method to evaluate a model's variance and bias. The thesis underscores the importance of selecting appropriate evaluation metrics, such as the root mean square error (RMSE), which aligns more closely with the nature of measurement errors than the coefficient of determination ($R^2$).

If a high level of precision is necessary for predictions made with the ML model, it may be constrained by the measurement error. Therefore, it is helpful to verify the measurement error in advance. For the $M_s$ dataset, it was found that the validation error of the ML models was about 20-23 K which can be compared to the measurement error of about 12 K. For the r-value dataset, the XGBoost model showed an error of about 0.14 while the measurement error is about 0.09. For completeness, the thesis also provides a review of applications of ML to predict mechanical properties. The review revealed that the best ML models provide validation errors of the same size as the measurement error. Models applied to more general steel classes were less performant and revealed validation errors of about twice the measurement error. In general, the validation error is close to the optimal error which indicates that the ML models are quite efficiently extracting the information present in the data.

Future research could focus on exploring Bayesian neural networks (BNNs) as an alternative to Gaussian process regression (GPR) in material science. Bayesian models, including GPR and BNNs, offer compelling advantages due to their ability to quantify uncertainty, which is critical when addressing measurement errors. While the more complex architecture and longer training times of BNNs have historically limited their use, recent advances in computational power and the development of improved software libraries have significantly improved their usability [64]. For example, Yang et al. demonstrated the effectiveness of BNNs in predicting the ultimate tensile strength of hot-rolled steel products, achieving an RMSE of approximately 17 MPa [65]. This work highlights the potential of BNNs to provide high prediction accuracy and also accurate uncertainty quantification which may be also used for smart data acquisition strategies such as e.g. active learning concepts.

# Bibliography

[1] Andriy Burkov. *Machine learning engineering*. Vol. 1. True Positive Incorporated Montreal, QC, Canada, 2020. Chap. 1, pp. 1–21.

[2] Iskender Kayabasi Adil Han Orta and MunireGulay Senol. "Prediction of mechanical properties of cold rolled and continuous annealed steel grades via analytical model integrated neural networks". In: *Ironmaking & Steelmaking* 47.6 (2020), pp. 596–605. DOI: 10.1080/03019233.2019.1568000.

[3] Qian Xie et al. "Online prediction of mechanical properties of hot rolled steel plate using machine learning". In: *Materials Design* 197 (2021), p. 109201. ISSN: 0264-1275. DOI: https://doi.org/10.1016/j.matdes.2020.109201. URL: https://www.sciencedirect.com/science/article/pii/S026412752030736X.

[4] Sibasis Sahoo Satyanarayana Lalam Prabhat Kr Tiwari and Achinta Kr Dalal. "Online prediction and monitoring of mechanical properties of industrial galvanised steel coils using neural networks". In: *Ironmaking & Steelmaking* 46.1 (2019), pp. 89–96. DOI: 10.1080/03019233.2017.1342424.

[5] P. F. Morris J. Tenner D. A. Linkens and T. J. Bailey. "Prediction of mechanical properties in steel heat treatment process using neural networks". In: *Ironmaking & Steelmaking* 28.1 (2001), pp. 15–22. DOI: 10.1179/irs.2001.28.1.15.

[6] Itishree Mohanty et al. "Prediction of properties over the length of the coil during thermo-mechanical processing using DNN". In: *Ironmaking & Steelmaking* 48.8 (2021), pp. 953–961. DOI: 10.1080/03019233.2020.1848303.

[7] P. Tamminen et al. "System for on and offline prediction of mechanical properties and microstructural evolution in hot rolled steel strip". In: *Ironmaking & Steelmaking* 34.2 (2007), pp. 157–165. DOI: 10.1179/174328107X165780.

[8] I. Mohanty et al. "Online mechanical property prediction system for hot rolled IF steel". In: *Ironmaking & Steelmaking* 41.8 (2014), pp. 618–627. DOI: 10.1179/1743281214Y.0000000178.

[9] Gerfried Millner et al. "Machine learning mechanical properties of steel sheets from an industrial production route". In: *Materialia* 30 (2023), p. 101810. ISSN: 2589-1529. DOI: https://doi.org/10.1016/j.mtla.2023.101810. URL: https://www.sciencedirect.com/science/article/pii/S2589152923001370.

[10] Ananya Mukhopadhyay and Asif Iqbal. “Prediction of Mechanical Properties of Hot Rolled, Low-Carbon Steel Strips Using Artificial Neural Network”. In: *Materials and Manufacturing Processes* 20.5 (2005), pp. 793–812. DOI: 10.1081/AMP-200055140.

[11] C. Dumortier et al. “Statistical Modelling of Mechanical Properties of Microalloyed Steels by Application of Artificial Neural Networks”. In: *Microalloying in Steels*. Vol. 284. Materials Science Forum. Trans Tech Publications Ltd, June 1998, pp. 393–402. DOI: 10.4028/www.scientific.net/MSF.284-286.393.

[12] C. Capdevila, F. G. Caballero, and C. García de Andrés. “Determination of Ms Temperature in Steels: A Bayesian Neural Network Model”. In: *ISIJ International* 42.8 (2002), pp. 894–902. DOI: 10.2355/isijinternational.42.894.

[13] S.M.C. van Bohemen and L. Morsdorf. “Predicting the Ms temperature of steels with a thermodynamic based model including the effect of the prior austenite grain size”. In: *Acta Materialia* 125 (2017), pp. 401–415. ISSN: 1359-6454. DOI: https://doi.org/10.1016/j.actamat.2016.12.029. URL: https://www.sciencedirect.com/science/article/pii/S135964541630965X.

[14] Moshiour Rahaman et al. “Machine learning to predict the martensite start temperature in steels”. In: *Metallurgical and Materials Transactions A* 50 (2019), pp. 2081–2091.

[15] *Metallic materials - Tensile testing - Part 1: Method of test at room temperature*. Annex J. ISO 6892-1:2009. International Organization for Standardization. 2009.

[16] HKDH Bhadeshia. “Neural networks and information in materials science”. In: *Statistical Analysis and Data Mining: The ASA Data Science Journal* 1.5 (2009), pp. 296–305.

[17] Esther Heid et al. “Characterizing Uncertainty in Machine Learning for Chemistry”. In: *Journal of Chemical Information and Modeling* 63.13 (2023), pp. 4012–4029. DOI: 10.1021/acs.jcim.3c00373. URL: https://doi.org/10.1021/acs.jcim.3c00373.

[18] Park M. Reilly and Hugo Patino-Leal. “A Bayesian Study of the Error-in-Variables Model”. In: *Technometrics* 23.3 (1981), pp. 221–231. ISSN: 00401706. (Visited on 06/03/2024).

[19] Yarin Gal. “Uncertainty in Deep Learning”. PhD thesis. University of Cambridge, 2016.

[20] IEC BIPM et al. *International Vocabulary of Metrology–Basic and general concepts and associated terms (VIM), 3rd edn, JCGM 200: 2008*. 2008.

[21] Barry N Taylor, Chris E Kuyatt, et al. *Guidelines for evaluating and expressing the uncertainty of NIST measurement results, Appendices D*. Vol. 1297. US Department of Commerce, Technology Administration, National Institute of ..., 1994.

[22] Amir Momeni-Boroujeni and Matthew R. Pincus. “Systematic Error Detection in Laboratory Medicine”. In: *Quality Control in Laboratory*. Ed. by Gaffar Sarwar Zaman. Rijeka: IntechOpen, 2018. Chap. 5. DOI: 10.5772/intechopen.72311. URL: https://doi.org/10.5772/intechopen.72311.

[23] Informal Working Group on Measurement Uncertainties. *Proposal for a Document for Reference: A General Approach to Handling Measurement Uncertainty*. Tech. rep. Submitted by the Informal Working Group on Measurement Uncertainties. Working Party on Noise and Tyres (GRBP), Economic Commission for Europe (ECE), 2022.

[24] Armen Der Kiureghian and Ove Ditlevsen. “Aleatory or epistemic? Does it matter?” In: *Structural Safety* 31.2 (2009). Risk Acceptance and Risk Communication, pp. 105–112. ISSN: 0167-4730. DOI: https://doi.org/10.1016/j.strusafe.2008.06.020. URL: https://www.sciencedirect.com/science/article/pii/S0167473008000556.

[25] Cornelia Gruber et al. *Sources of Uncertainty in Machine Learning – A Statisticians’ View*. 2023. arXiv: 2305.16703 [stat.ML].

[26] Bleck Wolfgang. *Werkstoffkunde Stahl: für Studium und Praxis*. Verlag Mainz, Wissenschaftsverlag, Aachen, 2010. Chap. 5, pp. 125–181. ISBN: 3-89653-820-9.

[27] Michael Kläs and Anna Maria Vollmer. “Uncertainty in machine learning applications: A practice-driven classification of uncertainty”. In: *Computer Safety, Reliability, and Security: SAFECOMP 2018 Workshops, ASSURE, DECSoS, SASSUR, STRIVE, and WAISE, Västerås, Sweden, September 18, 2018, Proceedings 37*. Springer. 2018, pp. 431–438.

[28] Fortmann-Roe Scott. *Understanding the bias-variance tradeoff*. Accessed: 28.02.2024. 2012. URL: https://scott.fortmann-roe.com/docs/BiasVariance.html.

[29] Trevor Hastie et al. *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. Springer, 2009, p. 223.

[30] Humberto Barreto and Frank Howland. *Introductory econometrics: using Monte Carlo simulation with Microsoft excel*. Cambridge University Press, 2006.

[31] Sancho Salcedo-Sanz et al. "Support vector machines in engineering: an overview". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 4.3 (2014), pp. 234–267.

[32] Sergio González et al. "A practical tutorial on bagging and boosting based ensembles for machine learning: Algorithms, software tools, performance study, practical perspectives and opportunities". In: *Information Fusion* 64 (2020), pp. 205–237. DOI: `10.1016/j.inffus.2020.07.007`.

[33] Leo Breiman. "Random forests". In: *Machine learning* 45 (2001), pp. 5–32.

[34] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". In: *CoRR* abs/1603.02754 (2016). arXiv: `1603.02754`. URL: `http://arxiv.org/abs/1603.02754`.

[35] David Duvenaud. "Automatic model construction with Gaussian processes". PhD thesis. University of Cambridge, 2014.

[36] Juan Emmanuel Johnson, Valero Laparra, and Gustau Camps-Valls. "Accounting for Input Noise in Gaussian Process Parameter Retrieval". In: *IEEE Geoscience and Remote Sensing Letters* 17.3 (2020), pp. 391–395. DOI: `10.1109/LGRS.2019.2921476`.

[37] Andrew McHutchon and Carl Rasmussen. "Gaussian process training with input noise". In: *Advances in neural information processing systems* 24 (2011).

[38] Carl Edward Rasmussen, Christopher KI Williams, et al. *Gaussian processes for machine learning*. Vol. 1. Springer, 2006.

[39] Jochen Görtler, Rebecca Kehlbeck, and Oliver Deussen. "A Visual Exploration of Gaussian Processes". In: *Distill* (2019). https://distill.pub/2019/visual-exploration-gaussian-processes. DOI: `10.23915/distill.00017`.

[40] Trevor Hastie, Robert Tibshirani, and Jerome H Friedman. "Chapter 11 [Chapter Title Here]". In: *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed. Vol. 2. Springer, 2009, pp. 389–415.

[41] Enzo Grossi and Massimo Buscema. "Introduction to artificial neural networks". In: *European journal of gastroenterology & hepatology* 19.12 (2007), pp. 1046–1054.

[42] The pandas development team. *pandas-dev/pandas: Pandas*. Version latest. Feb. 2020. DOI: 10.5281/zenodo.3509134. URL: https://doi.org/10.5281/zenodo.3509134.

[43] Charles R. Harris et al. "Array programming with NumPy". In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: https://doi.org/10.1038/s41586-020-2649-2.

[44] Michael L. Waskom. "seaborn: statistical data visualization". In: *Journal of Open Source Software* 6.60 (2021), p. 3021. DOI: 10.21105/joss.03021. URL: https://doi.org/10.21105/joss.03021.

[45] J. D. Hunter. "Matplotlib: A 2D graphics environment". In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55.

[46] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[47] Andrew Ng. *Machine learning yearning: Technical strategy for ai engineers in the era of deep learning*. deeplearning.ai, 2018.

[48] Charles Spearman. "The proof and measurement of association between two things." In: (1961).

[49] Jennifer A Hutcheon, Arnaud Chiolero, and James A Hanley. "Random measurement error and regression dilution bias". In: *BMJ* 340 (2010). DOI: 10.1136/bmj.c2289.

[50] Stefan Kuhle, Mary Margaret Brown, and Sanja Stanojevic. "Building a better model: abandon kitchen sink regression". In: *Archives of Disease in Childhood - Fetal and Neonatal Edition* (2023). DOI: 10.1136/archdischild-2023-326340.

[51] F. Wever et al. "Atlas zur Warmebehandlung der Stahle". In: *Verlag Stahleisen* (1954), 443 pp.

[52] Qi Lu et al. "Combination of thermodynamic knowledge and multilayer feedforward neural networks for accurate prediction of MS temperature in steels". In: *Materials & Design* 192 (2020), p. 108696. ISSN: 0264-1275. DOI: https://doi.org/10.1016/j.matdes.2020.108696. URL: https://www.sciencedirect.com/science/article/pii/S0264127520302306.

[53] George E Totten. *Steel heat treatment handbook-2 volume set*. Vol. 2. CRC press, 2006, pp. 277–414.

[54] George F Vander Voort. *Atlas of time-temperature diagrams for irons and steels*. ASM international, 1991.

[55] M Atkins. “Atlas of continuous cooling transformation diagrams for engineering steels”. In: (1980).

[56] HE Boyer and AG Gray. “Atlas of Isothermal Transformation and Cooling Transformation Diagrams”. In: *American Society for Metals* (1977), 443 pp.

[57] T Kasugai. “Atlas of CCT Diagrams for Welding (I)”. In: *National Research Institute of Metals, NRIM-special report No. 99-02* (1999).

[58] A. Farrar R.A. Zhang A. Zhuyao. *An atlas of continuous cooling transformation (CCT) diagrams applicable to low carbon low alloy weld metals*. CRC Press, 1995.

[59] Hong-Seok Yang and H. K. D. H. Bhadeshia. “Uncertainties in dilatometric determination of martensite start temperature”. In: *Materials Science and Technology* 23.5 (2007), pp. 556–560. DOI: `https://doi.org/10.1179/174328407X176857`.

[60] *Standard Test Method for Plastic Strain Ratio r for Sheet Metal*. ASTM E517-00. ASTM International. 2000.

[61] *Metallic materials - Tensile testing - Part 1: Method of test at room temperature*. Annex K. ISO 6892-1:2009. International Organization for Standardization. 2009.

[62] C.-L. Cheng, Shalabh, and G. Garg. “Coefficient of determination for multiple measurement error models”. In: *Journal of Multivariate Analysis* 126 (2014), pp. 137–152. ISSN: 0047-259X. DOI: `https://doi.org/10.1016/j.jmva.2014.01.006`.

[63] Andriy Burkov. *Machine learning engineering*. Vol. 1. True Positive Incorporated Montreal, QC, Canada, 2020. Chap. 2, pp. 1–14.

[64] Duo Wang et al. “Model Architecture Adaption for Bayesian Neural Networks”. In: *ArXiv* abs/2202.04392 (2022).

[65] Yong Y Yang et al. “Tensile strength prediction for hot rolled steels by Bayesian neural network model”. In: *IFAC Proceedings Volumes* 42.23 (2009), pp. 255–260.

# A APPENDIX

## A.1 Regression Dilution

In this Appendix the regression dilution is demonstrated for the linear function (equation 7) which was used in the section Investigation of errors on artificial data sets. Acknowledgment to Michael Schmid for deriving the intercept function (see the following page) for the given function. The intercept and slope for the function is calculated as follows:

$$\beta_1 = 1 - \frac{\sigma^2}{\frac{1}{n-1}\sum x_i^2 - \frac{1}{(n-1)}(\sum x_i)^2 + \sigma^2}$$

$$\beta_0 = \bar{y} - \beta_1 \bar{x}$$

where y is the dependent variable, x represents the independent variable, $\beta_1$ is the slope coefficient of the linear regression mode, $\beta_0$ is known as the intercept coefficient, $\sigma^2$ is the variance of the error, and n represents the sample size. In Figure 35 the regression dilution is shown for a sample size n = 5000 and a standard deviation of $\sigma = 0.15$. The blue line shows the ordinary least-squared regression model. The orange line represents the linear model with the analytical coefficient.

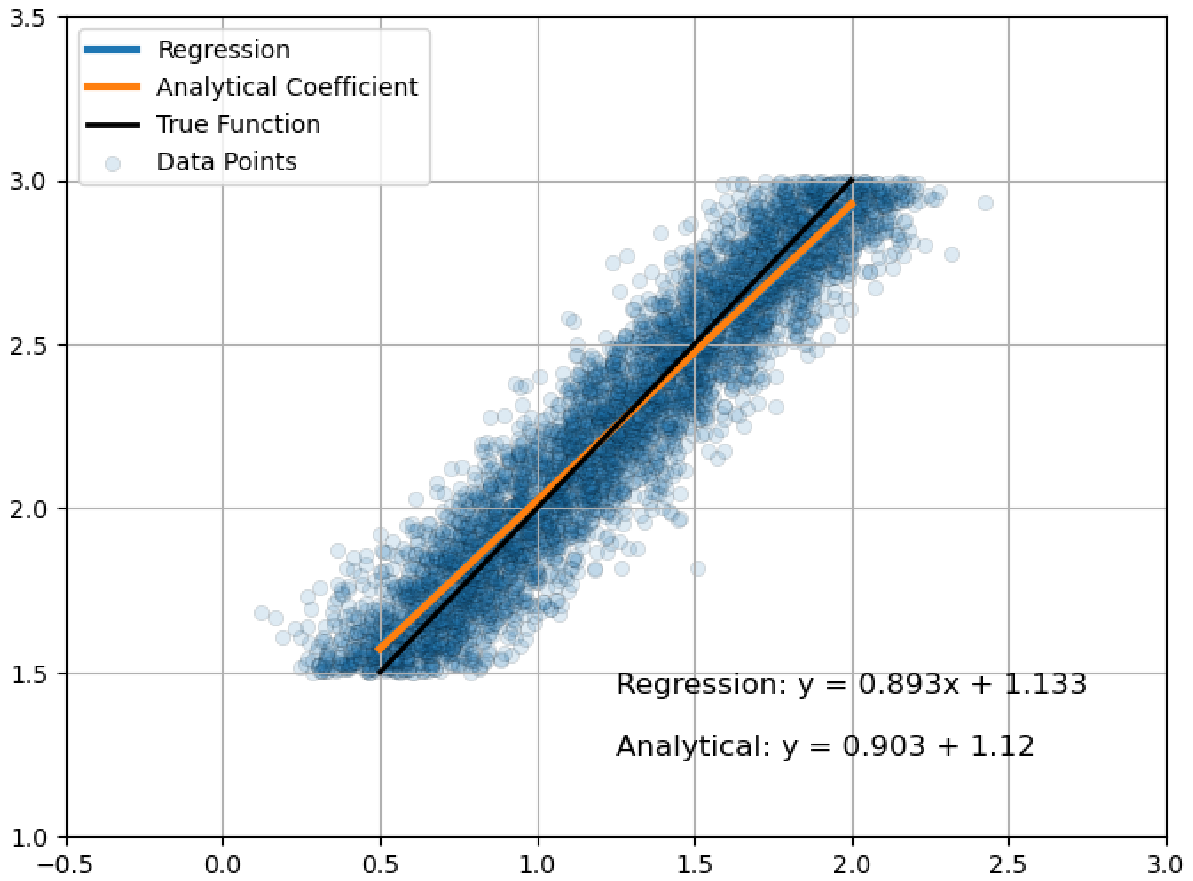


**Figure 35:** *Representation of the regression dilution. It can be seen that the slope decreases from 1 to 0.9 with* $\sigma = 0.15$.

Für den Schätzwert $\hat{\beta}_1$ des unbekannten Parameters $\beta_1$ im linearen Regressionsmodell $Y_i = \beta_0 + \beta_1 x_{i1} + E_i$, $(i = 1, \dots, n)$ gilt bekanntlich

$$\hat{\beta}_1 = \frac{n \sum_{i=1}^n x_i y_i - (\sum_{i=1}^n x_i)(\sum_{i=1}^n y_i)}{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2}. \tag{1}$$

1. Die $y_i$ sind verrauscht: Ausgehend von einem funktionalen Zusammenhang der Form $y = f(x) = 1 + x$ addiert man einen normalverteilten, zentrierten Fehler $\varepsilon_i$ mit Varianz $\sigma^2$ zu den Werten $y_i$. Dann gilt für den Zähler $Z$ von (1):

$$\begin{aligned} Z &= n \sum x_i(1 + x_i + \varepsilon_i) - (\sum x_i)(\sum(1 + x_i + \varepsilon_i)) \\ &= n \sum x_i + n \sum x_i^2 + n \sum x_i\varepsilon_i - (\sum x_i)(n + \sum x_i + \sum \varepsilon_i) \\ &= n \sum x_i + n \sum x_i^2 + n \sum x_i\varepsilon_i - n \sum x_i - (\sum x_i)^2 - (\sum x_i)(\sum \varepsilon_i) \\ &= n \sum x_i^2 - (\sum x_i)^2 + n \sum x_i\varepsilon_i - (\sum x_i)(\sum \varepsilon_i). \end{aligned} \tag{2}$$

   Somit ist

$$\hat{\beta}_1 = 1 + \frac{n \sum x_i\varepsilon_i - (\sum x_i)(\sum \varepsilon_i)}{n \sum x_i^2 - (\sum x_i)^2}. \tag{3}$$

   Nun sind die $\varepsilon_i$ als zentriert vorausgesetzt, d.h. es gilt

$$\sum \varepsilon_i \approx 0. \tag{4}$$

   Weiters gilt $\sum x_i\varepsilon_i = \sum(x_i - \overline{x})\varepsilon_i + \underbrace{\overline{x}\sum \varepsilon_i}_{\approx 0}$, also $\sum x_i\varepsilon_i \approx \text{const} \cdot r_{x,\varepsilon}$, wobei $r_{x,\varepsilon}$ den Pearsonschen Korrelationskoeffizienten zwischen den Stichproben $\{x_i\}$ und $\{\varepsilon_i\}$ bezeichnet. Verschwindet diese Korrelation annähernd, so wegen (4) auch der Zähler des Bruches in (3), und es gilt $\hat{\beta}_1 \approx 1$, wie es aufgrund des funktionalen Zusammenhangs auch zu erwarten war.

2. Die $x_i$ sind verrauscht, der Fehler wird zu den Werten $x_i$ addiert. Dann gilt mit den selben Annahmen wie unter 1. für den Zähler $Z$ von (1):

$$\begin{aligned} Z &= n \sum(x_i + \varepsilon_i)(1 + x_i) - (\sum(x_i + \varepsilon_i))(\sum(1 + x_i)) \\ &= n \sum(x_i + x_i^2 + \varepsilon_i + \varepsilon_i x_i) - (\sum x_i + \sum \varepsilon_i)(n + \sum x_i) \\ &= n \sum x_i + n \sum x_i^2 + \underbrace{n \sum \varepsilon_i}_{\approx 0} + \underbrace{n \sum \varepsilon_i x_i}_{\approx 0} \\ &\qquad - n \sum x_i - (\sum x_i)^2 - \underbrace{n \sum \varepsilon_i}_{\approx 0} - \underbrace{(\sum x_i)(\sum \varepsilon_i)}_{\approx 0} \\ &\approx n \sum x_i^2 - (\sum x_i)^2 \end{aligned} \tag{5}$$

und für den Nenner $N$ von (1):

$$\begin{aligned}
N &= n\sum(x_i+\varepsilon_i)^2 - (\sum(x_i+\varepsilon_i))^2 \\
&= n\sum x_i^2 + \underbrace{2n\sum x_i\varepsilon_i}_{\approx 0} + n\sum\varepsilon_i^2 - (\sum x_i + \sum\varepsilon_i)^2 \\
&= n\sum x_i^2 + n\sum\varepsilon_i^2 - (\sum x_i)^2 - \underbrace{2(\sum x_i)(\sum\varepsilon_i)}_{\approx 0} - \underbrace{(\sum\varepsilon_i)^2}_{\approx 0} \\
&\approx n\sum x_i^2 - (\sum x_i)^2 + \underbrace{n\sum\varepsilon_i^2}_{\approx n(n-1)\hat{\sigma^2}}\ .
\end{aligned} \tag{6}$$

Somit ist

$$\hat{\beta}_1 = 1 - \frac{\hat{\sigma^2}}{\frac{1}{n-1}\sum x_i^2 - \frac{1}{n(n-1)}(\sum x_i)^2 + \hat{\sigma^2}}. \tag{7}$$

Man beachte, daß wegen

$$\begin{aligned}
0 \leq n\sum(x_i - \frac{1}{n}\sum x_j)^2 &= n\sum(x_i^2 - \frac{2}{n}x_i\sum x_j + \frac{1}{n^2}(\sum x_j)^2) \\
&= n\sum x_i^2 - 2(\sum x_i)^2 + (\sum x_i)^2 = n\sum x_i^2 - (\sum x_i)^2
\end{aligned} \tag{8}$$

der Bruch in (7) streng mit $\hat{\sigma^2}$ wächst, die Steigung der Regressionsgerade nimmt also mit zunehmender Streuung des Fehlers ab.

## A.2 Additional Figures for the Friedman 1 dataset

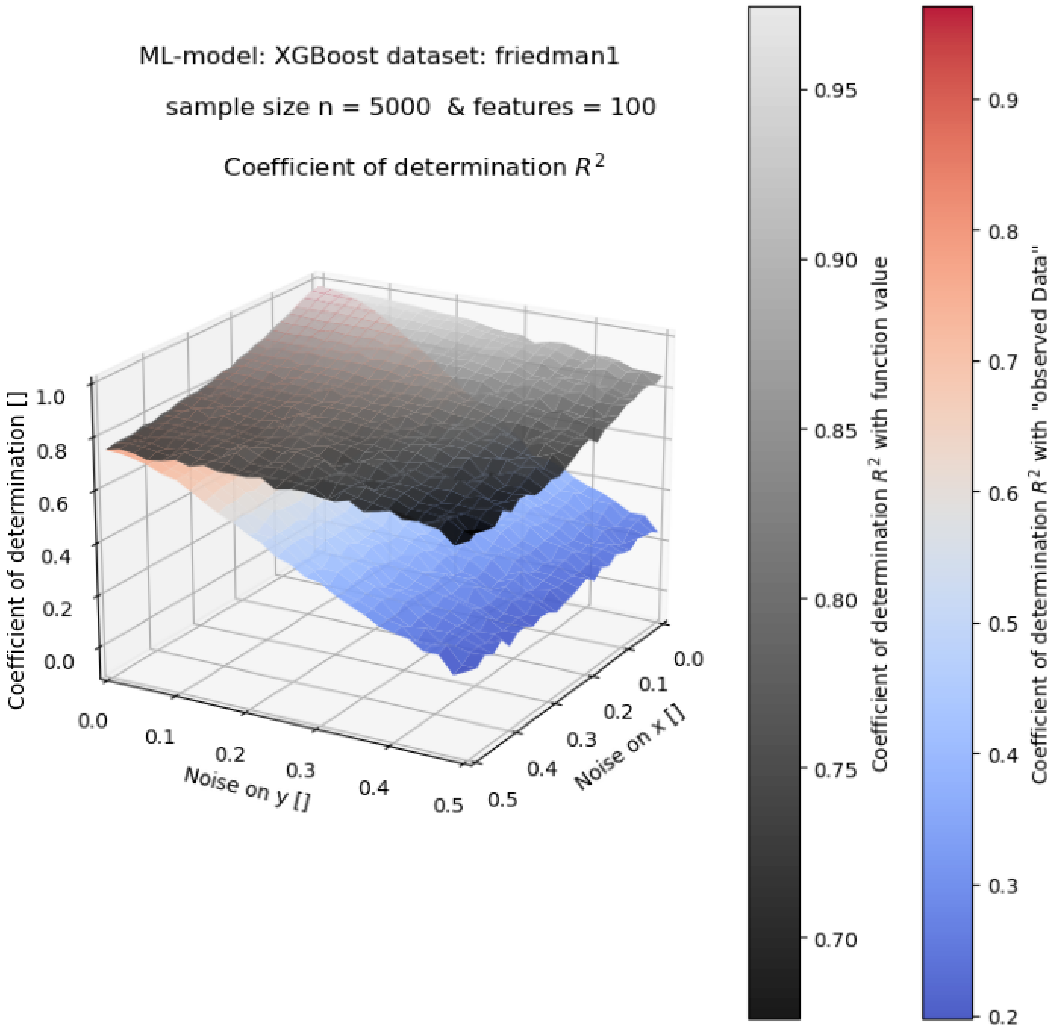


**Figure 36:** *$R^2$ :The data set was created normally with the Friedman 1 function (equation 9), but 95 additional independent features were added.Then the noise was added as described in the section Investigation of errors on artificial data sets.*

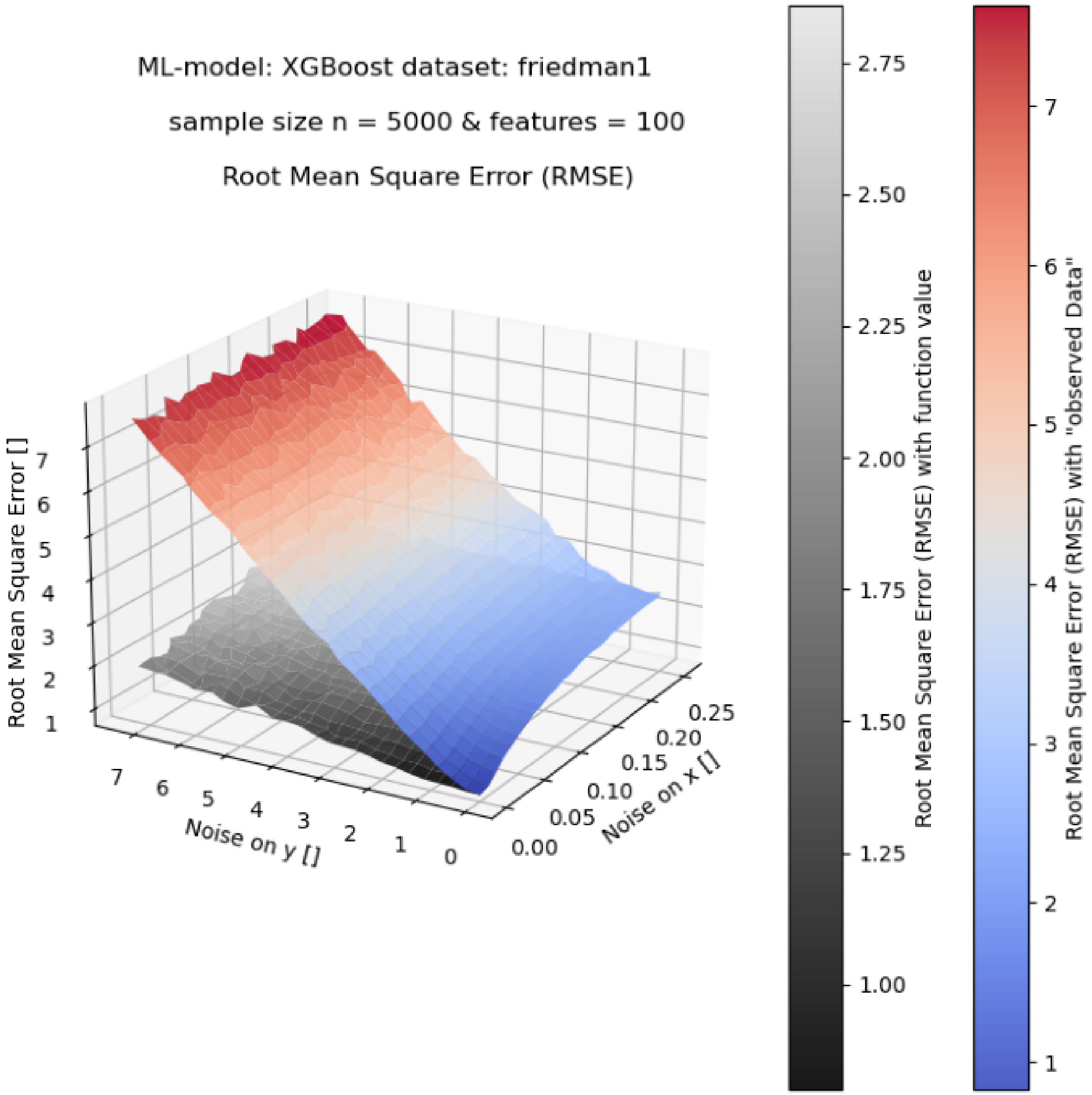


**Figure 37:** *RMSE: The data set was created normally with the Friedman 1 function (equation 9), but 95 additional independent features were added.Then the noise was added as described in the section Investigation of errors on artificial data sets.*

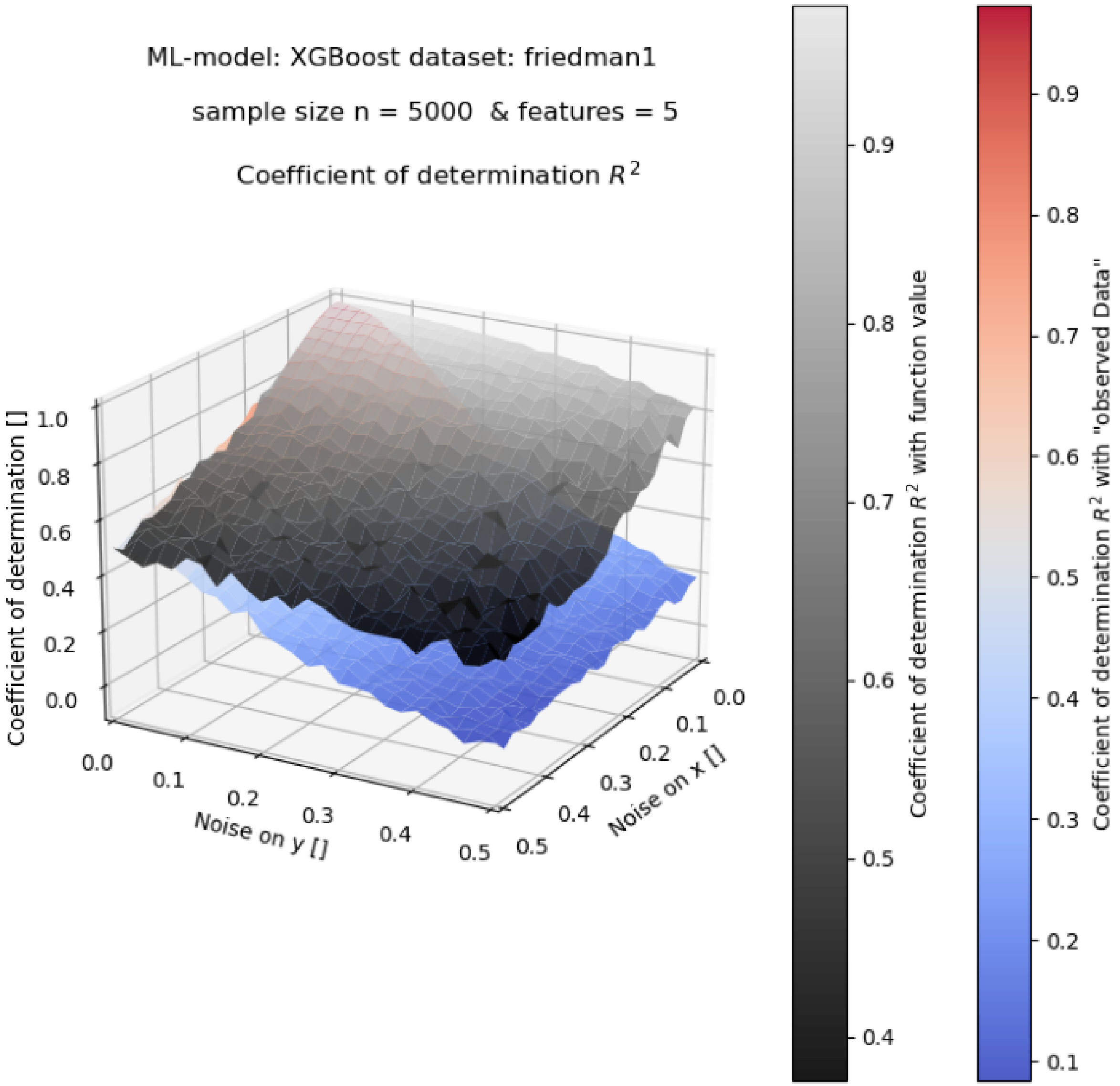


**Figure 38:** *$R^2$ :The data set was created normally with the Friedman 1 function (equation 9), but then instead of a normal distributed noise a Weibull noise was multiplied (numpy.random.weibull (a = 1.1)).*

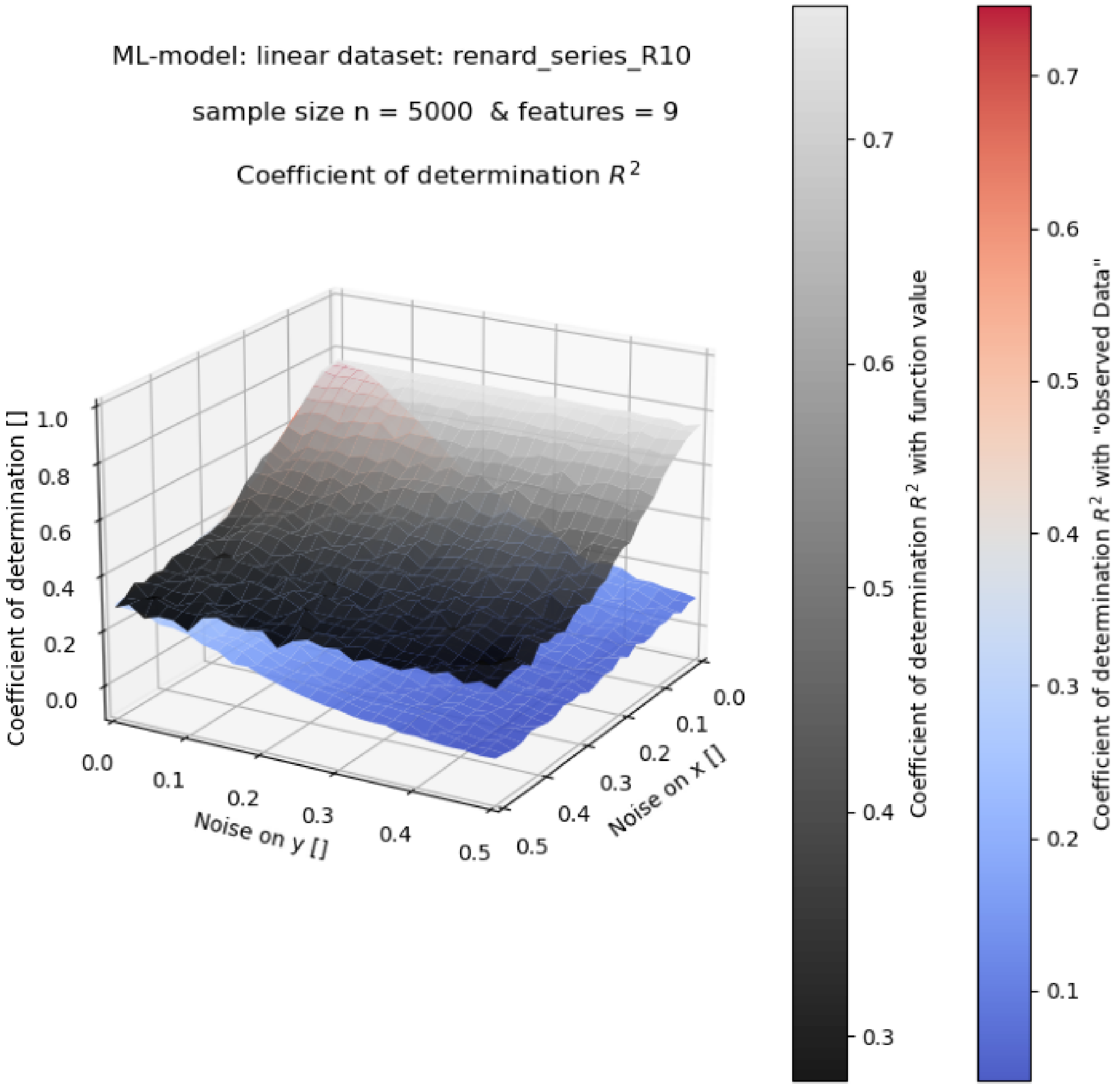


**Figure 39:** *$R^2$: The data set was created normally with the Renard series R10 function (equation 8). Then the noise was added as described in the section Investigation of errors on artificial data sets. But then the features $\mathbf{x_1}$ and $\mathbf{x_9}$ were removed from the dataset.*

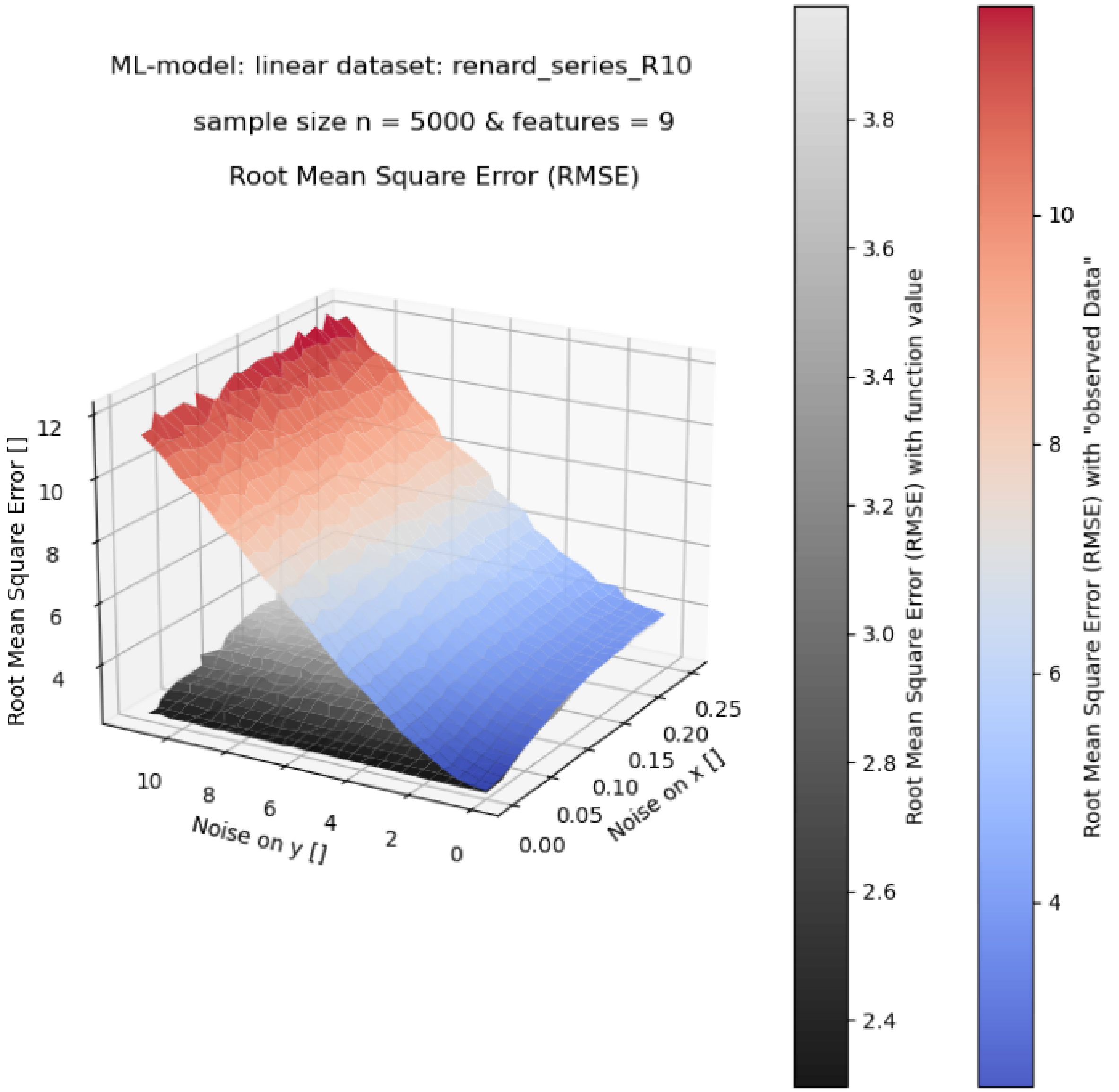


**Figure 40:** *RMSE: The data set was created normally with the Renard series R10 function (equation 8). Then the noise was added as described in the section Investigation of errors on artificial data sets. But then the features $\mathbf{x_1}$ and $\mathbf{x_9}$ were removed from the dataset.*

## A.3 Hyperparameters for the M$_s$ Temperature Dataset

```python

"Hyperparameter for the models for predicting Martensite Start Temperature from Steels"

#Gaussian Process model with Matern kernel
kernel = Matern(length_scale=10, nu=1.5) + WhiteKernel(noise_level=225)
gp = GaussianProcessRegressor(kernel=kernel, n_restarts_optimizer=0,alpha=1e-10,normalize_y=True)

# XGBoost regression model
model_XGB = xgb.XGBRegressor(
        objective="reg:squarederror",
        n_estimators=200,
        learning_rate=0.1,
        max_depth=5,
        reg_alpha=0.1,)

#Random Forest regression model
model_RF = RandomForestRegressor(
        n_estimators=150,
        max_depth=15,
        max_features= None,
        min_samples_split= 2,)

"Grid search parameters for the models to predict the martensite start temperature of steels"

#Gaussian Process model with Matern kernel
param_grid = {
        'kernel__k1__length_scale': [0.1, 0.5, 1, 1.5, 2.5, 10],
        'kernel__k2__noise_level': [100, 225, 400],
        'alpha': [1e-10, 1e-2, 1, 10],
        'n_restarts_optimizer': [0, 5, 10],
    }


#XGBoost regression model
    param_grid = {
        'n_estimators': [50,75,100, 200, 300, 400, 500],
        'learning_rate': [0.01, 0.05, 0.1, 0.15],
        'max_depth': [3, 5,6,7,10, 15, 30],
        'reg_alpha': [0, 0.1, 0.2, 0.3, 0.5]
    }

#Random Forest regression model
    param_grid = {
        'n_estimators': [100,150,200, 300, 400, 500],
        'max_depth': [3,5,7,10,15],
        'min_samples_split': [2, 5, 10],
        "max_features":[None, "log2"]}
```

## A.4 Error analysis of the r-value

The coefficient of variation $v(r)$ for the r-value is calculated as described in ASTM E 517-00 Appendix X1 using the following formula:

$$\nu(r) = \frac{1+r}{\varepsilon_l} \left\{ \nu(W_0)^2 \left( \frac{1+r}{r} \right)^2 \cdot [1 + \exp(-2\varepsilon_w)] + \nu(l_0)^2 [1 + \exp(-2\varepsilon_l)] \right\}^{1/2}$$

where:

- $r$: r-value (plastic strain ratio)
- $\nu(W_0)$: coefficient of variation of the original width measurement
- $\nu(l_0)$: coefficient of variation of the original length measurement
- $\epsilon_l$: length strain
- $\epsilon_w$: width strain

$$\epsilon_w = -\left( \frac{r}{1+r} \right) \epsilon_l$$

The standard deviation $s(r)$ was then calculated in the following way:

$$s(r) = v(r)r$$

For the coefficient of variation of the original width & length measurement, a value of 0.001 was chosen, as also done in ASTM E 517-00 Appendix X1. The value of 0.18 was chosen for the length strain.

# B Use of Generative AI tools

Figure 41 shows the workflow that I used for Generative AI in my thesis. The first step is that I put my writing ideas in ChatGPT and DeepL. This step involved formulating specific prompts to guide the AI in generating relevant content. The next step is that I reviewed the results generated by ChatGPT and DeepL. After receiving the initial results, I carefully evaluated the suitability of the content. I made the necessary changes to the AI-generated text to ensure that it met my research objectives and academic standards. This step involved editing, rephrasing, and fine-tuning the content to meet the requirements. This iterative workflow continued until I was satisfied with the quality of the final output.

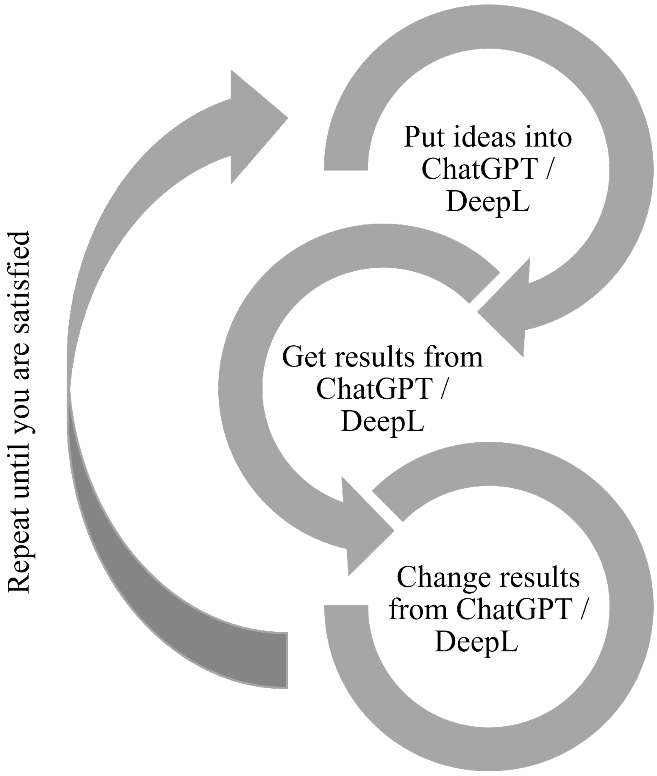

Put ideas into ChatGPT / DeepL
Get results from ChatGPT / DeepL
Change results from ChatGPT / DeepL
Repeat until you are satisfied

**Figure 41:** *Illustration of the iterative workflow for the generative AI that was used.*

Used generative AI tools for writing where: ChatGPT 3.5, ChatGPT 4.0, ChatGPT 4o, DeepL Write, DeepL Translate and Writefull. ChatGPT 3.5, ChatGPT 4.0 was also used for some programming and debugging tasks.