# Developing a Labour Cost Prediction Algorithm for an Industrial Packaging Configurator

Sten Eijkholt

*A thesis submitted to the Faculty of Behavioural, Management and Social Sciences (BMS) in partial fulfilment of the requirements of MSc Industrial Engineering and Management*

University of Twente

Enschede,

June 14, 2024

**University supervisor:**
Reinoud Joosten
Wouter van Heeswijk

**Company supervisor:**
Peter Smit

# COLOPHON

**MANAGEMENT**
Department BMS
Financial Engineering and Management

**DATE**
June 14, 2024

**VERSION**
Version 2.2.2

**TITLE**
Developing a Labour Cost Prediction Algorithm for an Industrial Packaging Configurator

**PROJECT**
Master thesis

**AUTHOR(S)**
S.M. Eijkholt

**POSTAL ADDRESS**
P.O. Box 217
7500 AE Enschede

**WEBSITE**
www.utwente.nl
www.meilink.com

UNIVERSITY OF TWENTE.

meilink

# ACKNOWLEDGEMENTS

The thesis "*Developing a Labour Cost Prediction Algorithm for an Industrial Packaging Configurator*" was created in fulfilment of the Industrial Engineering and Management Master's Degree. This thesis marks the end of my academic education at the University of Twente, specializing in Financial Engineering & Management. The research was commissioned by Meilink Beheer B.V. and supervised by Peter Smit.

I would like to express my sincere gratitude to Peter Smit for his incredible guidance, support, encouragement, and this unique opportunity. I extend my appreciation for providing access to the necessary resources for my research. Moreover, I would like to express my appreciation to my supervisors from the University of Twente, Reinoud Joosten and Wouter van Heeswijk. I am grateful for the valuable feedback, interesting meetings, and support during the execution of my research.

Lastly, I would like to express my deepest gratitude to my girlfriend Sofie Bruns, my family, and dear friends Ruben Gerhardus, Twan de Roode, Youri Stadsveld, who have been an invaluable source of support and inspiration throughout this project. From this point forward, I continue my career as an Operational Excellence Project Manager at Meilink.

Forever assuming $\mathrm{PV} = \mathrm{nRT}$,

*Sten Eijkholt*

Enschede, June 14, 2024

# EXECUTIVE SUMMARY

Meilink has been a Dutch family business in industrial packaging since 1874 with 470 employees and nine branches throughout the Netherlands. The term 'industrial packaging' entails tailored engineering of custom packaging for a variety of customers and industries. Meilink's products usually consist of a wooden crate with an engineered 'insert'. The insert is tailor-made based on the product transported inside. The introduction of a product configurator allowed customers to design a large number of configured-to-order (CTO) products. This configurator accurately calculates the quantities of materials used. However, labour cost remains difficult to estimate for a unique product. This problem prevents the firm from providing customers with accurate quotations, and scheduling activities precisely.

Therefore we initiated this investigation with the objective: *To develop an algorithm that predicts labour costs of CTOs with the actual labour cost falling within the 95% confidence interval of the predictions.* Based on the objective, we formulated the research question: *How can the firm accurately and systematically predict labour costs for configurated products?* Our methodological approach involved a literature review to create a framework of relevant approaches, data requirements, and validation methods. Based on our literature review, we found that machine learning techniques are suitable to solve our problem, due to the ability to process large numbers of data and handle complex underlying relationships. We concluded to test the following nine supervised machine learning techniques: *Linear Regression* (LR), Multi-Layer Perceptron (MLP), *Gaussian Process Regression* (GPR), *Random Forest Regression* (RGR), *Support Vector Machines* (SVM), *Decision Tree Regression* (DTR), *Gradient Boosting Regression* (GBR), *K-Nearest Neighbours* (KNN), and *Extreme Gradient Boosting* (XGB).

We evaluated performance using two performance metrics and an accuracy percentage. We selected Mean Squared Error (MSE) and Akaike Information Criterion (AIC) as performance metrics, and we define accuracy as the percentage of predictions which contain the actual value within its 95% confidence interval. The dataset we used contains 811 product samples manufactured from July 18, 2023, until March 14, 2024, separated into a subset for each product. Each sample in the product subsets describes a product with twenty-seven original features and two engineered features, linked to a unique labour time in minutes. Subsequently, we split the datasets in 80% training data and 20% test data. Training the model with the first 80% enables it to map the relationships with labour cost. We then use the remaining 20% test data to investigate the degree to which a model predicts accurately compared to other models, expressed in a composite score of the MSE and AIC. We assign products to their most appropriate technique based on the composite score; a higher composite score suggests the more appropriate fit of that product-model combination. After we assign every product to a method, we investigate further validation with K-fold cross validation. During the validation phase, we extract the most important features from the model, which provide valuable insights into what components of the manufacturing process contribute the most labour costs. Targeting these cost-intensive areas for optimisation or employee training has the most potential cost reductions.

We can assess the improvement of our algorithm by comparison to the firm's current approach. Table A lists the machine learning techniques assigned to each product, along with the accuracy before and after implementation of our labour cost prediction algorithm. We

assess the significance of the improvements with a paired t-test, where we observed a significant overall accuracy improvement of 21.74% with a 5% level of significance.

***Table A: Improvements in labour cost prediction accuracy per product.***

| Product | Method | Accuracy (before) | Accuracy (after) | $df$ | t-statistic | p-value | Improvement |
|---|---|---|---|---|---|---|---|
| **PCF102** | GBR | 17.73% | 50.68% | 218 | 5.2220 | <0.0001 | 32.95% |
| **PCF103** | XGB | 29.45% | 39.38% | 159 | 3.1767 | 0.00190 | 9.93% |
| **PCF201** | SVM | 29.41% | 50.00% | 67 | 3.8245 | 0.00033 | 20.59% |
| **PCF206** | DTR | 25.26% | 44.21% | 94 | 3.6775 | 0.00041 | 18.95% |
| **PCF401** | KNN | 38.95% | 55.06% | 186 | 3.1607 | 0.00200 | 16.11% |
| **PCF407** | LR | 14.43% | 46.34% | 81 | 2.5301 | 0.01312 | 31.91% |
| **Overall** | | **25.87%** | **47.74%** | | | | **21.74%** |

We analysed error distribution, which suggests that the model is generally accurate. The error distributions show a peak around zero, which indicates that the majority of the predictions are relatively close to the true value and there are fewer instances where the model makes large errors. Additionally, we investigated the skewness of the error distributions, which suggested that three of the techniques are inclined to underestimate labour costs, while other techniques showed approximately symmetrical error distributions. The sales department can benefit from underestimation because it allows them to quote more competitive prices, on the other hand, underestimating labour time disadvantages operations by causing a tight schedule, as well as lower profit margins.

Our results indicate that we achieved a significant improvement in labour cost estimation, by using historical data to train our model, selecting appropriate machine learning techniques, and validating its performance. An increased labour cost prediction accuracy enables Meilink to quote more competitively and make informed decisions. Furthermore, the algorithm contributes to improved communication between sales and operations by minimizing disagreement over scheduled time. Feature importances extracted from our algorithm show what features add the most value to a product. Identifying cost intensive components within the production process allows for efficient resource allocation and to identify areas where cost improvement or employee training impacts the most. Overall, our solution not only improves Meilink's labour cost prediction accuracy by 21.74%, but also aligns with Meilink's long term commitment of improving customer satisfaction, maintaining a competitive advantage, and ensuring continuity of the firm.

# GLOSSARY

| Term | Definition |
|---|---|
| *Black Box Model* | Process description focussed on inputs and outputs, disregarding the internal process. |
| *Epoch* | Artificial Neural Network iteration through all code. |
| *Example* | A set of input-output combinations used a training data for machine learning, referred to as a labelled record. |
| *Invoice* | Document that maintains a record of transaction between buyer and seller. |
| *Label* | The result or outcome of a record. |
| *Merkato* | Product configurator software. |
| *Post-calculation* | Calculated labour cost of a product based on e.g., production times, used materials, and other costs. |
| *Pre-calculation* | Estimated labour cost of a product prior to manufacturing. |
| *Quotation costing* | Strategy in negotiations where supplier presents a clear cost structure and breakdown of costs. |
| *Quotation or Quote* | Formal statement setting out the estimated cost for a particular product or set of products. |
| *Record* | Collectively forms a dataset for machine learning and holds the features. |
| *Reinforcement learning* | Optimising a decision-making process through interactions and adjusting the process according to the response. |
| *Residuals* | Deviation of prediction from actual value. |
| *Shopfloorcontrol* | Extension of the ERP system that allows users to track labour times. |
| *Training data* | A dataset provided to a machine learning algorithm to find relationships and make predictions based on the obtained knowledge. |

# NOMENCLATURE

| Acronym | Definition |
|---|---|
| ABC | Activity-based costing |
| AIC | Akaike Information Criterion |
| ANN | Artificial Neural Network |
| CER | Cost Estimation Relationship |
| CI | Confidence Interval |
| CPC | Cost Price Calculation |
| CTO | Configure to Order |
| DTR | Decision Tree Regression |
| ERP | Enterprise Resource Planning |
| ETO | Engineering to Order |
| FE | Feature Engineering |
| FS | Feature Selection |
| GBR | Gradient Boosting Regression |
| GPR | Gaussian Process Regression |
| IQR | Interquartile Range |
| KNN | K-Nearest Neighbour |
| LR | Linear Regression |
| ML | Machine Learning |
| MLE | Maximum Likelihood Estimation |
| MLP | Multi-Layer Perceptron |
| MSE | Mean Squared Error |
| MST | Mean Total Sum of Squares |
| MTO | Made to Order |
| OFAT | One Factor at a Time |
| PCF | Product Configuration |
| PDF | Probability Density Function |
| PP | Prediction Propagation |
| RFR | Random Forest Regressor |
| RMSE | Root Mean Squared Error |
| RMSLE | Root Mean Squared Logarithmic Error |
| RV | Random Variable |
| SFC | Shopfloorcontrol |
| SHAP | Shapley Additive Explanations |
| SLR | Systematic Literature Review |
| SMAPE | Symmetric Mean Absolute Percentage Error |
| SVM | Support Vector Machines |
| XGB | Extreme Gradient Boosting |

| Variable | Definition |
|---|---|
| $W_i$ | Relative weight |
| $\hat{p}$ | Sample proportion |
| $q_0$ | Non-systematic quote |
| $q_1$ | Systematic quote |
| $x_i$ | Instance |
| $x_i$ | Predicted value |
| $\bar{y}$ | Mean |
| $y_i$ | Actual value |
| $\hat{\theta}$ | Maximum likelihood estimator |
| $\sigma_\epsilon^2$ | Maximum likelihood of the residual's variance |
| $\mathcal{L}$ | Maximised value of the likelihood function model |
| $C$ | Cost |
| $C$ | Confidence level |
| $K$ | Number of folds |
| $Q$ | Quantity |
| $d$ | Dimensionality |
| $k$ | Number of estimated parameters |
| $m$ | Number of instances |
| $n$ | Number |
| $t$ | Time |
| $z$ | Z-score corresponding to the confidence level |

# LIST OF FIGURES



# LIST OF TABLES

# TABLE OF CONTENTS

# 1

# INTRODUCTION

Meilink Beheer B.V. (referred to as “Meilink”) started in 1874 as a timber yard and sawmill in Borculo. The firm later specialised in manufacturing wooden packaging, high-tech cleaning, and transporting capital goods all around the world. Meilink has expanded this specialty as a family business over 150 years. Acquisitions and strategic choices, such as in-house forwarding activities, added them to the market leaders in the Benelux in 2024. Moreover, the firm is amongst the top internationally (Nieuwenhuis, 2014) in industrial packaging with approximately 470 employees and 9 branches.

Satellites and advanced chip machines from renowned companies are not unusual contents of the tailor-made packaging. The wide range of daily activities involves processing product components ranging from wood, cardboard, steel, plastics, and foam into packaging. Under the slogan ‘Securing Value’, the firm strives to meet all customer requirements with minimal use of materials and fast delivery times, reinforced by the engineering departments that incorporate technical requirements in the design. We present Figure 1.1 as an example of one of the products and services.

We distinguish product ranges into products which are made-to-order (MTO), products which can be configured-to-order (CTO), and tailored packaging. The latter product type is engineered-to-order (ETO), due to the engineering process involved to incorporate specific and unique requirements. All production processes feature a range of manual and automated operations, making human expertise essential. Labour costs are responsible for a significant share of the product (cost) price amongst the other costs (i.e., material, machines, capital, energy, etc.).

***Figure 1.1: Example of Meilink’s industrial packaging (Meilink B.V., 2024).***

The cost price and the corresponding margin are basic measures to ensure a profit on a product. CTOs are responsible for approximately 30% of the turnover. The specifications of a CTO are configured in the software *Merkato*, which allows the customer to adjust many factors, such as length, width, and height. More specific factors can also be customised in this software, for instance, positioning forklift and carriage beams under the crates, adding lifting

eyes, and adjusting the position of hinges. Although material quantities can be derived from the design, labour time remains difficult to estimate for unique products. Labour significantly affects the cost price of a product, therefore offering customers a quotation based on an estimation alone, comes paired with a risk.

Employees of the sales department aim to provide customers with a quotation, listing a price that approximates the actual price that appears on the invoice. The price listed on the invoice is based on the expected costs and profit margin of that product. Frequent discrepancies between quotation and invoice can induce frustration for customers as prices can exceed their expectations. Besides, an incorrect cost price calculation resulting in a lower price than the actual cost price, can cause loss or missed profit. Hence, pricing accurately is a key requirement to ensure Meilink's profit. The finance department desires a practical method that accurately calculates labour costs for a unique product. The main difficulty lies in the isolation of the effect of changing a single input parameter in the configurator. If such relationships exist, the model must also find relationships between input parameters that affect the labour cost. Ultimately, to provide a systematic method to determine the labour cost based on an expanding dataset.

Therefore, we intend to develop a model that improves the accuracy of quoting labour costs. We define improving accuracy as bringing our prediction closer to the measured value. The primary focus is on improving the accuracy of the labour cost calculations. We aim to investigate how the combination of labour times and configurator data can improve the labour cost calculations. An advantage of mapping the labour cost is that components of the product cost price can be traced back to find unnecessary costs. A solution enables Meilink to find which process components can improve to affect (or reduce) the labour cost.

To conclude, we aim to improve the product pricing process by analysing labour cost prediction for varying parameters in the configurator. Due to the large number of unique products, the labour costs are complex to estimate, while the configurator only calculates resources accurately. Hence, we focus on systematically determining labour costs prior to production, to allow the sales department to quote a reliable price by reducing the distance between quote and invoice.

A practical solution to systematically estimate labour costs benefits the entire firm. Scheduling efficiency can be improved by gaining insights on more exact labour times. We expect to contribute to customer satisfaction by improving quotation accuracy, because quoting with a higher precision reduces the risk of cost overruns for the customer.

# 2

# CONTEXT ANALYSIS

In this chapter, we go into more detail of the context of the research problem. Placing the problem in the context of the firm helps to understand the stakeholders involved, influences from departments, and decision making. We described the outline of the general problem in the earlier chapter. We build on that problem and analyse its causes and effects in more detail. Furthermore, we describe the current situation, and illustrate its processes. We identify the core problem with the problem cluster we developed. Finally, we derive research questions to design a step-by-step research plan and solve the core problem.

## *2.1 Product Configurator*

Customers showed demand for variations on existing products, which brought the configurator to life, where customers can tailor their own CTOs. Imagine that Meilink produces a standard wooden crate, and a customer can configure numerous variables such as length, width, and height. Logically, an infinite number of products are possible. Scheduling manufacturing for a standard product is not complex, it only requires some time to show what the approximate labour time is for that product. This is not necessarily the case for CTOs, where an infinite number of possibilities make it difficult to estimate the labour time needed to manufacture accurately. Merkato contains a list of continuous and discrete parameters, adding up to a total of eighty-five variables.

Table 2.1 describes the turnover and the characteristics of each product type. An improvement can have high impact as CTOs make up about 30% of the turnover. In contrast to the standard products, which the firm produces in high volume with low variety, CTOs are produced in medium volumes and high variety. High variety is the result of the large number of possibilities in the configurator. Therefore, it is difficult to estimate manufacturing time and schedule operations.

***Table 2.1: Variety, volume and turnover share per product category.***

| Category | Acronym | Turnover [%] | Variety | Volume |
|---|---|---|---|---|
| *Made to Order* | *MTO* | *50%* | *Low* | *High* |
| *Configure to Order* | *CTO* | *30%* | *High* | *Medium* |
| *Engineering to Order* | *ETO* | *20%* | *High* | *Low* |

Estimated labour time is currently based on the *pre-calculation*, determined with a tool in the ERP system that uses configurator parameters. The number of scheduled actions is multiplied with the average time per action, for each operation. For example, the configurator indicates that long wooden beams should be divided in smaller sections. A number of saw cuts are required: If 14 saw cuts are listed, and the standard time for a single saw cut is 6 seconds, this means that the pre-calculation reserves 84 seconds for saw cuts. The same applies for all other operations, such as, the number of holes drilled, screws screwed, or nails hammered.

The firm introduced *Shopfloorcontrol* (SFC) as the designated software to document processing times. SFC is a software, allowing employees to track processing times by

scanning the barcode on the production instructions at the start and finish. The collected data are the resource for *post-calculation* and could assist in improving pre-calculations. However, in current operations, no valid method exists to estimate the labour time for a new unique product, based on the collected data.

## 2.2 Quotation Costing Process

Now we established the current method of quotation based on pre-calculation, the process of sales, quotation, and invoice can be described and analysed. Furthermore, the integration of SFC can also be integrated into the design for the application in this context. By Figure 2.1, we illustrate the interaction between customer, sales department, and production, including the data update recycle.

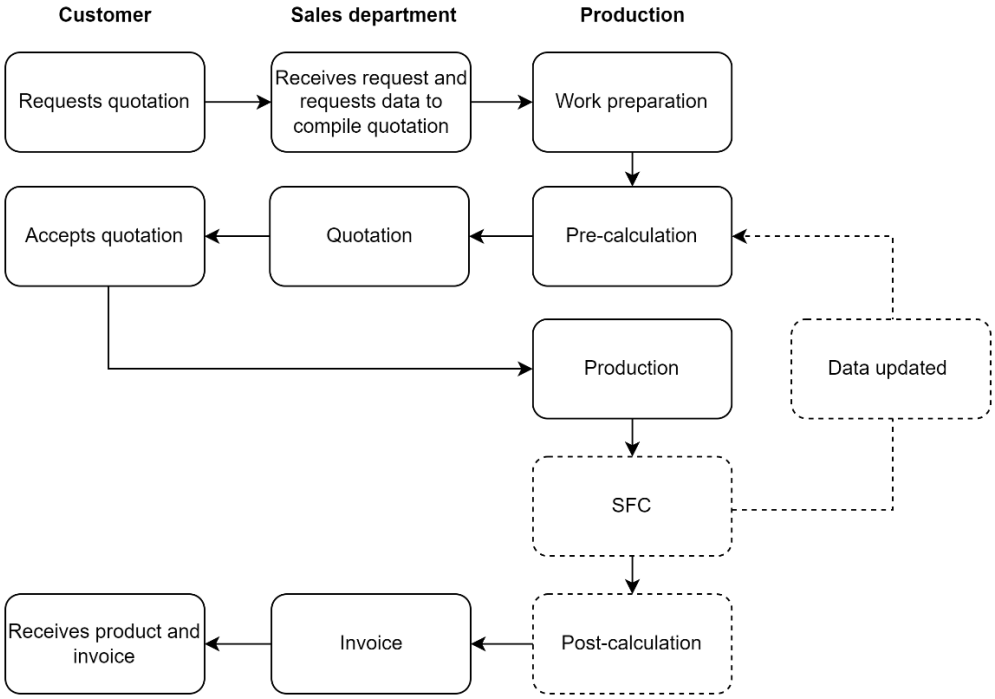



***Figure 2.1: Labour costing process for pre- and post-calculation.***

'Data updated' functions as an update from SFC back to pre- calculation. MTOs are assigned to the proper pre-calculation. However, estimating labour costs for CTOs and ETOs that have not been produced before, remains an obstacle.

## 2.3 Product Costing in Manufacturing

The price that a customer must pay for a product depends on a number of factors. Costs incurred in the manufacturing process are typically divided in direct and indirect costs. Indirect costs (or overhead costs) are not directly related to the manufacturing process (Eksteen & Rosenberg, 2002). Instead, these costs are more related to the general operation of the firm (Narong, 2009). Examples of indirect costs are energy supply, facility, and (essential) staff activities like accounting. Direct costs are typically the materials, resources, transportation, and machines used in the manufacturing process.

In the current cost estimation process, the resources used in manufacturing are summarised to a total, taking the hourly cost of machines into account. Subsequently, the firm bills labour hours based on a calculated estimate, which partially relies on a comparison with

similar products and experiences. Transportation costs are calculated as a function of a fixed rate per kilometre, storage as a function of time and surface, and machine costs according to the running hours. The margin and overhead costs are calculated as a percentage.

## *2.4 Problem Analysis*

The context we previously provided combines a cluster of problems, interconnected with cause-and-effect relationships. The core problem in its centre, visible in Figure 2.2, represents the root of the negative effects experienced. Summarised, the customers demanded variations of existing products to tailor their needs. Meilink developed a product configurator to allow for a smooth and simple configuration process for sales as well as customers. However, the configurator contains up to eighty-five discrete and continuous variables. Therefore, the number of combinations is infinite. This result brought along some problems. Every CTO is unique, and therefore, manufacturing processes must adapt. Although the resources are immediately specified, the labour time remains complex to estimate. Which resulted in the core problem that the circumstances are too complex. Consequently, required labour time and cost cannot be systematically determined without development of a mathematical model. Employees of the sales department experience customer complaints when invoices exceed quotations due to inaccurate cost estimations. In this case, the consideration between taking a loss or risking disappointing a customer is difficult. However, considering a situation where a quote turns out to be higher than the actual costs is a trade-off as well. The opportunity of some additional profit arises due to the customer's agreement to the (high) quote, but it can affect buyer-supplier relationships. Furthermore, scheduling issues and operation delays occur due to projects requiring more labour time than estimated.

Following the chain of problems back to the problems that do not have a direct cause, leads to the core problem. Symptoms noticed by staff or customers would initially be perceived as the problem itself. However, the problem cluster in Figure 2.2 reveals a deeper layer of cause-and-effect relationships. Investigation of the core problem of the initial cluster revealed a problem with no direct cause. In this case, the core problem lies in the complexity of modelling the behaviour of labour cost. The enlarged area in Figure 2.2 is not part of the problem cluster but is an expansion of the core problem marked in grey.

The problem cluster only describes problems related to the core problem addressed in our research. Hence, we intentionally excluded a number of items to increase the clarity of Figure 2.2. For instance, we only record a problem in the cluster if we are sufficiently confident that the problem actually occurs. Therefore, reducing the risk of solving problems that are not a problem after all. Additionally, when we cannot influence an occurrence, it cannot be a core problem. Furthermore, for the initial node of the cluster (‘Demand for variations in existing products’) we do not stipulate why this occurs. There is no relevance to speculate the decision of expanding the product range, thus, we exclude research surrounding this subject from the scope of research.

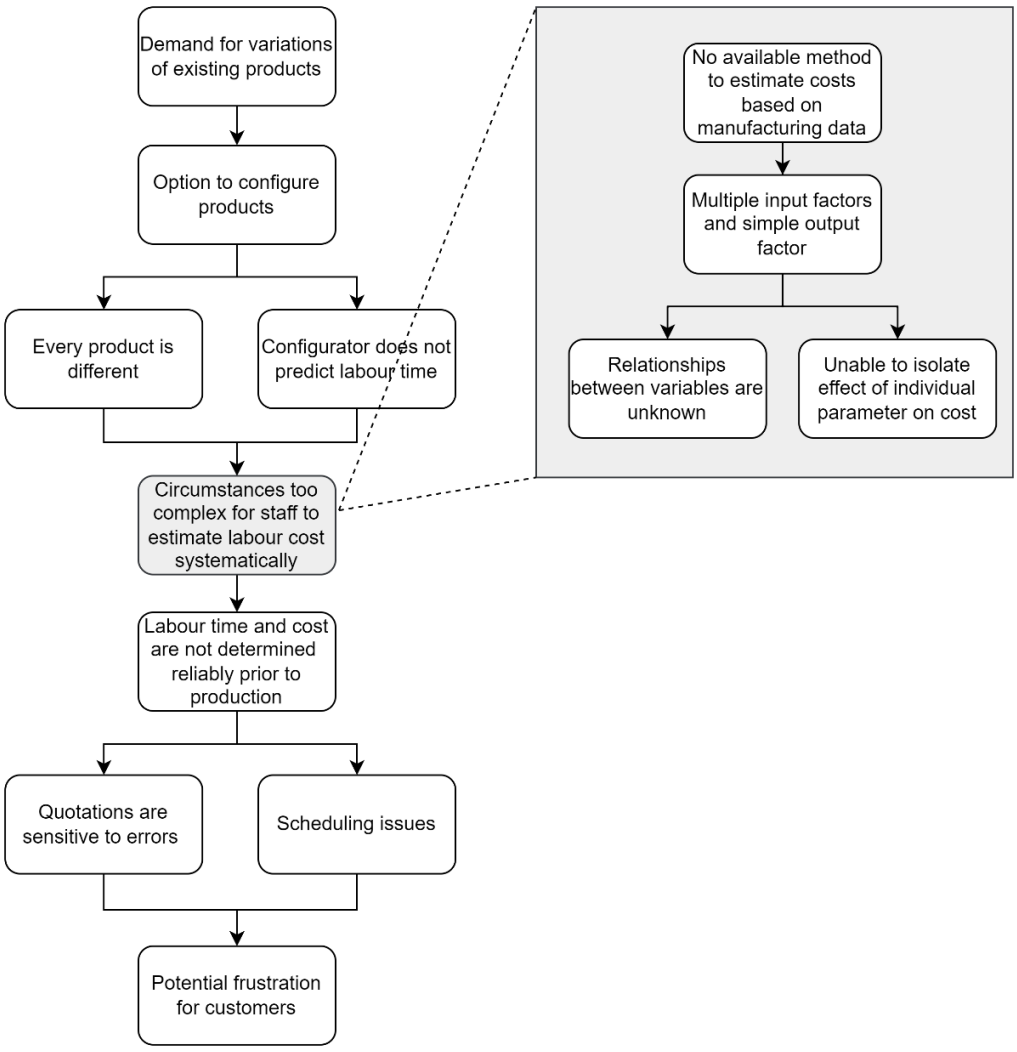



***Figure 2.2: Problem cluster.***

We further decompose the core problem into *action problems* and *knowledge problems*, as the next step of analysing the core problem we identified. With our problem cluster, we established that the core problem relates to the complexity of modelling behaviour of labour cost. Furthermore, the discrepancy between norm and reality lies in the reliability of quotations. Reality is that quotations are not systematically priced and therefore sensitive to error. The norm, however, is that quotes represent the actual costs. Assigning variables to the norm and reality allows for a measurement of the discrepancy. Current estimations $(q_0)$ represent the variable for reality and the actual labour cost represents the variable for the norm. The actual labour cost $(q_1)$ is available since employees empirically measure labour hours during manufacturing. However, these data are not available at the time of quotation. Therefore, an objective is to predict an accurate labour cost, in such a way that the actual labour cost is within its 95% confidence-interval (CI).

We express the variable for the discrepancy between $q_0$ and $q_1$ with the Mean Squared Error (MSE). The MSE is generally used to compare the fit of models or to quantitatively express a deviation between to variables, where larger errors are penalised more (Flach, 2019; Gupta et al., 2009). The variables and each of their properties are summarised in Table 2.2.

***Table 2.2: Definition of variables in problem context.***

| Variable | Indicator | Symbol | Unit |
|---|---|---|---|
| Reality | *Current estimation* | $q_0$ | [ min ] |
| Norm | *Actual labour cost* | $q_1$ | [ min ] |
| Discrepancy | *Measured with the Mean Squared Error* | $MSE$ | [ min ] |

We evaluate the current performance of the quotation estimation to set a benchmark, and eventually, gain an insightful comparison with our improvement. We express performance with MSE and list accuracy in Table 2.3, for six types of products. For each estimation, we assess whether it is sufficient by finding its 95% confidence interval. If the true value lies within its confidence interval, we conclude the estimation to be sufficient. We defined the unit of accuracy as the percentage of sufficient estimations out of all estimations. We obtained these values by comparing estimated labour to actual labour (measured in minutes).

***Table 2.3: Current performance of quote estimation.***

| Product | n | MSE | Accuracy |
|---|---|---|---|
| PCF102 | 219 | 71339.79 | 17.73% |
| PCF103 | 160 | 18927.00 | 29.45% |
| PCF201 | 68 | 11069.96 | 29.41% |
| PCF206 | 95 | 4814.56 | 25.26% |
| PCF401 | 187 | 10920.45 | 38.95% |
| PCF407 | 82 | 2822.42 | 14.43% |

We mapped the cause-effect relationships between the problems, and we identified the core problem. We expressed norm, reality, and the discrepancy as variables. A distinction can be made between action and knowledge problems, in search of the solution. An action problem is the actual discrepancy, in this case, the gap between systematic and non-systematic quotations. The problem owners are the departments of sales, finance, and operations, since sales experiences negative effects in customer relationships, finance is unable to trace costs systematically, and operations experiences delays in manufacturing.

Knowledge problems, on the other hand, deal with situations in which information is missing. While missing knowledge is a part of the cause, these knowledge problems occur in the initial approach of solving the core problem. For instance, a knowledge gap exists for methods directly applicable to model labour cost behaviour. The research population concerned in this knowledge problem are the stakeholders that decide the constraints and criteria that should be involved in the model. Furthermore, research problems can be further divided into two subcategories: descriptive or explanatory (Heerkens & Winden, 2017). Descriptive aims to know *what,* while the latter describes the *why*. Considering the context of this research, finding out *what* behaviour occurs is more relevant, as opposed to *why* it occurs. However, we do not exclude the *why* from our research.

To summarise, the general action problem results from the discrepancy between norm and reality. The norm is providing customers with reliable quotations, representing the actual

costs. Reality is that components of the quotation rely on estimation and are not systematic. We must develop a practical and systematic model or method to solve this problem. However, we require more knowledge. The knowledge problems originate from the action problem. We need to understand relationships between variables that affect the quotations to improve quotation estimation. We will investigate relevant applications of similar methods, subsequently, we aim to combine these methods into a tailored solution. In the context of our research, we investigate methods that find and use the relationships between features and labour cost.

The Managerial Problem Solving Method (MPSM) by Heerkens and Winden (2017) mainly focusses on action problems, though, it recognises that knowledge problems are unavoidable. The research takes a sidestep into the research cycle whenever knowledge is needed while problem solving. Once our research cycle is completed and we acquired new information, we re-enter into the MPSM at the phase that was interrupted by the knowledge problem. Although we do not fully apply the MPSM, this principle of diverging into the research cycle for knowledge problems does apply. Therefore, we illustrate our approach and the diversions from action problems to knowledge problems, in Figure 2.3.

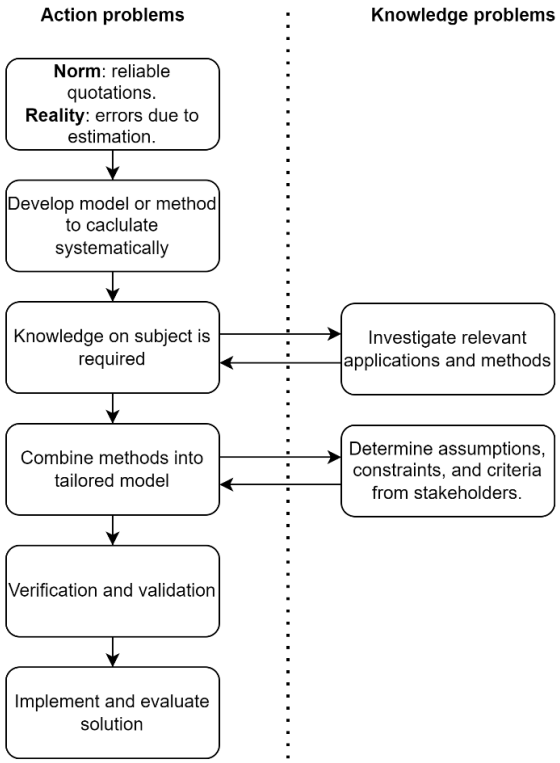



***Figure 2.3: Problem analysis.***

Although we exclude implementation in the scope of our research, we included it in Figure 2.3 to provide context for the next steps. Moreover, gathering the exact specification required for the model represented by the second knowledge problem; "*Determine assumptions, constraints, and criteria from stakeholders*". The node "*Verification and validation*" refers to testing the model and comparing result to the norm.

Now we identified the core problem and formally stated the action problems and knowledge problems, the subsequent step is to turn the problems statements into practical research questions.

## *2.5 Research Questions*

In this section, we translate the action problems and knowledge problems established in the problem analysis into research questions. Research questions are, by nature, more relevant to the knowledge problems. However, answering research questions provides the information required to solve the action problem. The action problem aims to develop a labour cost behaviour model that can systematically provide accurate quotations. Building on that, Figure 2.3 in the previous section illustrated the conceptual steps of action problems, occasionally sidestepping into a knowledge problem. We formulated the central objective based on the context and the problem analysis as follows.

*To develop an algorithm that predicts labour costs of CTOs with the actual labour cost falling within the 95% confidence interval of the predictions.*

Keeping the objective in mind and consulting the steps in Figure 2.3, the main research question and sub-questions can be formulated. The main knowledge gap in this stadium lies in the identification of relevant methods and the criteria that the model must meet. Therefore, we divide the research questions into multiple subsections. Based on the context, we formulate following research question and sub-questions.

**RQ:** *How can the firm accurately and systematically predict labour costs for configurated products?*

Literature and related works:

**RQ 1.1:** Which methods do researchers in literature apply to solve similar problems and what are the conditions of each method?

Data collection and processing:

**RQ 2.1:** What is a suitable amount of data for our approach?

**RQ 2.2:** What is a suitable data expansion method in case the amount of data is insufficient?

**RQ 2.3:** How is performance of predictions evaluated?

**RQ 2.4:** What feature engineering approach is appropriate?

Model design:

**RQ 3.1:** What is an appropriate programming language for this application?

**RQ 3.2:** How do we select underlying models in our algorithm?

**RQ 3.3:** How does the model detect and manage outliers?

**RQ 3.4:** How does the model track labour cost predictions?

**RQ 3.5:** How does the model reduce the risk of overfitting?

Model performance:

**RQ 4.1:** What is the relative feature importance?

**RQ 4.2:** What accuracy improvement can we achieve with our algorithm?

## 2.6 Scope and Limitations

We explain the extent to which we explore the research area in the scope and limitations of this study. In other words, we describe what fields we *include* and *exclude* from our study. We define subjects we research within this study as within *scope*, while we classify specifically excluded subjects as *limitations*.

Our objective relates to improving accuracy of the quotations by calculating labour cost systematically. We conducted our research from February 5, 2024, until June 14, 2024. We assume that the samples measured over time are representative for the products. We neglect the effect of increased employee experience impact on the labour costs over the relatively short time interval of our dataset. Additionally, we assume that every employee performs equally.

We define 'Improving accuracy' as predicting a value closer to the real value than the current estimation approach. Ultimately, calculating a labour cost where the true value lies within the 95% confidence interval of the prediction. As we specified earlier, the model developed in this research focusses on calculating labour cost, specifically, labour costs from the manufacturing process of CTOs. MTO and ETO are explicitly excluded from the scope because the production processes of these products are not eligible for the model we aim to develop, refer to Section 2.1 for an elaboration on the differences in quotation costing processes between product ranges.

The complexity lies in the estimation of labour hours in manufacturing processes; therefore, labour hours are the focus of the study. The firm approaches other labour and operations costs differently and therefore we disregard those costs from our research. The firm approaches staff hours as overhead, and the engineering department track their own specific costs itself. Furthermore, this also counts for internal transportation (i.e., forklifts), as the firms makes no distinction between the internal transportation of products. We also exclude integration into the ERP, training employees, and overall implementation in practice are from the scope. Our main focus is the development of the model, although, properties necessary for implementation are taken into account. Due to practical reasons in the context of this study, it is more relevant to describe *what* behaviour occurs in the labour cost than the explanation *why* it occurs, however, the why is not excluded.

Data collection is beyond our control in this study. The availability of data is limited to the xml file 'stuklijstregels' (Bill of Materials) and the Excel file 'bewerkingstijden' (processing times) from July 18, 2023, until March 14, 2024. Furthermore, the data collection for the dataset was established prior to this research by an unknown method. Therefore, we must assume reliability and reproducibility of the data collection methods.

# 3

# LITERATURE REVIEW

The literature review chapter contains a framework of relevant literature surrounding labour cost calculation for production processes. This framework serves as foundation for the research design and methodology. Furthermore, with research conducted in the literature review, we showed the discrepancy between the existing literature and the objective. We used the systematic literature review method in the development of this framework, of which we attached a detailed description in Appendix A.

## *3.1 Related Works*

The literature we consulted, covered various problems encountered in pricing and scheduling make-to-order products. Different methods are known to determine the cost statement (material cost method, kilo-cost method, division costing, equivalence costing, similarity costing, surcharge costing, and target pricing). These costing methods mainly depend on estimation, and therefore lack precision (Berwing et al., 2022). Overhead costing is, amongst the previously stated, by far the most frequently applied method (Schuh & Schmidt, 2014). However, due to the lack of precision and focus on low variety products, the methods stated are not suitable for CTO and ETO products. Literature covers a number of related examples in which a model is developed to predict a quotation cost accurately, for products that have not been produced before.

Asaolu and Nassar (2007) defined cost behaviour as the study of the ways in which costs vary with the amount of labour practiced. Drury (2013) describes costs as expenses consumed in the process of generating revenue. Profit is defined as the excess of revenue as the cost is deducted (Oluwagbemiga et al., 2014). The distinction between fixed and variable costs (also known as direct and indirect expenses) is justified in describing the reaction of profit to activity levels in an organization. Fixed costs remain constant for a given period of time, despite changes in related level of activity (Horngren et al., 2010). However, over a sufficient amount of time, virtually all fixed costs turn variable (Hansen & Mowen, 2007). Attempting to solve problems in traditional cost management systems, Kaplan & Anderson (2007) published the activity-based costing (ABC) method. The authors designed this method to allocate costs to each product, if and only if, manufacturing this product required that activity. In case of overhead costs, this means that a fixed percentage is no longer the case. Activity based costing accumulates all costs associated with the production process required to produce the output (Cokins, 2002).

Denkena et al. (2009), developed a rule-based quotation costing system for pressure die casting moulds. The researchers developed a calculation system that integrated the experiences of manufacturers in rules and subsequently made an optimised calculation possible. An automated generation of a bill of materials and the corresponding process plans were realised with the help of this model. The costs could then be determined according to the specified amount of work of each operation. Therefore, the authors developed a model for determining an accurate quotation cost by analysing past data.

Chienwichai et al. (2016), developed a process-based costing model which estimated the production costs of a gas-induced semisolid process. The described model is based on

three factors: cycle time, rate of waste, and die life. These factors were found to affect the cycle time, hence, affecting the unit production costs.

Lan & Ding (2007) developed an algorithm that predicts build-time of stereolithography parts to implement accurate quotations. The model incorporates geometrical features and support structures through a statistical method. The authors compared two quotation approaches, which include rough quotation based on weight and precise quotation based on build-time. The authors developed an algorithm to predict build-time, which incorporates the geometrical features drawn from the product design. The result turned to a web-based automated quotation system that provides an accurate price quotation instantly.

Shehab & Abdalla (2002) propose an intelligent knowledge-based product cost modelling method. This system does not require detailed design input; therefore, it enabled application in an early design stage, hence, reducing cost and lead times. Furthermore, this model has the capability of selecting materials, machining processes, and parameters based on a set of design and production parameters. Additionally, through the capabilities of this model, the product cost could be estimated throughout the entire product development cycle, including assembly costs.

Furthermore, Cavalieri et al. (2004) conducted research comparing the results of two cost estimation approaches: statistical methods and artificial neural network (ANN) techniques, respectively. Cavalieri et al. (2004) specifically investigated the cost estimation of the unitary manufacturing costs for brake discs in the automotive industry. The authors confirmed the validity of the ANN model, although it did not display clear superiority with respect to the statistical approach. The ANN model was characterised by a better trade-off between precision and accuracy. However, this advantage comes with a reduced transparency of interpreting output data.

## *3.2 Cost Estimation Methods*

Estimating cost increases in complexity as more factors get involved. Especially when more products are manufactured, and the marginal costs decrease. For example, the division of overhead costs and the decrease of engineering costs. Ruffo et al. (2006) and Lan and Ding (2007) developed compact frameworks of approaches to estimate costs. The principal quantitative methods to estimate costs in manufacturing are:

1) Analogy-based techniques: This approach is based on a derivation of an estimation from actual data about similar products.
2) Statistical models: This approach expresses costs as an analytical function of a set of variables, typically referred to as *cost-estimation relationships* (CERs).
3) Engineering approaches: The estimated cost is analytically calculated as an aggregate of its elementary components, represented by the cost of the resources used in each step of the process. An accurately defined process is required for this approach.
4) Machine learning: Model that identifies relationships, dependencies, and cause-effect relationships between different design solutions, based on classic cost/benefit ratio and applies regression techniques. A limitation of this method is that some of the relationships found cannot be logically argued, in other words, transparency is limited (Cavalieri et al., 2004).

The first category, i.e., analogy-based techniques, relates most to Meilink's current approach of cost estimation and are shown to be non-systematic (Lan & Ding, 2007). Hence, analogy-based techniques are sensitive to error and lack reproducibility. The third method, i.e., the engineering approach, relies on the decomposition of work and resources required to complete the product. This method lacks the ability to estimate labour cost due to the unpredictable nature of labour cost in the context of our research. The statistical method aims to evaluate the product cost from characterising the product without describing it completely. In other words, it develops statistical relationships between the features and the price of previous products. Machine learning effectively operates similar to the statistical approach in an automated fashion at the cost of a reduced transparency. However, a single method might not be sufficient in practice. A model that combines several of the methods synthetically, might outperform application of a single method (Layer et al., 2002). We aim to investigate statistical methods, multi-parameter approaches, and machine learning methods in the following sections.

## *3.3 Multi-Parametric Modelling*

Experiments in manufacturing organisations are often organised as a series of trials or tests which yield quantitative results. For instance, to test the impact of adjustments on the efficiency of the process. These experiments aim to explore, estimate, or confirm. For manufacturing processes, it is often important to find relationships between input factors. Not all input parameters influence the results equally, some parameters have a strong relationship, and some weak or not at all. One of the most common approaches is the One-Factor-At-a-Time (OFAT), where the effect of changing a single input variable is measured. This approach is time consuming, requires a lot of resources and high costs are often associated with experimenting on a manufacturing line. Hence, OFAT reveals a limited amount of information and requires a relatively large investment (Antony, 2023).

An effective method that handles multiple input variables is Design of Experiments (DOE). This method plans, designs, and analyses so that valid and objective conclusions can be drawn. It integrates statistical methods into the experimental design. What distinguishes DOE from OFAT is the possibility to test the response of multiple variables in fewer runs. Additionally, DOE considers test runs in which multiple factors are changed simultaneously, therefore, it can analyse combined effects and identify interactions and dependent variables. OFAT is unable to identify interactions between parameters. DOE runs multiple tests in a randomised order, preferably with replication of runs. The number of replications is generally determined 'with degrees of freedom', in other words, the number of independent and fair comparisons that can be made in a dataset. Statistics in the context of DOE claim that the degree of freedom related to a process variable is one less than the number of levels a factor can have. Consider an experiment where output is observed for three different temperature levels. The number of degrees of freedom for this example is two. If an experiment is conducted in eight trials and each trial condition is replicated, the number of observations is 16. Therefore, the total degrees of freedom is 15 (i.e., $16 - 1 = 15$) (Franceschini & Macchietto, 2008). To conclude, DOE is a dynamic, statistical, and mathematical method. Applying DOE to a manufacturing process leads to a mathematical representation of measured output behaviour at different input factor levels. Furthermore, it allows to process more input variation in less time than the traditional OFAT. Therefore, the principle of DOE is useful for the problem addressed in this research.

## *3.4 Machine Learning*

Machine learning (ML) is generally useful to handle data more efficiently than traditional approaches (Mahesh, 2020). In some cases, machines are more efficient in extracting information from data. The purpose is to learn from the data and subsequently make decisions based on that knowledge. In other words, instead of prompting a computer to do a repetitive task, it learns from experience (Harrington, 2012). Once the algorithm knows what it has to do and how, it can execute tasks autonomously. Machine learning algorithms can be developed in many common programming languages. The popularity of *Python* in data analytics increases, partly due to its increasing availability of libraries (Hao & Ho, 2019; Lee, 2019; Srinath, 2017). Other commonly used programming languages for machine learning are: *R*, *Javascript*, and *C++*.

Mahesh (2020) developed a framework of commonly used machine learning algorithms, reviewing the fundamentals of each type and its applications. We illustrate the subsections of machine learning in Figure 3.1, most of which contain multiple variants. Each of the discussed subcategories of require different conditions and perform better in specific applications. The following sections provide a brief review of the different methods,

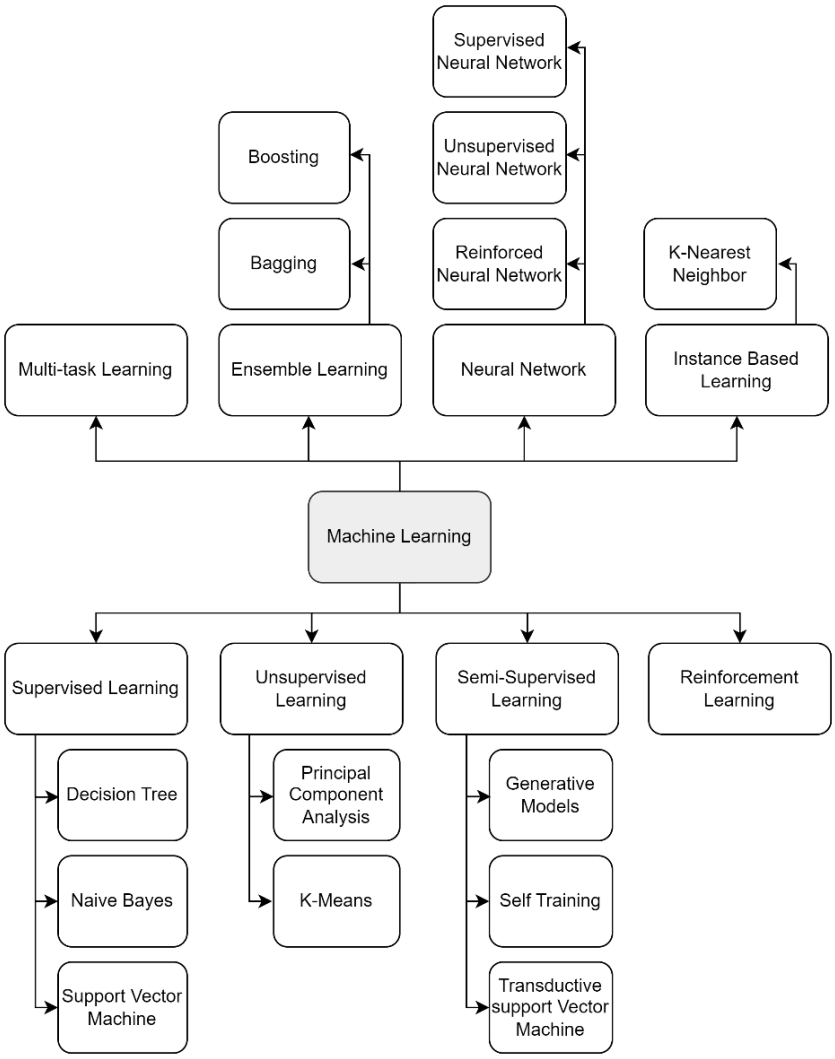



***Figure 3.1: Commonly used types of machine learning.***

An understanding of terminology in machine learning is useful to explain the process of each of the types, and ultimately, to select a suitable method. The terminology is as follows:

regression problems aim to find a number and classification problems aim to classify or categorise (Loh, 2011). *Records* (also referred to as *instances* or *samples*) collectively form a dataset where each record holds a description of an event (e.g., length, width, height) called *attributes* or *features*. We refer to their specific values as *feature values*. The space or range a value can take is the *feature space*, *sample spac*e, or *input space*. The outcome of a record is referred to as a *label*, and a sample with a label is called an *example*. The ability to work on new samples is called the *generalization ability*. Since in the optimal situation, the model works with the whole sample space. Although the training dataset is usually a small proportion of the sample space, it is desired that it reflects the characteristics of the entire sample space, to some extent (Carleo et al., 2019; Zhou, 2021). Furthermore, we divide the learning process into two categories: *supervised* and *unsupervised* learning. Supervised refers to presence clustering or labelling in the training data. A machine can distinguish between factors more easily if the data are labelled in different categories. Unsupervised refers to the absence of underlying labels and therefore the machine is required to use automated methods or algorithms on the data that have not been classified or categorised. In other words, the algorithm learns underlying relationships from available data (Alloghani et al., 2020).

In the following section, we provide a brief review of the main categories of machine learning (Dangeti, 2017; Kotsiantis et al., 2007; Mahesh, 2020; Zhou, 2021). Since there are multiple types of machine learning, we make a preliminary selection to focus on the relevant methods based on the conditions of this research.

The three main machine learning categories are: **1)** *Supervised learning* connects an input to an output based on examples it learns from. This type of machine learning applies a function obtained through the labelled training data. Furthermore, this type of machine learning requires external assistance. **2)** *Unsupervised machine learning* requires no external assistance. The algorithm is left to its own devices to discover the structure and relationships in the data. It only learns some relationships from the data, when new data are introduced, it uses the previously learned knowledge to classify the new data. **3)** *Semi-supervised learning* is a combination of supervised and unsupervised learning methods. It is efficient in applications where unlabelled data are present and labelling the data is too complex. This method trains the algorithm with labelled data, where each record contains the outcome information.

Machine learning techniques can usually operate under any of the categories**.** *Reinforcement learning* continuously learns through interactions with its environment and adjusts according to the response. *Multitask learning* is a type of machine learning that aims to solve multiple, different tasks simultaneously. By accounting for the similarities between different tasks, it can improve learning efficiency and act as a regulator. *Ensemble learning* applies a process where a combination of multiple models is applied to solve a computational problem, generally to improve performance of a model. *Neural networks* contain a series of algorithms that aim to identify relationships in a dataset, this mimics the fashion a human brain works. Neural networks adapt when an input changes; it generates the optimal result without requiring redesigning the output criteria. This method can be applied in supervised, unsupervised, and reinforced. *Instance based learning* refers to a group of methods for classification and regression that generate a prediction based on similarity of its nearest neighbour(s) in the training dataset.

Regarding the context and the data available, the most relevant category of machine learning is supervised learning. We elaborate and describe different techniques in this category in the following sections.

## *3.5 Supervised Machine Learning*

Supervised machine learning refers to the presence of labels in the training data. In other words, the outcome of the records is included in the data. This method maps input-output relationships and applies this knowledge to calculate outputs for new (unlabelled) inputs. The process of learning a set of rules from instances (examples) is referred to as *inductive learning*. A classifier is created that can be used to generalise new instances (Kotsiantis et al., 2007). Figure 3.2 illustrates the schematic process of supervised machine learning, where a dataset is divided in training data and test data. The model is trained with the training data and subsequently tested with the test data.

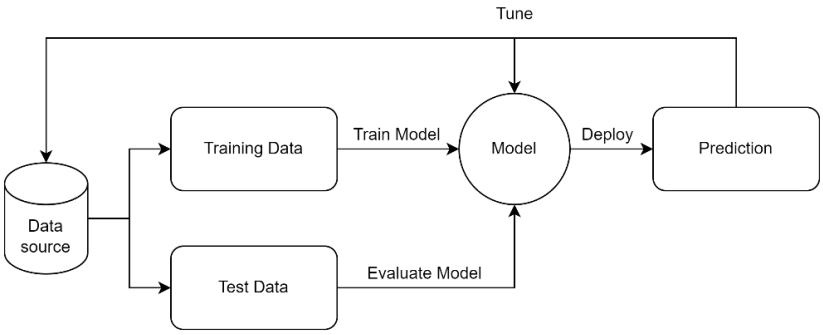



***Figure 3.2: Schematic representation of supervised machine learning.***

Consider a three-feature record as three axes, spanning a three-dimensional space that describes the product. For example, a product with varying length, width, and height, where the time to build this product is the label. Since every combination of the three feature levels can be positioned in this space, every point in the space corresponds to a position vector, called a *feature vector*. Let $D = \{x_1, x_2, \dots, x_m\}$ be a dataset containing $m$ instances, where each instance is described by $d$ features. In this case, three features are used to describe the product. Each instance $x_i = (x_{i1}; x_{i2}; \dots; x_{id}) \in \chi$ is a vector in the $d$-dimensional sample space $\chi$, where $d$ is called the *dimensionality* of the instance $x_i$.

The outcome information should also be available, in our case, labour cost of the product. This outcome is referred to as a *label*, and a sample with a label is called an *example*. Generally, the $i$th sample can be written as $(x_i, y_i)$, where $y_i \in \Upsilon$ is the label of the sample $x_i$, and $\Upsilon$ is the set of all labels, also referred to as *label space* or *output space*. If the prediction output is discrete (e.g., true or false) it is called a classification problem. In this example, the prediction output is continuous, the labour time, it is called a regression problem (Gonzalez-Carrasco et al., 2012; Zhou, 2021).

Referring to Figure 3.1, the supervised machine learning principle hosts multiple sub-methods. We explain the underlying principles of the main methods that can be used for supervised machine learning: Support Vector Machines (SVM), Linear Regression (LR), Random Forest Regressor (RGR), and Artificial Neural Networks (ANN)

### 3.5.1 Support Vector Machines

Classification problems and regression problems are both candidates for SVM. Mountrakis et al. (2011) describe these models as non-parametric statistical learning models, which make no assumption on the underlying data distribution. The method uses a labelled dataset to find a hyperplane in the training phase (Anguita et al., 2012). The hyperplane is chosen by the algorithm in such a way that the distance between the hyperplane and the nearest data points (called *support vectors*), is maximised (Meyer & Wien, 2001; Noble, 2006).

### 3.5.2 Linear Regression

Linear regression is especially suitable for predicting continuous variables based on a labelled dataset (James et al., 2023). LR models the relationship between input variables (referred to as independent variables) and the target variable (referred to as dependent variable) by using a linear equation. The goal of the training phase is to learn the values of the parameters that minimise the error between the predicted and the actual values. Typically, the algorithm achieves an optimum by minimising a loss function like Mean Squared Error (MSE) or Mean Absolute Error (MAE). Once the model is trained and the optimal parameters are known, it can be applied to make predictions on new, unseen data (Fox, 2019; Uyanık & Güler, 2013).

### 3.5.3 Random Forest Regressor

This technique is suitable for regression problems and belongs to the family of decision tree-based models. However, it extends on traditional decision trees by building multiple trees and combining their predictions. Each tree in the combination is trained on a random subset of the training data. RFR aggregates the individual predictions but reduces overfitting by taking the mean of median of each of the predictions (Segal, 2004; Sekhar & Madhu, 2016).

### 3.5.4 Artificial Neural Networks

Over the last years, ANNs have grown in popularity (Gurney, 2018; Tkáč & Verner, 2016). This principle is inspired by how a human brain functions, which is generally represented as a network of interconnected neurons (Yegnanarayana, 2009). The connections between neurons are the synapses. This network of connections stores the knowledge in a distributed fashion. Figure 3.3 illustrates the network of interconnected elements. Another similarity between ANNs and the human brain is the learning approach. Both the human brain and ANNs need to train to obtain knowledge. Generally, an ANN is trained by means of “training data”, to identify a set of patterns. This set of patterns represents the experience gained by the ANN to recognise in future application (Basheer & Hajmeer, 2000). Hence, an ANN has the ability to infer from the knowledge to answer new inputs that have not been presented to the ANN before (Abiodun et al., 2018; Agatonovic-Kustrin & Beresford, 2000; Graupe, 2013).

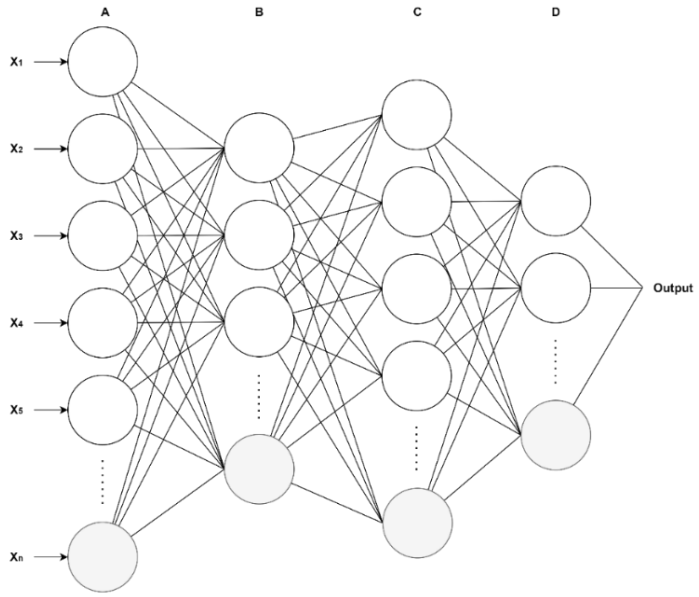



***Figure 3.3: Schematic representation of an ANN with $n$ inputs and one output.***

Multi-Layer Perceptron is a type of supervised ANN that is suitable for regression problems. The algorithm is provided with data and feedforward propagation predicts the outputs by passing activation through the layers of neurons (Park & Lek, 2016; Popescu et al., 2009). Backpropagation then adjusts the weights and connections based on the calculated error between the prediction and the true value. The aim of the training phase is to minimise a specific loss function (Costa et al., 2023). The model can make predictions after completing the training phase and the network of neurons is configurated.

## *3.6 Data Requirements*

The amount of data available impacts the accuracy and reliability of a machine learning algorithm (L'heureux et al., 2017). Training the model is a process of tuning its internal parameters that aims for a balance between training accuracy and regularity. Proper training inhibits the effect of overfitting as well as underfitting (Ying, 2019). Therefore, the model requires sufficient training samples to reach this balance and the sample size is proportional to the number of variables. Huang et al. (2002) found that the number of available samples has a larger impact on the accuracy than the type of machine learning algorithm that is applied. The general rule of thumb for a number of training samples in machine learning is at least 10 times the number of features (Maxwell et al., 2018). Additionally, regardless of the variable range, the dataset is split into training data, validation data, and test data in a 50-25-25 ratio, respectively, when hyperparameter tuning is involved (Dangeti, 2017). Common practice is an 80-20 ratio of training data and test data (Joseph & Vakayil, 2022; Muraina, 2022; Nguyen et al., 2021). The amount of data required for a statistical approach can be derived from Equation 1 at the preferred confidence level, assuming a normal distribution.

$$[95\%]CI = \bar{x} \mp z \cdot \left(\frac{\sigma}{\sqrt{n}}\right) \quad (1)$$

Where:

$\bar{x}$: sample mean

$z$: z-score corresponding to the confidence level

$n$: sample size
$\sigma$: sample standard deviation

The $z$-score for a 95% interval is 1.96 and the margin of error for a confidence interval for a population mean can be calculated with Equation 2:

$$E = \frac{z \cdot \sigma}{\sqrt{n}} \tag{2}$$

Subsequently obtain sample size $n$ by rearranging Equation 2:

$$n = \left(\frac{z \cdot \sigma}{E}\right)^2$$

The following methods can be applied in case the amount of available data to train a model is limited (Karystinos & Pados, 2000; Sun et al., 2014; Yip & Gerstein, 2009):

1) Acquisition of additional data.
2) Addition of random noise to the existing dataset.
3) Re-acquire information from existing dataset.
4) Generate new data based on distributions observed in the existing dataset.

Karystinos and Pados (2000) conducted research regarding overfitting, generalization, and expanding datasets. Overfitting occurs when the dataset is too small, and the objects do not have enough information to form local models. An infinite sequence of artificial input-output vectors was created to combat overfitting. The initial approach was to add (*white Gaussian*) noise to the training set. The addition of noise refers to generating a set of standard-normal distributed ($\mu = 0, \sigma = 1$) values and adding it to the input of the model. Noise is only added during training, it is excluded from the evaluation and when the model is actually making predictions. Additionally, the authors applied a statistical method that expands the training data by observing the distribution and generating data that fits that specific distribution.

Yip and Gerstein (2009) propose a concept of training set expansion named *Prediction Propagation* (PP). The method effectively re-acquires data from the existing dataset by generating auxiliary training examples. In other words, it allows the model to re-learn from *information-rich* regions.

It has become clear what criteria our data must meet. The rule of thumb for machine learning approaches is that the sample size should be at least ten times the number of features. The amount of data for a statistical approach can be derived from the confidence interval equation. In case the data amount is insufficient, it can be reused from information rich regions.

## *3.7 Feature Engineering*

Feature Engineering (FE) directly affects the performance of the model by modifying shape, distribution, or size the dataset it works with. This technique assists in creating new features of transforming old ones to improve the model's ability to learn and generalise the data (Khalid et al., 2014; Miotto et al., 2018). For example, a certain model might struggle with a heavy tailed dataset. Reducing the probability mass in the high end and converting it to a

longer tail can improve the readability for the model. The top figure in Figure 3.4 resembles the original distribution with a heavy probability mass near zero. The bottom figure resembles the same dataset after FE transformation.

Many methods belong to FE. Feature Selection (FS) is a data preprocessing strategy to select relevant features from the dataset (Li et al., 2017). This way, most relevant information is preserved in the dataset. Other scaling methods are min-max scaling, standardization, log scaling and z-score normalisation (Ambarwari et al., 2020). Imputation of missing values implies filling the missing values in the dataset with either the mean, median, mode or other advanced methods like KNN Imputation (Donders et al., 2006). Alternative methods like One-Hot Encoding and Label Encoding represent categorical variables as binary vector or as integers, respectively (Rodríguez et al., 2018; Yang et al., 2021).

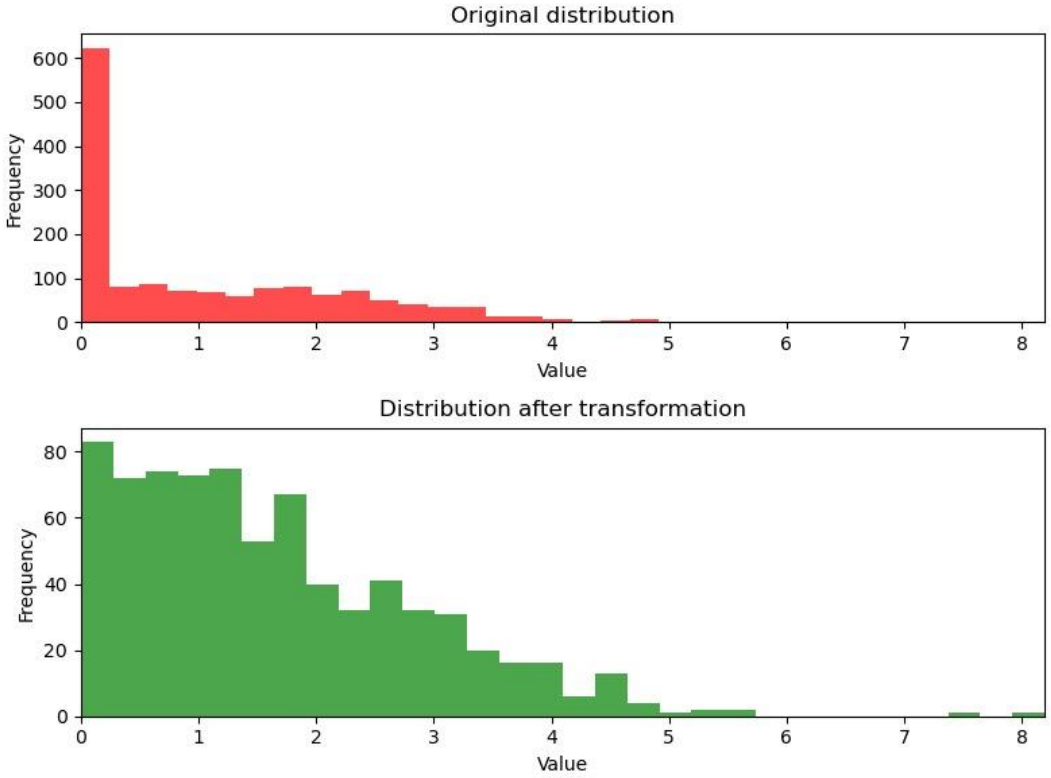



***Figure 3.4: Distribution before (top) and after transformation (bottom).***

Feature engineering assists in improving the algorithm's interpretation of our dataset, for instance, by eliminating a heavy-tailed distribution by normalisation, and engineering new features the algorithm can extract information from.

## *3.8 Performance Metrics*

Evaluating the performance and fit of a model or method requires performance metrics. Evaluating a classification problem is straightforward, it can be expressed in a percentage of accuracy. A regression problem, however, requires greater consideration (Flach, 2019; Rodriguez-Galiano et al., 2015). Chicco et al. (2021) state that no consensus has been reached on a unified performance metric for regression problems. We elaborate a number of performance metrics that are used in literature to evaluate the performance of models.

### 3.8.1 Akaike Information Criterion

The metric AIC is generally used to compare the fit or performance of different regression models (Bonakdari & Zeynoddin, 2022; Cavanaugh & Neath, 2019; Li et al., 2020; Oshan et al., 2019). The AIC rewards a level of fit where the lowest value represents the best fit. We calculate AIC with Equation 3:

$$AIC_i = 2k_i - 2\ln(\mathcal{L}_i) \tag{3}$$

Where:
$\mathcal{L}_i$: maximised value of the likelihood function of model $i$
$k_i$: number of estimated parameters of model $i$

$$AIC_i = 2\ln\left(\frac{e^{k_i}}{\widehat{\mathcal{L}_\iota}}\right)$$

The *maximum likelihood estimation* (MLE) is obtained through the log-likelihood function. Let the set $Y_1, \dots, Y_n$ be $n$ independent random variables (RVs) with probability density function (PDF) $f_i(y_i; \boldsymbol{\theta})$ that depends on the vector parameter $\boldsymbol{\theta}$. We denote the *likelihood function* $\widehat{\mathcal{L}_\iota}$ as $\mathcal{L}(\boldsymbol{\theta}; \boldsymbol{y})$, for the unknown parameter $\boldsymbol{\theta}$ given the data $\boldsymbol{y} = (y_1 \dots y_n)'$ for $n$ independent observations is:

$$f(\boldsymbol{y}; \boldsymbol{\theta}) = \prod_{i=1}^{n} f_i(y_i; \boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{\theta}; \boldsymbol{y}) =$$

The log-likelihood function is formulated as follows:

$$\log \mathcal{L}\,(\boldsymbol{\theta}; \boldsymbol{y}) = \sum_{i=1}^{n} \log f_i(y_i; \boldsymbol{\theta})$$

Maximization of the likelihood function (or equivalently the log-likelihood function) to estimate the parameter $\boldsymbol{\theta}$ given the data $\boldsymbol{y}$, works by choosing the parameter value that makes the observed data as likely as possible. The maximum likelihood estimator is denoted as $\widehat{\boldsymbol{\theta}}$.

### 3.8.2 R-squared

The *Coefficient of Determination* or *R-squared* expresses the proportion of variation in the dependent variable that is predictable from the independent variable(s) (Nakagawa et al., 2017). Chicco et al. (2021) claim that the R-squared is more informative than any of the other treated metrics to assess regression analysis performance. The R-squared can take values in the range $(-\infty, 1]$, a value closer to 1 indicates a stronger relationship and a value closer to 0 indicates no relationship between the variables (Asuero et al., 2006; Chicco et al., 2021). We calculate R-squared with Equation 4:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(x_i - y_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{4}$$

Where:

$n$: sample size

$x_i$: $i$th observed value

$y_i$: $i$th predicted value

$\bar{y}$: mean of the predicted values

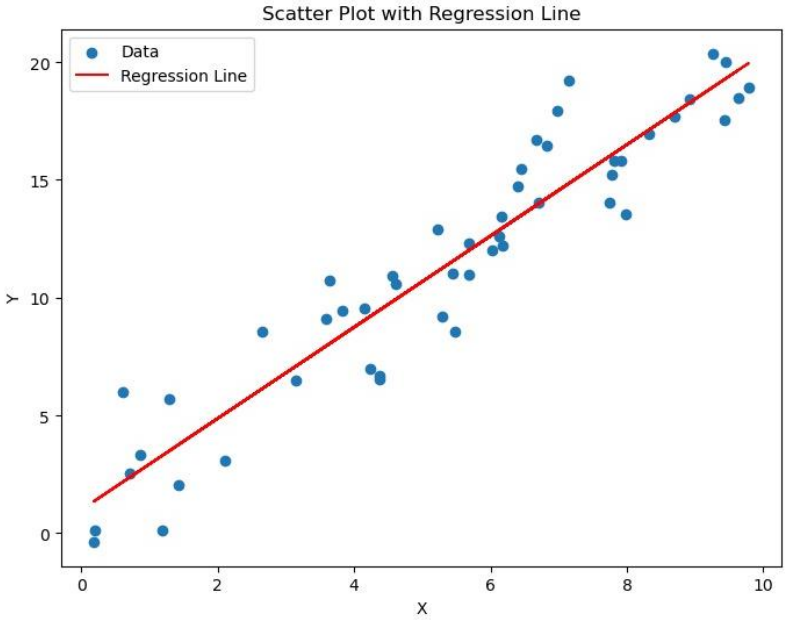



***Figure 3.5: Example of $R^2$ in a scatterplot.***

The scatterplot in Figure 3.5 plots fifty data pairs along with a regression line. The $R^2$ describes a proportion of the independent variable's ability to predict the dependent variable.

### *3.8.3 Mean Squared Error*

Mean Squared error (MSE) or *Brier score* is a performance metric which decomposes into *calibration loss* and *refinement loss* (Flach, 2019). The obtained value of MSE depends on the unit of the predicted variable and lies in the interval $[0, \infty)$ (Gupta et al., 2009) and is calculated with Equation 5:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2 \tag{5}$$

Where:

$n$: the sample size

$x_i$: the $i$th observed value

$y_i$: the $i$th predicted value

Root Mean Squared Error (RMSE) and MSE are closely related through the square root. Evaluation results of models based on MSE are generally equal to an evaluation based on RMSE, calculated with Equation 6:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2} \quad (6)$$

Where:
$n$: the sample size
$x_i$: the $i$th observed value
$y_i$: the $i$th predicted value

Chicco et al. (2021) state that the MSE is more suitable when outliers need to be detected because it can attribute greater weights to outliers. Since $R^2 = 1 - \frac{MSE}{MST}$ and the Mean Total Sum of Squares (MST) is fixed for the dataset, R-squared is (negatively) linearly related to MSE. We calculate MST with Equation 7:

$$MST = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2 \quad (7)$$

with

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

Where:
$n$: sample size
$y_i$: $i$th predicted value
$\bar{y}$: mean of the predicted values

### *3.8.4 Pearson Correlation Coefficient*

The Pearson correlation coefficient (Equation 8) for linear correlation $r$ is another way to express the strength and direction of a linear relationship (Asuero et al., 2006; Zou et al., 2003). The range of $\rho$ is between -1 and 1, where a negative value shows a negative correlation, and a positive value indicates a positive correlation. Closer to either -1 or 1 shows a stronger relationship, and 0 indicates no correlation (Cohen et al., 2009).

$$\rho = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} \quad (8)$$

Where:
$n$: sample size
$x_i$: the $i$th observed value
$y_i$: the $i$th predicted value
$\bar{x}$: sample mean of $x$
$\bar{y}$: sample mean of $y$

## *3.9 Intermediate Conclusions*

**RQ 1.1:** *Which methods do researchers in literature apply to solve similar problems and what are the conditions of each method?* The reviewed literature covered several approaches to calculating costs in a multi-parametric context. Four prominent cost estimation methods in literature are: analogy, statistical, engineering, and machine learning. Analogy-based techniques are similar to the current approach of an estimation based on a comparison to similar products. The engineering approach calculates the cost from its elementary components and does not work for labour costs. Statistical cost calculation evaluates the cost from a statistical relationship between the features and the price of historical products. Machine learning applies the same principle but automated and the ability to learn. We discussed the subcategories of machine learning, from which, we select the following techniques for evaluation: *Linear Regression* (LR), Multi-Layer Perceptron (MLP), *Gaussian Process Regression* (GPR), *Random Forest Regression* (RGR), *Support Vector Machines* (SVM), *Decision Tree Regression* (DTR), *Gradient Boosting Regression* (GBR), *K-Nearest Neighbours* (KNN), and Extreme Gradient Boosting (XGB).

**RQ 2.1:** *What is a suitable amount of data for our approach?* The general rule of thumb in machine learning for data amounts is that the dataset should be at least ten times the number of variables, split into training data and test data in an 80-20 ratio, respectively. Statistical methods derive minimal sample size from the normal distributed confidence interval equation.

**RQ 2.3:** *How is performance of predictions evaluated?* We can evaluate the performance and fit of models with MSE, RMSE, R-squared, AIC and PCC. A positive value closer zero indicates a better fit for MSE, RMSE, and AIC. R-squared can take values up to one, where one is a perfect fit.

**RQ 2.4:** *What feature engineering approach is appropriate?* Based in the literature we consulted, we propose normalising the data with min-max normalisation. Eliminating a heavy tailed distribution improves the readability of the dataset for the model. We use the most important features of each product to engineer new features, we discuss the method in Appendix B.

**RQ 3.1:** *What is an appropriate programming language for this application?* Machine learning algorithms and statistic methods can be developed in many common programming languages, we select python due to its availability of libraries.

# 4

# METHODOLOGY

In this chapter, we describe the methods applied during our research and justify why we selected the methods. We included the process of collecting, preparing, and analysing data in Appendix B. The methodology is limited to less common methods and principles, more common methods are attached in Appendices A and C.

## *4.1 Introduction*

The firm first noticed the problem since the product range expanded to a configuration option. Customers were now able to configure an infinite number of products. The core problem brought along by the innovation was that it became too complex for staff to estimate labour cost systematically. Labour times are empirically tracked during manufacturing; therefore, a quantitative dataset of product parameters and labour times is available. By literature review we found an advantage of computational decision-making over that of a human, as well as advantages in accuracy and learning speed. This directed the objective toward the development of an algorithm:

> *"To develop an algorithm that predicts labour costs of CTOs with the actual labour cost falling within the 95% confidence interval of the predictions."*

We formulate a concrete solution approach and elaborate the decision-making process in the following sections. We describe the source and composition of the dataset in more detail, discuss model validation and outlier treatment methods.

## *4.2 Solution Approach*

The desired model and its properties can be illustrated in a *black box model*, i.e., disregarding the internal process. We designed it as follows: the available input, training data, and desired output are known, along with a dataset of specific configuration parameters and the labour times of products manufactured in the past. The model should find the relationship between levels of parameters to identify exactly what effect each parameter has on the labour cost and with what intensity. The dataset provides the model with all available information and the model calculates an educated estimate of the labour cost of a product. In other words, the model finds relations in the training-data and applies this *knowledge* to the newly presented input to estimate the output based on known relationships between parameters. Figure 4.1 represents the described black box model.

This dataset is based on products that have been manufactured in the past and is referred to as *training data*. After the model is 'trained', we present to with a new set of configuration parameters from a product that has not been manufactured (unknown labour time). The model then uses the relationships learned to calculate the estimated labour time. Labour times are tracked during manufacturing of that specific products. This information can be redirected back to the model so it can compare the estimation to the measured value and adjusted accordingly. Therefore, the model improves as more data are presented. The model determines the sensitivity of each parameter on the final labour time, and therefore the cost.

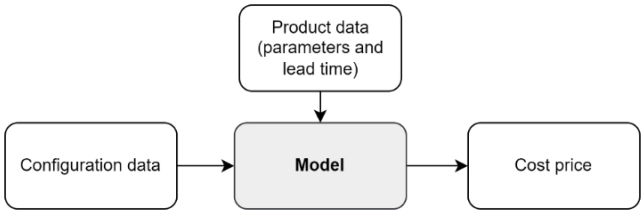



***Figure 4.1: Black box model.***

Connecting the Blackbox model in Figure 4.1 to the labour costing process in Figure 2.1, yields Figure 4.2, a process description of the labour price algorithm to be developed. This combination includes important components to the development of the model. For instance, the data recycle from the manufacturing phase, back into the model.

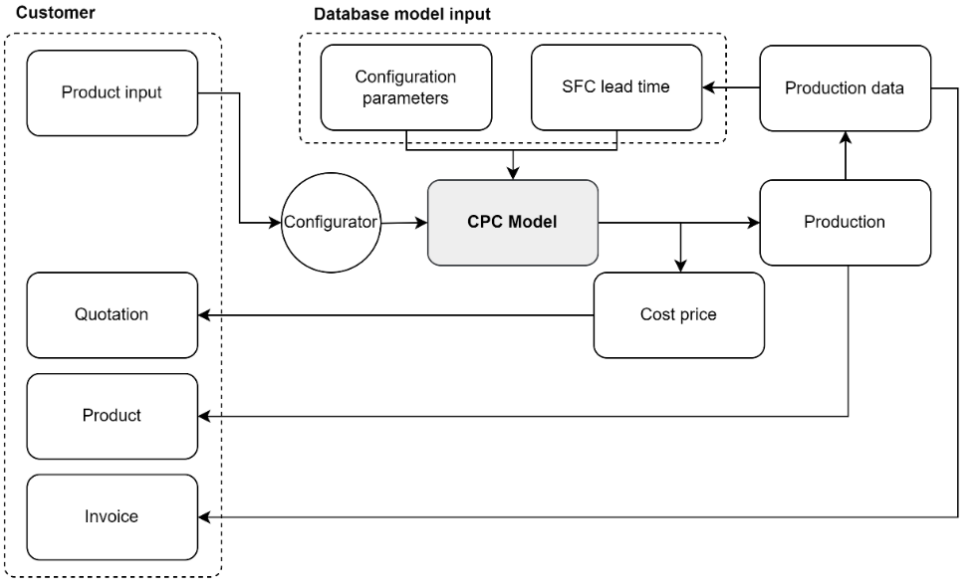



***Figure 4.2: Detailed labour costing with CPC model.***

The literature we reviewed, presented several approaches to calculating costs in a multi-parametric context. We investigate what type of data are available in the context analysis. We identified four main cost estimation methods: analogy, statistical, engineering, and machine learning, two of which are unsuitable for this research, since analogy-based techniques are similar to the current approach of an estimation based on a comparison to similar products. Engineering approaches calculate the cost from elementary components and do not prove effective for labour costs. The remaining two methods remain appropriate: statistical approach and machine learning. Statistical approaches to cost calculation evaluate the cost from a statistical relationship between the features and the price of previous products. Machine learning applies the same principle but automated with the ability to learn. However, machine learning comes with reduced transparency.

We discussed and compared eight subcategories of machine learning in the literature review. Two of which, suit the conditions of application: supervised machine learning and statistical modelling. Supervised machine learning requires a labelled dataset, which is available in our situation. Statistical modelling finds underlying relationships between variables and makes decisions based on that. Table 4.1 summarises and compares the main characteristics of the approaches (Baker et al., 2016; Dangeti, 2017; Henderson et al., 2018; Szepesvári, 2022).

***Table 4.1: Differences in statistical and machine learning methods.***

| Statistical modelling | Machine learning |
|---|---|
| Formalises relationships between variables by expressing them as mathematical equations. | It is an algorithm that learns relationships from the dataset without needing specific on rule-based programming. |
| Assuming the shape of the model curve is required before fitting the model on the dataset (e.g., linear, polynomial, exponential etc.) | No assumption of the underlying shape is required. It learns complex relationships automatically based on the dataset. |
| Multiple diagnostics of parameters are performed (e.g., p-value etc.) | Does not perform any statistical diagnostic significance tests. |
| Data are split into 70%-30% to create training and testing data. The model is developed on the training dataset and tested on the remaining data. | Data are split in 80%-20% to create training, and testing data. The model is developed on training data, hyperparameters tuning with validation dataset, and is evaluation with remaining test data. |
| Can be developed on a single (training) dataset, performance is evaluated by overall accuracy and at individual variable level. | Needs to be trained on two datasets, due to lack of diagnostics on variables (training and validation data). |
| Mostly used for research purposes. | Mostly used for practical applications. |

We can draw the conclusion which approach best the conditions suits from the gathered knowledge in relation to the conditions, context, scope, objective, and the dataset. In Chapter 2, we described the context of the research in detail. The key factors to include in the considerations of the methods are: the number of parameters to describe a configurated product and the desired accuracy level. The main takeaways from Section 2.6, are that the focus lies on the prediction of labour times and that a description of what occurs is more prevalent than why it occurs. We described methods and principles of related research in Chapter 3, including (data) conditions in Section 3.6 and the trade-offs between advantages and disadvantages. We present a more detailed comparison between the properties of the three closest contesters in Table 4.1. We explain the composition and properties of the dataset in Section 4.3.

The dataset consists of labelled records which is a condition for supervised learning, which is also able to handle complex underlying model shapes. The number of different parameters in the dataset makes it complex to perform numerically with a statistical approach, which also has the disadvantage of not being fully automated. Supervised learning comes with the disadvantage of a reduced transparency. However, in the scope we state that modelling *what* happens to the labour cost is more prominent than explaining *why* it happens. Hence, this disadvantage is insignificant.

To conclude, supervised learning outperforms statistical modelling in terms of accuracy when processing large numbers of parameters. Therefore, statistical modelling is disregarded because it becomes too complex for the number of variables considered in the dataset. Therefore, supervised learning appears to suit the conditions the most out of the reviewed

methods. The advantage of continuous learning, full automation, and complex underlying patterns, make it suitable for our context. Most importantly, the number of parameters is no obstacle for the size of the dataset.

Several methods are known in the supervised learning territory. For instance, Support Vector Machines, Linear Regression, Artificial Neural Networks, Multi-Layer Perceptron, and Random Forest Regression. One method might have a better fit on the dataset than the other. Appropriate methods can be identified by consulting the performance metrics MSE and AIC. We consider using just MSE and AIC to assess the fit of each model. These two parameters are considered sufficient because MSE is related to the other performance metrics discussed and demonstrated the relationships between R-squared, RMSE, and MSE. We select AIC because it penalizes larger numbers of parameters. We investigate further validation with K-fold cross Validation method after we assign every product to a method. During the validation phase, the most important features can be extracted from the model. Based on that information, we can use feature engineering to create new features and increase accuracy.

In addition to the performance metrics, the results also include visualisations to allow for efficient interpretations and error identification. A scatterplot of the predictions versus the actual values allows us to get an impression the accuracy. Ideally, the points in the scatterplot gather around in the shape of a 45° between the y-axis and the x-axis.

## *4.3 Dataset Description*

The result of the data preparation process, attached in Appendix B, is the definitive dataset. This definitive dataset holds the samples for six products, referred to as Product Configurations (PCFs) followed by the reference number. Separating the dataset into subsets for each PCF results in six datasets, which describe the products manufactured from July 18th, 2023, until March 14th, 2024, where each sample holds twenty-seven original features and two engineered features, linked to a unique labour time. Table 4.2 contains the names of the dependent and independent variables in the dataset.

***Table 4.2: Dependent and independent variables in the datasets.***

| **Independent variables** | | | **Dependent variable** |
|---|---|---|---|
| '10BAK' | 10KOPSCHOTTEN' | '3241' | 'GemTijd' |
| '10BEKLEDING' | 10LOSMATERIAAL' | '12401' | |
| '10BODEM' | 10MERKEN' | '_hijstrek' | |
| '10DEKSEL' | 10SJABLOON' | '_VERBINDING2' | |
| '10DEKSELFRAME' | 10STAALWERK' | '_VERBINDING3' | |
| '10HOEZEN' | 10STOPHOUT' | '_wvbBalken' | |
| '10INTERIEUR' | 10ZIJSCHOTTEN' | '_wvbPLanken' | |
| '10JUK2' | 20MONTAGE' | 'FE01' | |
| '10JUK3' | '2049' | 'FE02' | |
| '10JUKKEN' | '2085' | | |

We refer to one complete set of all the variables in Table 4.2 as a *sample*. The sample size can affect the reliability or outcome of an experiment. The sample size in each of the subsets are illustrated in Figure 4.3.

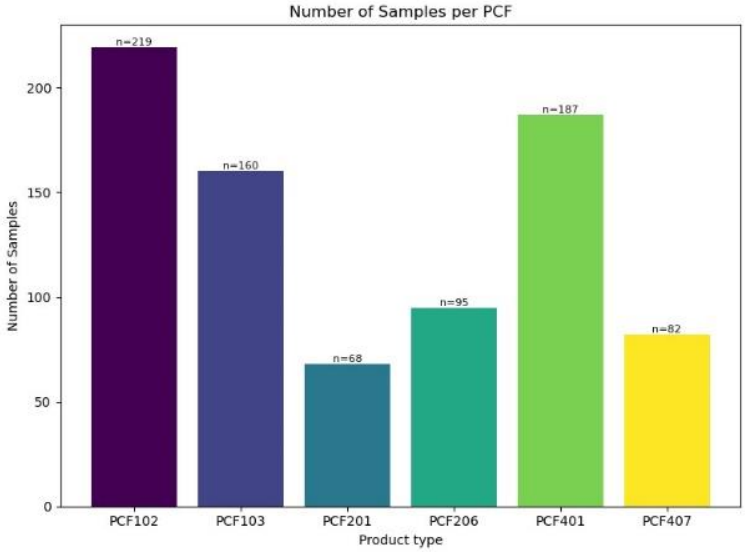



***Figure 4.3: Product types with largest sample sizes (n>50).***

Visualisations of the Product types from Figure 4.3 are displayed in Figure 4.4. The selected products consist of three sets of a pair of similar CTOs. PCF102 and PCF103 are general crates with *standing* or *lying* wood (referring to the orientation of the planks on the side walls of the crate). PCF201 and PCF206 are *plate crates*, distinguished by the internal and external braces. We refer to PCF401 and PCF407 as *skids*, which essentially are large pallets.

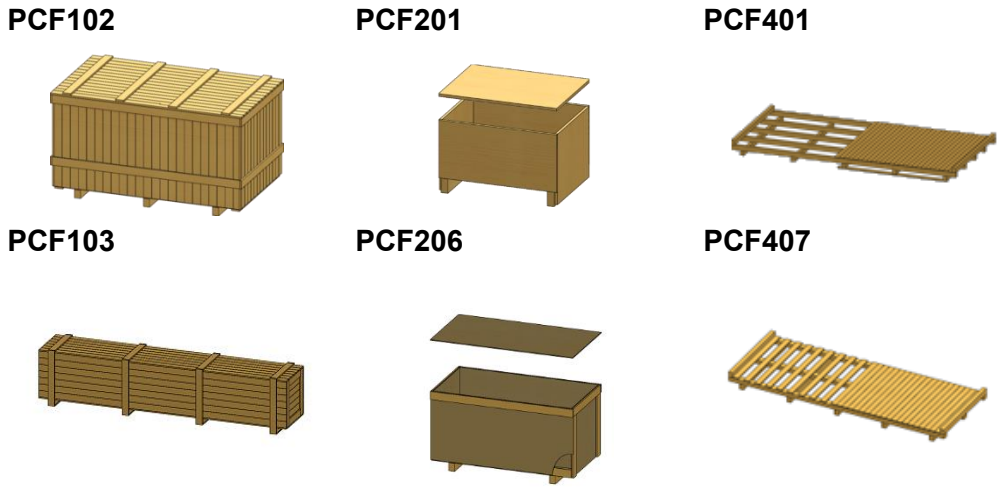



***Figure 4.4: Illustrations of the CTOs with PCF classification.***

## *4.4 Outlier Detection and Management*

Outliers in regression analysis are unavoidable and can deflect the results. Outliers can negatively impact the performance of an algorithm, leading to biased or inaccurate predictions. Overfitting occurs when a model learns all the details, noise, and outliers of a model, properly removing outliers assists in preventing overfitting and therefore improves the generalisability. Therefore, outliers should be detected and managed properly. In other words, we must establish what classifies as an outlier, and subsequently, how is the outlier dealt with. The approach is as follows: the model divides the dataset in 80% training data and 20% test data.

The model makes predictions ($Y_{pred}$) over the features in the latter 20%, which can then be compared to the label ($Y_{test}$). The difference between each of these test and prediction pairs is calculated and referred to as *residuals*. Residuals represent the deviation from the actual (test) values.

We determine the first quartile ($Q_1$), the third quartile ($Q_3$), and the Interquartile Range (IQR = $Q_1 - Q_3$) to identify the outliers (Schwertman et al., 2004; Walfish, 2006). Subsequently, we determine the upper and lower bounds using the following formulas, as demonstrated in Figure 4.5. The values below the lower or above the upper bound are considered outliers and excluded from the results.

$$\text{Lower bound: } Q_1 - 1.5 \times IQR$$
$$\text{Upper bound: } Q_3 + 1.5 \times IQR$$

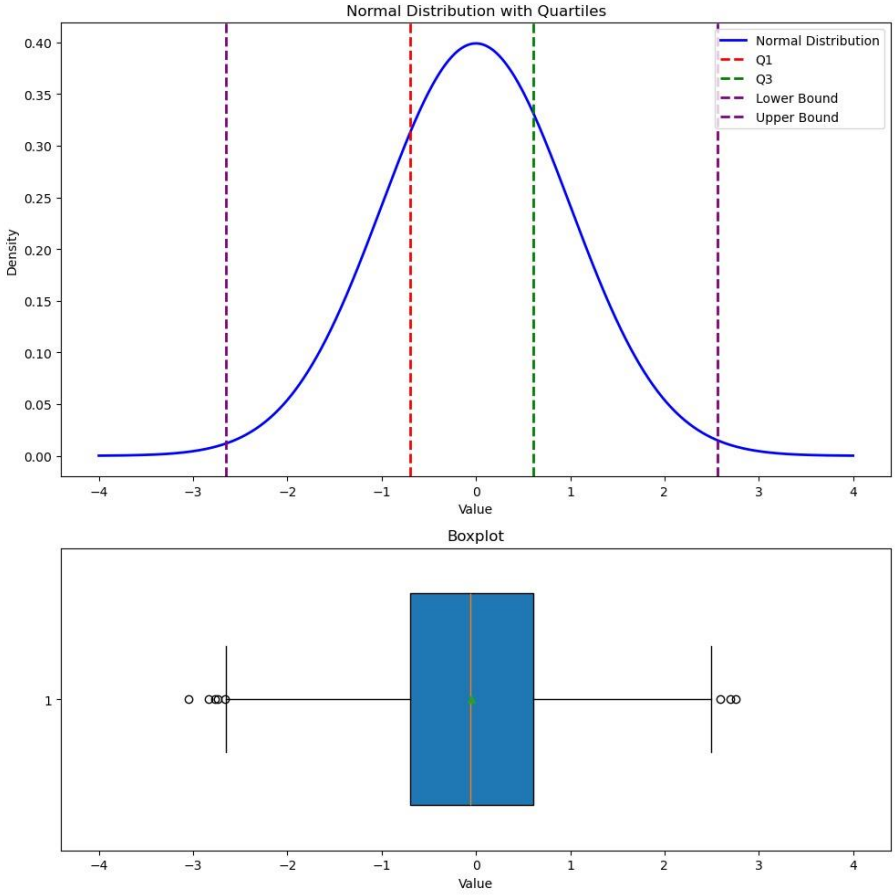



***Figure 4.5: Visualisation of Quartiles.***

## *4.5 Model Validation*

We address two principles to estimate the model performance after training the model with 80% of the data. Firstly, the model undergoes testing with nine different underlying algorithms using the remaining 20% of the dataset. The level of fit is calculated with the composite score based on the AIC and MSE, where the highest composite score (CS) represents the best fit. We evaluate the performance of the models with multiple metrics instead of one, to make a balanced evaluation, instead of optimizing for one specific criterion. The CS is obtained by normalising the performance metrics to bring them to a common scale. Dividing each value by the maximum value in its range results in a normalised range between

0 and 1. Subsequently, we assign weights to the normalised values to reflect the relative importance. The CS to compare different aspects of model performance in a single balanced score. Besides an indication of the model's errors, MSE penalises large errors more, and AIC discourages overfitting by penalising the number of parameters. The CS for MSE ($x_1$) and AIC ($x_2$) is calculated with Equation 9:

$$CS = \sum_{i=2}^{n} w_i \times (1 - \hat{x}_i) \tag{9}$$

Where:
$\hat{x}_i$: the normalised value of the $i$th $x$
$W_i$: the weight assigned to $\hat{x}_i$

The algorithm with the best fit is selected for 10-fold cross-validation to further analyse the model performance. The overall performance of the model is expressed as percentage of predictions where the actual cost value lies within the 95% confidence interval of the calculated value. In the subsequent sections, we substantiate why these criteria fit the research design and elaborate the working principles each. We illustrate the sequence of validation steps in Figure 4.6.

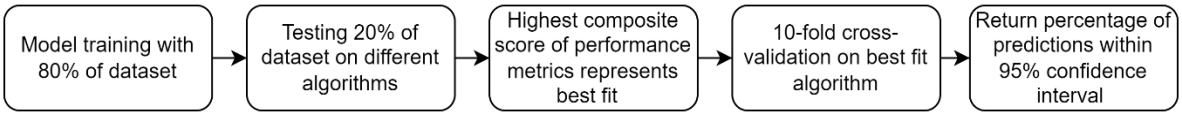



***Figure 4.6: Model validation sequence.***

### *4.5.1 K-fold Cross Validation*

Validation is useful to judge a model's accuracy (Anguita et al., 2012; Berrar, 2019). For machine learning algorithm, it is usual to train a model on the majority of the dataset and test the validity with the residual. However, the exact part of the dataset for validation is not fixed (Wong & Yeh, 2019). The '$k$' in $k$-fold model validation indicates the number of parts, and therefore the number repetition (or *folds*). In 10-fold cross-validation ($k = 10$), data are divided into ten parts, subsequently trained on nine parts of the data, and tested on the one part. In the next fold, a different part is used for the validation. This principle is illustrated in Figure 4.7.

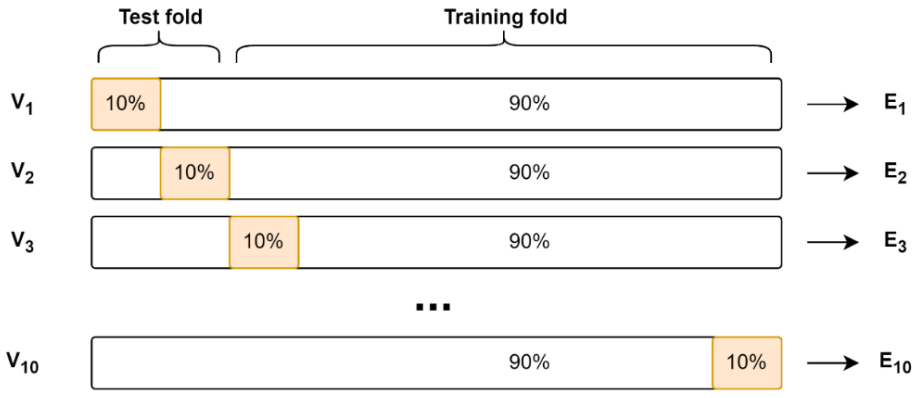



***Figure 4.7: Ten-fold cross validation.***

The performance metrics ($E_i$) are evaluated during each iteration ($V_i$). After the last fold, we can express accuracy as a degree of the model performance with Equation 10 (Sontakke et al., 2019).

$$E = \frac{1}{k}\sum_{i=1}^{k} E_i \tag{10}$$

Where:
$E_i$ = the $i$th performance metric
$E$ = the overall performance
$k$ = the number of folds

## *4.6 Statistical Test of Improvement*

The hypothesised improvement can be determined with a statistical test. In our case, we aim to verify whether an observed difference is statistically significant. We denote mean accuracy of current approach as $A_0$ and test it against our algorithm's mean accuracy $A_1$. Therefore, we formulate the following null hypothesis and alternative hypothesis:

$H_0$: $A_1 = A_0$ There is no improvement
$H_1$: $A_1 > A_0$ There is a significant improvement

We test whether the mean change in accuracy is significantly different from zero. Under the assumption that the two independent samples are normally distributed, the t-statistic is calculated with Equation 11:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2 + s_2^2 - 2\rho s_1 s_2}{n}}} \tag{11}$$

Where:
$s^2$: sample variance
$\bar{X}$: sample mean
$n$ : sample size ($n = n_1 = n_2$)
$\rho$: correlation coefficient

The significance level is $\alpha = 0.05$, and the condition of $n > 20$ must be satisfied. We calculate the probability of observing our test statistic under the null hypothesis with a t-distribution and $n - 1$ degrees of freedom. We reject our null hypotheses if we observe a p-value lower than the significance level ($p < \alpha$), and we fail to reject our null hypothesis if we observe a p-value larger than our significance level ($p > \alpha$).

## *4.7 Intermediate Conclusions*

**RQ 2.2:** *What is a suitable data expansion method in case the amount of data is insufficient?* Literature presented several data expansion methods, with each method dependent on the conditions of the research and properties of the dataset. We found that information can be re-acquired from the existing dataset by re-using the information dense areas or be reproduced based on the distribution of the existing dataset. We decided to apply K-fold cross validation feature engineering from this perspective.

**RQ 3.2:** *How do we select underlying models in our algorithm?* We calculate a composite score based on the AIC and MSE of a product-model combination. The highest composite score of a model indicates the best fit to a product.

**RQ 3.3:** *How does the model detect and manage outliers?* We apply the IQR model to detect and manage outliers in the results of the model. According to the IQR model, values below a lower bound and above an upper bound are considered outliers. We first divide the data in quartiles, then, the lower bound is calculated with $Q_1 - 1.5 \times IQR$ and the upper bound with $Q_3 + 1.5 \times IQR$, where IQR = $Q_1 - Q_3$.

**RQ 3.4:** *How does the model track labour cost predictions?* The model makes predictions based on the features presented to it in the test dataset. The prediction is a continuous, positive number that resembles the number of minutes labour required for that specific product. The range of estimations is visualised in a scatterplot which plots the predicted values against the actual values. The degree to which points gather around a 45° line between the y-axis and the x-axis gives an impression of the fit and accuracy of the model. The model calculates the percentage of predictions where the actual value is within its 95%-confidence interval. Performance metrics describe the degree of how the data fits the model and relative feature importances can be extracted from the model.

**RQ 3.5:** *How does the model reduce the risk of overfitting?* Overfitting can result in a reduced generalisability and occurs when a model learns the details of a dataset to an extent that it negatively impacts the performance of the model on a new dataset. We apply outlier management as part of combatting this risk. Presence of outliers in the data can distort the learning process of the model as it tries to fit the outliers instead of the true underlying patterns. Insufficient training data or and excessive number of parameters can also lead to overfitting. The evaluation metric AIC penalizes larger numbers of underlying parameters more, therefore awarding a preference for fewer parameters, reducing the risk of overfitting. Lastly, by using K-fold cross validation, we ensure that every data point is once used to train and test the models. This gives us an indication of how the models perform across different subsets of the data and the generalisation ability.

# 5

# RESULTS

We subjected our model to the dataset to predict labour times of CTOs. In this chapter, we describe the results of the application of the dataset to our Machine Learning model, as well as further analyses, and statistical interpretation of significance. We attached a detailed description of the model design and its computational steps in Appendix D. We present the results as processed output collected from the models.

## *5.1 Method Selection*

The first step is to find the best fitting model for each product. We assess relative performance using the composite score we described in Section 4.5, with equal weights assigned to AIC and MSE. We calculate the weighted average of the normalised values of MSE and AIC to account for the difference in order of magnitude. The composite scores in Table 5.1 indicates the relative goodness of fit of each model for that product and range from a minimum score of zero to a maximum score of two. Complete results of model selection and the performance metrics table are attached in Appendix E and the normalised values in Appendix F.

***Table 5.1: Composite scores of performance metrics per product per method.***

| PCF | LR | GPR | RFR | MLP | SVM | DTR | GBR | KNN | XGB |
|---|---|---|---|---|---|---|---|---|---|
| PCF102 | 0.672 | 0.745 | 0.630 | 0.612 | 0.642 | 0.000 | **0.751** | 0.744 | 0.494 |
| PCF103 | 0.159 | 0.000 | 0.310 | 0.155 | 0.274 | 0.124 | 0.328 | 0.335 | **0.393** |
| PCF201 | 0.712 | 0.087 | 0.700 | 0.633 | **0.856** | 0.327 | 0.000 | 0.804 | 0.312 |
| PCF206 | 0.302 | 0.694 | 0.968 | 0.486 | 0.659 | **0.972** | 0.565 | 0.449 | 0.959 |
| PCF401 | 0.953 | 1.106 | 1.100 | 0.000 | 0.942 | 1.097 | 1.083 | **1.149** | 1.080 |
| PCF407 | **0.415** | 0.296 | 0.220 | 0.064 | 0.086 | 0.240 | 0.408 | 0.000 | 0.109 |

The highest composite score indicates the best fit to that product and are highlighted in Table 5.1. The highest scoring method for a product means that that method's predictions most correspond the to the true values out of all tested methods.

***Table 5.2: Most appropriate techniques per product.***

| PCF | Method | Composite score |
|---|---|---|
| 102 | GBR | 0.751 |
| 103 | XGB | 0.393 |
| 201 | SVM | 0.856 |
| 206 | DTR | 0.972 |
| 401 | KNN | 1.149 |
| 407 | LR | 0.415 |

Remarkably, there is no single method that fits all products, each product scores best with a different method. This variation confirms the importance of proper model selection for (other) future products. We condensed the best fitting models per PCF into Table 5.2. These product-model combinations are subjected to further validation in the following section to assess their performance.

## *5.2 Model Performance*

Now we assigned each product to its best-performing method, the next step is to validate the models using K-fold cross validation, as described in 4.5.1. We perform K-fold cross validation to provide a more accurate estimate of the model performance. We divide the data into ten subsets, use each subset as test data once, and calculate performance metrics for each fold, the average of the metrics gives a better approximation of how the model will perform on new (unseen) data. The following metrics calculated for each of the ten folds: Average MSE, Average AIC, and Accuracy. The values in Table 5.3 represent the average over all 10 folds, we attached detailed results of each fold in Appendix G.

***Table 5.3: Average of the performance metrics from 10-Fold Validation.***

| PCF | Method | n | Average MSE | Average AIC | Accuracy |
|---|---|---|---|---|---|
| 102 | GBR | 219 | 15156.51 | 261.49 | 50.68% |
| 103 | XGB | 160 | 2960.47 | 171.85 | 39.38% |
| 201 | SVM | 68 | 1394.80 | 89.44 | 50.00% |
| 206 | DTR | 95 | 1471.81 | 121.60 | 44.21% |
| 401 | KNN | 187 | 1685.49 | 191.28 | 55.06% |
| 407 | LR | 82 | 766.08 | 103.47 | 46.34% |

Table 5.3 displays the results of each the product-model combinations. The results consist of an average MSE, average AIC and accuracy. The average MSE gives an indication of the errors in the predictions. We observe a relatively high MSE for PCF102, we know MSE penalizes larger errors more, therefore, we analyse the error distributions later this chapter. Average AICs are similarly distributed as the average MSEs, we also observe a higher value for PCF102. Higher values for this product compared to the other products does not necessarily mean that this product-model combination performs inaccurate. We can judge by Figure 4.4 that PCF102 and PCF103 are relatively larger and require more labour (time), which explains the relatively larger errors. Therefore, we must assess the error distributions relative to their order of magnitude to account for this issue. The accuracy percentages represent the percentage of predictions where the true value lies within the 95%-CI of the prediction. We compare the found values to the benchmark values later in this chapter

We extracted the relative feature importance for each product and display the top three in Table 5.4. We attached the complete list in Appendix H. For each product, we present a list of three features with the most impact on labour cost. The first number is the identification code of the feature, followed by the name of the feature and the relative score. A higher score represents a larger impact on the prediction. From another perspective, higher importance suggests that it is more closely related to the labour cost. The order of magnitude of the relative

scores depends on the nature of the models and cannot be compared between different models.

***Table 5.4: Top three most features impacting the labour cost per product.***

**PCF102 (GBR)** Feature Importances:

| | Feature | Importance |
|---|---|---|
| 6 | _VERBINDING3 | 2.294067e-01 |
| 1 | _D2085 | 2.029338e-01 |
| 5 | _VERBINDING2 | 1.304876e-01 |

**PCF103 (XGB)** Feature Importances:

| | Feature | Importance |
|---|---|---|
| 19 | 10KOPSCHOTTEN | 0.350150 |
| 18 | 10JUKKEN | 0.241663 |
| 11 | 10BODEM | 0.106932 |

**PCF206 (DTR)** Feature Importances:

| | Feature | Importance |
|---|---|---|
| 7 | _wvbBalken | 0.241509 |
| 1 | _D2085 | 0.198818 |
| 26 | 20MONTAGE | 0.142096 |

**PCF201 (SVM)** Feature Importances:

| | Feature | Importance |
|---|---|---|
| 0 | _D2049 | 0.514581 |
| 7 | _wvbBalken | 0.367783 |
| 25 | 10ZIJSCHOTTEN | 0.145435 |

**PCF401 (KNN)** Feature Importances:

| | Feature | Importance |
|---|---|---|
| 5 | _VERBINDING2 | 0.141970081799 |
| 1 | _D2085 | 0.110134167218 |
| 2 | _D3241 | 0.095009054677 |

**PCF407 (LR)** Feature Importances:

| | Feature | Importance |
|---|---|---|
| 4 | _hijstrek | 1.554981e+03 |
| 7 | _wvbBalken | 8.111384e+01 |
| 1 | _D2085 | 3.038155e+01 |

This knowledge is valuable to the firm to identify cost intensive areas within their manufacturing process. When a feature significantly impacts labour cost, it also holds the most potential in cost reductions. The firm can use this decision support to target these components of process steps for optimisation or additional employee training to reduce costs effectively. The key takeaway from the feature importances is the insight into impacts of different process components on labour costs. While an in-depth analysis of feature importance is beyond the scope of this research, we provide this method to access this information, enabling the firm to use it in future research.

Consider an example of a practical analysis of an insight gained from Table 5.4: We can relate the feature importances to the product types, for example, PCF102 and PCF103 are (generally large) crates with *standing* or *lying* wood (see Figure 4.4), respectively. Crates with standing wood are structurally efficient for packaging tall objects, lying wood crates are more used for packaging long and slim objects (e.g., long pipes). This explains the prominent presence of '10JUKKEN' for PCF103, which implies the presence of supports for (long) products in the crate. These extra parts have to be produced and fitted in the crate, which requires additional labour time. Logically, the presence of supports directly contributes to labour cost, this knowledge enables the firm to target cost intensive parts of manufacturing processes.

## *5.3 Statistical Interpretation*

We test for significant difference between the current accuracy and the accuracy of our model, using paired t-test. In this statistical test, we test whether the mean change of the residuals is significantly different from zero. We assume observations are independent, and the variables are approximately normally distributed. We denote mean accuracy of current approach as $A_0$ and test it against our algorithm's mean accuracy, $A_1$. The significance level

is set to $\alpha = 0.05$, and we satisfy the paired t-test sample size condition of $n > 20$. We formulate the following hypotheses:

$H_0$: $A_1 = A_0$ There is no improvement
$H_1$: $A_1 > A_0$ There is a significant improvement

We reject our null hypotheses if we observe a p-value lower than the significance level ($p < \alpha$), which indicates that the difference does not equal zero. We fail to reject our null hypothesis if we observe a p-value larger than our significance level ($p > \alpha$). Table 5.5 displays our summarized findings of the paired t-test, including the degrees of freedom ($df$), t-statistic, p-value, and a significance conclusion.

***Table 5.5: Paired t-test results for the comparison of prediction means.***

| Product | $df$ | t-statistic | p-value | Improvement | Significant |
|---|---|---|---|---|---|
| **PCF102** | 218 | 5.2220 | <0.0001 | 32.95% | Yes |
| **PCF103** | 159 | 3.1767 | 0.00190 | 9.93% | Yes |
| **PCF201** | 67 | 3.8245 | 0.00033 | 20.59% | Yes |
| **PCF206** | 94 | 3.6775 | 0.00041 | 18.95% | Yes |
| **PCF401** | 186 | 3.1607 | 0.00200 | 16.11% | Yes |
| **PCF407** | 81 | 2.5301 | 0.01312 | 31.91% | Yes |

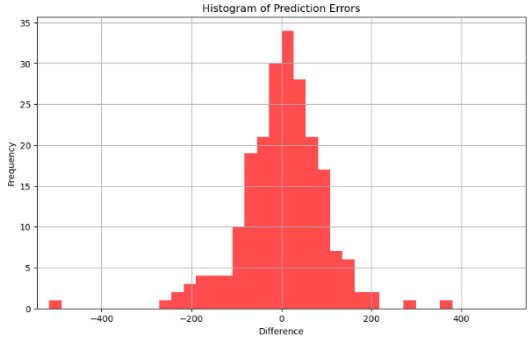
Our sample data supports our alternative hypothesis that difference in mean accuracy does not equal zero. The p-value of the paired t-tests of every product is lower than our level of significance ($p < \alpha$). Therefore, our algorithm significantly improves labour cost prediction for each product. This implies that our algorithm's predictions are systematically closer to the actual labour cost than the firm's current approach. Not only does this affect the precision of quotations, but it also enables the firm to schedule operations with greater precision.

## *5.4 Error Analysis*

Understanding the nature and patterns of errors made by the models are important to interpret its performance. We intend to gain understanding of the error by visualising the residuals, including outliers. We divided Table 5.6 into six histograms, where each plot shows the frequency and the intensity of the errors, including outliers. We attached outlier management details in Appendix I and absolute error distribution in Appendix J.

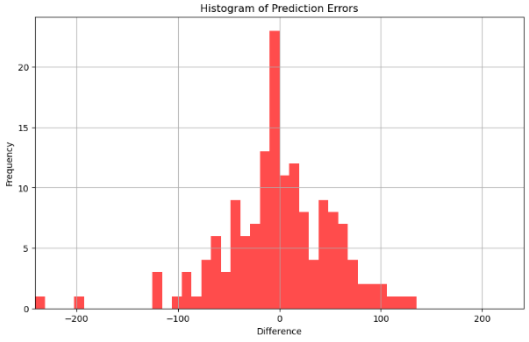
***Table 5.6: Prediction error distribution per product.***

**PCF102 (GBR)**



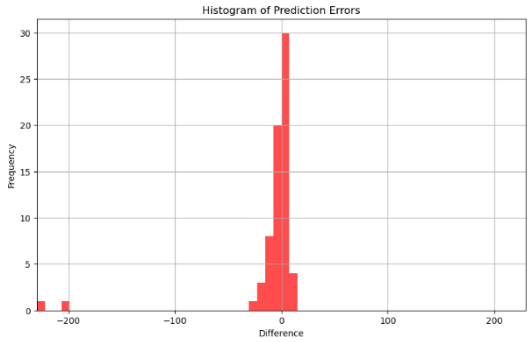

Mean: -1.9968
Median: 6.2940
The distribution is skewed to the left with a skewness (k3) of -3.85.

**PCF103 (XGB)**



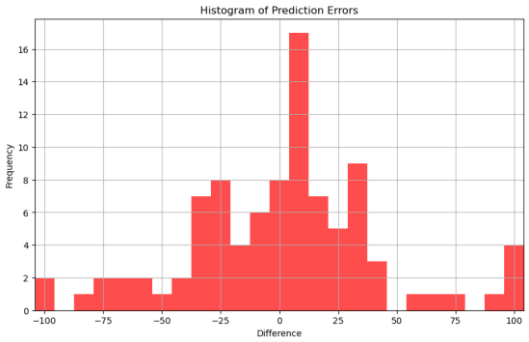

Mean: -1.0967
Median: -2.5675
The distribution is skewed to the left with a skewness (k3) of -0.75.

**PCF201 (SVM)**



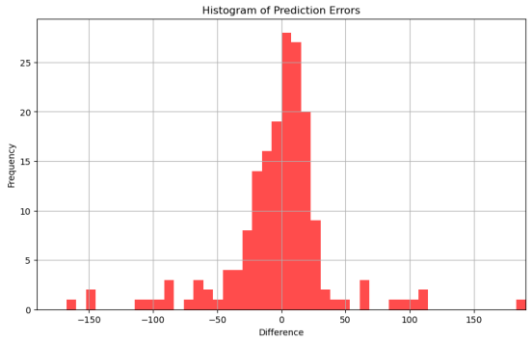

Mean: -7.8698
Median: -0.0940
The distribution is skewed to the left with a skewness (k3) of -5.24.

**PCF206 (DTR)**



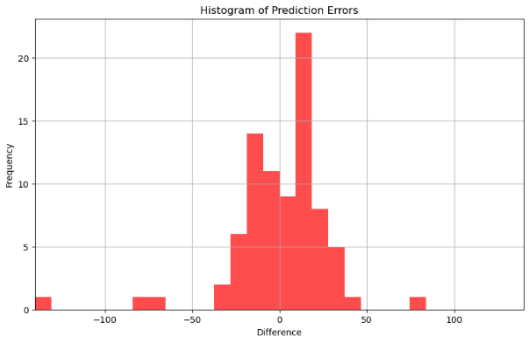

Mean: 4.1692
Median: 7.5500
The distribution is skewed to the right with a skewness (k3) of 0.18.

**PCF401 (KNN)**




Mean: -2.2932
Median: 2.3500
The distribution is skewed to the left with a skewness of (k3) -0.26.

**PCF407 (LR)**




Mean: 0.8116
Median: 1.8524
The distribution is skewed to the left with a skewness (k3) of -1.72.

The general distribution in the error analysis suggests several key points: The peaks around zero indicates that the majority of the predictions are relatively close to the true value, which suggests that the model is generally accurate. Furthermore, an exponential decrease in the frequency of errors suggests that there are fewer instances where the model makes large errors. Lastly, the distributions for PCF206 and PCF407 display non-zero peaks, suggesting that the prediction are affected by a systematic error. This systematic error can be the result of overfitting, possibly by a lack of data. In Table 5.6, we included the mean, median, and skewness ($k_3$) to measure the level of asymmetry of the error distributions. Skewness of the error distributions is calculated with Equation 12:

$$k_3 = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{X_i - \mu}{\sigma}\right)^3 \tag{12}$$

Where:
$X_i$: individual errors
$\mu$: mean error
$\sigma$: error standard deviation
$n$: sample size

The mean indicates whether, on average, the predictions tend to underestimate (negative value) or overestimate (positive value). The median is the middle value when all errors are sorted in ascending order. A positive median indicates that more than half of the predictions are positive, a negative median indicates that more than half of the predictions are negative. Skewness is interpreted as follows (Doane & Seward, 2011):

$|k_3| \approx 0$ : The distribution is symmetrical.
$|k_3| < 0$ : The distribution is left-skewed.
$|k_3| > 0$ : The distribution is right-skewed.

We select $k_3$ to determine the skewness of the error distributions, Table 5.7 lists the skewness of each product.

***Table 5.7: Skewness of error distributions per product.***

| Product | Skewness ($k_3$) | Skewed: |
|---|---|---|
| PCF102 | -3.85 | Left |
| PCF103 | -0.75 | Left |
| PCF201 | -5.24 | Left |
| PCF206 | 0.18 | right |
| PCF401 | -0.26 | Left |
| PCF407 | -1.72 | Left |

Negative skewness or left-skewed error distributions imply an asymmetry in the predictions, i.e., systematic underestimations, and a right-skewed error distribution indicates systematic overestimation in the predictions. We observe values near zero for PCF103,

PCF206, and PCF401, which indicates that the distributions are slightly skewed but approximately symmetrical. The other products, however, show higher levels of left-skewed error distributions, this indicates that there are a number of large underestimations that pull the distribution to the left. Such a phenomenon suggests that the model does not handle large values optimally and is inclined to underestimate in some cases (Flach, 2003). We discuss further implications of systematic prediction deviations in Chapter 6.

We presented the results of the models on the corresponding products in this chapter. Additionally, we showed statistical significance, relative feature importances, and an error analysis. The results allow us to answer the last sub-research questions:

**RQ 4.1:** *What is the relative feature importance?* With the results in Table 5.4 and Appendix H, we can analyse the relative impact of each parameter on the labour cost. We analysed one example for the presence of supports in PCF103. The availability of feature importances provides decision support for the firm to targeted optimization of cost intensive components in manufacturing processes.

**RQ 4.2:** *What accuracy improvement can we achieve with our algorithm?* We observed improvements of labour cost prediction accuracy for each of the products and achieved an overall labour cost prediction accuracy improvement of 21.74%. Furthermore, our algorithm showed generalisability in K-fold cross validation, which suggests that it is likely to be applicable on other products.

# 6

# DISCUSSION

In this chapter, we discuss the results and their implications. This includes the interpretation of the results, the validity, and the limitations. Components resulting from this discussion form the foundation for the conclusion and recommendations for future research. We start with a brief review of the research questions and how we addressed those. Subsequently, we assess the validity of our findings, and explore the implications.

## *6.1 Review of Objective and Research Questions*

We recall our primary objective: *To develop an algorithm that predicts labour costs of CTOs with the actual labour cost falling within the 95% confidence interval of the predictions,* from which we formulated the main research question: *How can the firm accurately and systematically predict labour costs for configurated products?* We formulated multiple sub-questions to answer the main research question, and ultimately, to reach our objective. We briefly go through the process of our research in the following paragraphs.

Authors of reviewed literature discussed several approaches to calculate costs in a multi-parametric context. We investigated multiple machine learning categories and selected nine supervised techniques, with Python selected as the programming language due to the availability of libraries. A common rule of thumb in machine learning is that the dataset should be at least ten times the number of features, our dataset failed to satisfy this condition. To address this, we reviewed literature regarding data expansion methods and re-acquired from our dataset by re-using the information dense areas through K-fold cross validation and feature engineering. K-fold cross validation also reduces the risk of overfitting and gives an indication of a model's generalisation ability.

We evaluated relative performance and fit of models with the metrics MSE and AIC. A positive value closer zero indicated a better fit. We applied the IQR method to detect and manage outliers in the predictions of the model. Predictions were made based on the features in the test dataset. Prediction values are continuous, positive numbers that resemble the number of minutes labour required to manufacture that specific product. The model determined, for each prediction, whether the actual value was within the prediction's 95%-confidence interval.

In accordance with our expectations, the models predicted labour cost with accuracies greater than that of the benchmark, therefore indicating a significant improvement. Furthermore, our error analysis of the residuals between predictions and true values revealed a number of insights. The distributions of the errors are centred near zero, indicating that the majority of the predictions is closer to the actual value and larger errors occur with a lower frequency. However, the error distributions for PCF102, PCF201, and PCF407 display non-zero peaks or left-skewed error distributions, suggesting that a systematic error affects the prediction. This phenomenon suggests that the selected machine learning techniques for those products do not perform optimally for large numbers and are therefore sensitive to underestimation. Training with additional data that involves larger numbers could improve the systematic underestimations and overall accuracy.

The implications of a systematic error in labour cost predictions vary per department. it might favour the sales department if the predictions are systematically below par because it allows them to quote more competitive prices. However, a systematic higher prediction favours both sales and operations, as it allows for a less tight schedule and a higher profit margin. Logically argued, operations prefer too much time over too little time. To conclude, skewness needs to be accounted for in the interpretation of the prediction, and accuracy must improve with additional data, to predict labour costs objectively, that favour neither sales nor operations.

## *6.2 Validity*

We used 811 samples of six products to train our algorithm that predicts labours costs, we observed patterns in the results that match our expectations. We observed that performance for each product varies with different models and the best fit was identified with the composite score of the performance metrics. K-fold cross validation revealed an underlying variation that was not initially detected in the model selection phase, we used this method for exactly that purpose. We observed labour cost prediction accuracies that are significantly greater than the benchmark.

Our research relates to our literature review by the combination of the methods investigated. We gained information regarding properties and conditions of methods in the literature review and applied this in our model selection method. From this perspective, we consider our model an addition to the literature. We expected a reduced transparency of underlying relations in machine learning, however, we were able to extract valuable information, we found the relative feature importances and created several visualisations of results versus predictions, residuals, and comprehensive tables. With our research, we have shown that labour costs can be predicted with higher accuracy than the firm's previous approach.

The research design includes several elements to ensure the validity. We systematically excluded outliers using the IQR method. We tested significance with a paired t-test with a 5% level op significance. We systematically calculated composite scores for multiple performance metrics to select methods objectively. We performed data preparation processes methodically and we developed a feature engineering method for reproducibility and improving accuracy. Based on this, we can state that this research is replicable.

There are some areas where further improvements can be made. It must be taken into account that performance metrics might score differently for each observed model. For instance, some metrics might be more suitable to evaluate KNN and another metric might be more suitable to evaluate LR. However, we made the decision to use MSE and AIC for all models to allow for comparison in model selection, based on the generalisability of the metrics demonstrated in literature. We improved quality of the data through methodical data cleaning and feature engineering. We found the rule of thumb for the minimal sample size in machine learning to be at least ten times the number of features. With twenty-seven features, none of the products satisfy this requirement. We expect the performance to improve as more data are collected over time, since it exposes our model to a broader range of feature values. Lastly, as data are collected as a series over time, new data can become more relevant than older data. Efficiency of the staff can improve with experience or due to implemented improvements, we chose not to account for this effect in our model due to the relatively short period in which the data were collected.

# 7

# CONCLUSION

In this chapter, we summarise the key findings and conclusions. We addressed the research question: *How can the firm accurately and systematically predict labour costs for configurated products?* Through exploration of literature, model validation, and data analysis, we have developed an algorithm to answer this question. Our algorithm finds the underlying relations between features of CTOs and its labour cost by learning from historical product data. The dataset consists of unique products, each describes by 27 features, and their corresponding labour costs. We used 80% of the dataset to train the models and reserved 20% for testing. During the training phase, the was exposed to both the features and the labour costs. In the testing phase, we provided only the features to the model to observe whether the model predictions match the actual labour costs.

In this algorithm, each product was assigned to its most appropriate machine learning technique, based on performance metrics MSE and AIC. Outliers were excluded by using the IQR method to reduce the risk of overfitting. We investigated generalisability by cross-validation and extracted important features to engineer new features, increase accuracy, and gain insights into the value streams of the manufacturing processes. Comparison of predictions to the actual values forms our error analysis, which enabled us to compute an accuracy percentage. We observed a significant improvement of the overall accuracy of 21.74% based on 811 observations. Table 7.1 shows the results of a paired t-test on the accuracy improvement per product, at a 5% level of significance.

***Table 7.1: Observed improvements in labour cost prediction accuracy ($\alpha = 0.05$).***

| Product | Method | Accuracy (before) | Accuracy (after) | $df$ | t-statistic | p-value | Improvement |
|---|---|---|---|---|---|---|---|
| **PCF102** | GBR | 17.73% | 50.68% | 218 | 5.2220 | <0.0001 | 32.95% |
| **PCF103** | XGB | 29.45% | 39.38% | 159 | 3.1767 | 0.00190 | 9.93% |
| **PCF201** | SVM | 29.41% | 50.00% | 67 | 3.8245 | 0.00033 | 20.59% |
| **PCF206** | DTR | 25.26% | 44.21% | 94 | 3.6775 | 0.00041 | 18.95% |
| **PCF401** | KNN | 38.95% | 55.06% | 186 | 3.1607 | 0.00200 | 16.11% |
| **PCF407** | LR | 14.43% | 46.34% | 81 | 2.5301 | 0.01312 | 31.91% |
| **Overall** | | **25.87%** | **47.74%** | | | | **21.74%** |

We can conclude that we achieved a significant improvement in the labour cost prediction for all investigated CTOs, and our approach provides new insights into methods to gain more information from the data at hand. Implementing our algorithm allows the firm to estimate labour costs 21.74% more accurately, this impacts multiple facets of the company. The improved accuracy enables the sales department to make more informed decisions and to provide customers with more accurate quotations, reducing the frequency of cost overruns and improving client satisfaction. Additionally, a greater insight into labour time prediction improves the ability to schedule activities with greater precision.

## *7.1 Contribution*

The labour cost prediction algorithm we developed, provides a practical and systematic approach to predict labour costs of CTOs more accurately, contributing to cost estimation in manufacturing processes. Our research contributes to literature by exploring application of machine learning in cost estimation for configurable products. Our algorithm compares performance of different machine learning principles and can be adapted and scaled to different processes. Furthermore, the performance evaluation of multiple machine learning algorithms with a composite score of MSE and AIC, provided insight in their effectiveness in cost estimation.

We developed a solution that utilises historical data to increase precision in the future. Our labour cost prediction algorithm significantly improves the accuracy of cost prediction, enabling the firm to quote more accurately and schedule activities with greater precision. The implications of our development affect many facets of the firm. The improvement of labour cost estimation precision allows the sales department to quote more accurately and make more informed decisions, effectively reducing the frequency of cost overruns. This directly improves client satisfaction and leads to stronger relationships with customers, and therefore, increasing competitive advantage.

An increase in scheduling precision benefits the Meilink in multiple ways, firstly, it relieves the scheduling department by effectively simplifying their task. Secondly, the scheduled time is based on historical times and more accurate than the previous approach. There will be fewer instances where operations have too little or too much time scheduled, therefore, potentially wasting less time waiting, or correcting potential errors incurred by hurrying. Lastly, every scheduled action will finally add to the historical database, further improving (scheduling) precision.

Our more practical and precise approach also improves communication between operations and the sales department. Disagreement over scheduled labour time is less likely if all involved parties are informed of the underlying method with demonstrated accuracy, which should favour neither the sales department nor operations. In addition, our algorithm helps to identify cost intensive components of the manufacturing process. This information helps to allocate resources effectively, identify areas to investigate cost efficiency, and insight into the activities that add value the most. Cost and waste reduction can be achieved by investigating important features and allocate resources accordingly. This knowledge also enables the firm to target operational areas where employees require more training or skill development.

Ultimately, practical implications of our labour cost prediction algorithm align with the commitment to improving customer satisfaction, maintaining a competitive advantage, and ensuring the continuity of the firm.

# 8

# RECOMMENDATIONS

In chapter, we briefly describe recommendations for future research and why we expect this to be of added value. Our recommendations focus on improving accuracy and validity, as well as the further analysing underlying relationships and implementation of our algorithm.

The current dataset is based on an output of Merkato (recall: the configurator software), using the input of the configurator directly as input for our machine learning algorithm might reveal more accurate results, because the direct and unprocessed input of the configurator might reveal more underlying relationships and improve the accuracy. Moreover, if the algorithm is optimised on the direct input, the output of the algorithm can be immediately displayed during configuration. Integrating our algorithm directly into the configurator could be achieved in collaboration with the developer of the configurator. Future research should therefore examine the options for direct integration, possibly by only using a function of the regression coefficient values.

In addition to integration, we recommend expanding the scope of testing, for instance, testing on a broader range of products, with more samples that contain a wider range of features. We concluded that accuracy increases with a larger sample size. Generalisability can be further investigated if data of a broader range of products are available and higher relevance is assigned to more recent data. Furthermore, evaluation with alternative performance metrics can also be beneficial. Metrics such as *Symmetric Mean Absolute Percentage Error* (SMAPE) that incorporate symmetry in the evaluation (Chicco et al., 2021) or *Root Mean Squared Logarithmic Error* (RMSLE) for skewed targets. RMSLE penalises underestimates more than overestimates (Aldrees et al., 2022). RMSLE as a performance metric can be useful to reduce the risk of selling a product with a negative profit. Specifically, to reduce risks from the systematic error we observed in our error analysis.

To address the robustness of the algorithm, we suggest introducing various levels of noise to the dataset before conducting experiments. We expect that this provides insights into the resilience of the algorithm's ability to handle real-world variability. Machine learning techniques in our algorithm can be further tuned to the dataset by adjusting hyperparameters, to further improve the accuracy of the model. This can be achieved by investigating the effect of different hyperparameter levels on the performance of the model, for example, by *grid search* or *random search (Li et al., 2018)*.

Furthermore, we recommend investigating the advantages and disadvantages of assigning a model to one or all products permanently, neglecting the model selection phase. At a certain point, there is sufficient empirical evidence that a product performs stably and consistently with a certain technique and can be permanently assigned to the specific model. Valuable computational time can be saved by omitting the model selection phase. Lastly, we recommend performing a feature importance analysis with e.g., *permutation importance* (Altmann et al., 2010) or *Shapley Additive Explanations* (SHAP) (Nohara et al., 2022), to determine the reason some features have more impact on the labour cost than others. Insights into this behaviour can be used to identify parts of the manufacturing process that are cost intensive or add the most value to the products, thereby guiding decisions to optimize efficiency and maximize product value.

# REFERENCES


Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., & Arshad, H. (2018). State-of-the-Art in Artificial Neural Network Applications: A Survey. *Heliyon*, *4* (11).

Agatonovic-Kustrin, S., & Beresford, R. (2000). Basic Concepts of Artificial Neural Network (ANN) Modeling and its Application in Pharmaceutical Research. *Journal of Pharmaceutical and Biomedical Analysis*, *22* (5), 717-727.

Aldrees, A., Khan, M. A., Tariq, M. A. U. R., Mustafa Mohamed, A., Ng, A. W. M., & Bakheit Taha, A. T. (2022). Multi-Expression Programming (MEP): Water Quality Assessment Using Water Quality Indices. *Water*, *14* (6), 1-24.

Alloghani, M., Al-Jumeily, D., Mustafina, J., Hussain, A., & Aljaaf, A. J. (2020). A Systematic Review on Supervised and Unsupervised Machine Learning Algorithms for Data Science. *Supervised and Unsupervised Learning for Data Science*, pp. 3-21.

Altmann, A., Toloşi, L., Sander, O., & Lengauer, T. (2010). Permutation Importance: a Corrected Feature Importance Measure. *Bioinformatics*, *26* (10), 1340-1347.

Ambarwari, A., Adrian, Q. J., & Herdiyeni, Y. (2020). Analysis of the Effect of Data Scaling on the Performance of the Machine Learning Algorithm for Plant Identification. *Jurnal Rekayasa Sistem Dan Teknologi Informasi*, *4* (1), 117-122.

Anguita, D., Ghelardoni, L., Ghio, A., Oneto, L., & Ridella, S. (2012). The 'K' in K-fold Cross Validation. European Symposium on Artificial Neural Networks, Computational Intelligence and Machine learning, 102, 441-446.

Antony, J. (2023). *Design of Experiments for Engineers and Scientists*. Elsevier, Amsterdam.

Asaolu, T., & Nassar, M. (2007). Essentials of Management Accounting & Financial Management, 2. Cedar Productions, Nigeria.

Asuero, A. G., Sayago, A., & González, A. (2006). The correlation coefficient: An Overview. *Critical Reviews in Analytical Chemistry*, *36* (1), 41-59.

Baker, B., Gupta, O., Naik, N., & Raskar, R. (2016). Designing Neural Network Architectures Using Reinforcement Learning. arXiv preprint: 1611.02167.

Basheer, I. A., & Hajmeer, M. (2000). Artificial Neural Networks: Fundamentals, Computing, Design, and Application. *Journal of Microbiological Methods*, *43* (1), 3-31.

Berrar, D. (2019). Cross-Validation. *Encyclopedia of Bioinformatics and Computational Biology*, 1, Elsevier, Amsterdam, 542-545.

Berwing, K., Schuh, G., & Stich, V. (2022). Generation of a Data Model For Quotation Costing Of Make To Order Manufacturers From Case Studies. Proceedings of the Conference on Production Systems and Logistics.

Bonakdari, H., & Zeynoddin, M. (2022). Goodness-of-fit & Precision Criteria. *Stochastic Modeling: A Thorough Guide to Evaluate, Preprocess, Model and Compare Time Series With MATLAB Software*, pp. 187-264.

Carleo, G., Cirac, I., Cranmer, K., Daudet, L., Schuld, M., Tishby, N., Vogt-Maranto, L., & Zdeborová, L. (2019). Machine Learning and the Physical Sciences. *Reviews of Modern Physics*, *91* (4), 39-45.

Cavalieri, S., Maccarrone, P., & Pinto, R. (2004). Parametric vs. Neural Network Models for the Estimation of Production Costs: A Case Study in the Automotive Industry. *International Journal of Production Economics*, *91* (2), 165-177.

Cavanaugh, J. E., & Neath, A. A. (2019). The Akaike Information Criterion: Background, Derivation, Properties, Application, Interpretation, and Refinements. *Wiley Interdisciplinary Reviews: Computational Statistics*, *11* (3), 1460-1471.

Chicco, D., Warrens, M. J., & Jurman, G. (2021). The Coefficient of Determination R-squared is More Informative Than SMAPE, MAE, MAPE, MSE and RMSE in Regression Analysis Evaluation. *Peerj computer science*, *7,* 1-24.

Chienwichai, W., Wannasin, J., Sinthavalai, R., & Meemongkol, N. (2016). Model-based cost estimates for selecting a die casting process. *The Engineering Economist*, *61* (1), 57-69.

Cohen, I., Huang, Y., Chen, J., Benesty, J., Benesty, J., Chen, J., Huang, Y., & Cohen, I. (2009). Pearson Correlation Coefficient. *Noise Reduction in Speech Processing*, 1-4.

Cokins, G. (2002). *Activity-Based Cost Management: an Executive's Guide*. John Wiley & Sons, NY.

Costa, L., Guerreiro, M., Puchta, E., de Souza Tadano, Y., Alves, T. A., Kaster, M., & Siqueira, H. V. (2023) Multilayer Perceptron. *Introduction to Computational Intelligence*, *105.*

Dangeti, P. (2017). *Statistics for Machine Learning*. Packt Publishing Ltd, Birmingham.

Denkena, B., Lorenzen, L.-E., & Schürmeyer, J. (2009). Rule-Based Quotation Costing of Pressure Die Casting Moulds. *Production Engineering*, *3, 87-94*.

Doane, D. P., & Seward, L. E. (2011). Measuring Skewness: a forgotten statistic?. *Journal of Statistics Education*, 19 (2), 1-18.

Donders, A. R. T., Van Der Heijden, G. J., Stijnen, T., & Moons, K. G. (2006). A Gentle Introduction to Imputation of Missing Values. *Journal of Clinical Epidemiology*, *59* (10), 1087-1091.

Drury, C. M. (2013). *Management and Cost Accounting*. Springer, NY.

Eksteen, B., & Rosenberg, D. (2002). The Management of Overhead Costs in Construction Companies. Bildiri. 18th Annual Association of Researchers in Construction Management Conference, 2-4.

Flach, P. A. (2003). The Geometry of ROC Space: understanding machine learning metrics through ROC isometrics. In Proceedings of the 20th International Conference on Machine Learning, 194-201.

Flach, P. (2019). Performance Evaluation in Machine Learning: The Good, The Bad, The Ugly, and The Way Forward. Proceedings of the Association for the Advancement of Artificial Iintelligence Conference on Artificial Intelligence Vol. 33 (1), 9808-9814.

Fox, J. (2019). *Regression Diagnostics: An Introduction*. Sage publications, Thousand Oaks.

Franceschini, G., & Macchietto, S. (2008). Model-Based Design of Experiments for Parameter Precision: State of the Art. *Chemical Engineering Science*, *63* (19), 4846-4872.

Gonzalez-Carrasco, I., Garcia-Crespo, A., Ruiz-Mezcua, B., & Lopez-Cuadrado, J. L. (2012). An optimization Methodology For Machine Learning Strategies and Regression Problems in Ballistic Impact Scenarios. *Applied Intelligence* (36), 424-441.

Graupe, D. (2013). *Principles of Artificial Neural Networks* (Vol. 7). World Scientific, London.

Gupta, H. V., Kling, H., Yilmaz, K. K., & Martinez, G. F. (2009). Decomposition of the Mean Squared Error and NSE Performance Criteria: Implications for Improving Hydrological Modelling. *Journal of Hydrology*, *377* (1-2), 80-91.

Gurney, K. (2018). *An introduction to Neural Networks*. CRC press, Boca Raton.

Hansen, D. R., & Mowen, M. M. (2007). *Managerial Accounting*. South-Western, Nashville.

Hao, J., & Ho, T. K. (2019). Machine Learning Made Easy: a Review of Scikit-Learn Package in Python Programming Language. *Journal of Educational and Behavioral Statistics*, *44* (3), 348-361.

Harrington, P. (2012). *Machine Learning in Action*. Simon and Schuster, New York.

Heerkens, H., & Winden, A. v. (2017). *Solving Managerial Problems Systematically*. Noordhoff Uitgevers, Groningen.

Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., & Meger, D. (2018). Deep Reinforcement Learning That Matters. Proceedings of the Association for the Advancement of Artificial Iintelligence Conference on artificial intelligence, 32 (1).

Horngren, C. T., Foster, G., Datar, S. M., Rajan, M., Ittner, C., & Baldwin, A. A. (2010). Cost Accounting: a Managerial Emphasis. *Issues in Accounting Education*, *25* (4), 789-790.

Huang, C., Davis, L., & Townshend, J. (2002). An Assessment of Support Vector Machines for Land Cover Classification. *International Journal of remote sensing*, *23* (4), 725-749.

James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. (2023). Linear Regression. *An Introduction to Statistical Learning: With Applications in Python*, pp. 69-134, Springer, New York.

Joseph, V. R., & Vakayil, A. (2022). SPlit: An Optimal Method for Data Splitting. *Technometrics*, *64* (2), 166-176.

Kaplan, R. S., & Anderson, S. R. (2007). *Time-Driven Activity-Based Costing: a Simpler and More Powerful Path to Higher Profits*. Harvard Business Press, Massachusetts.

Karystinos, G. N., & Pados, D. A. (2000). On overfitting, Generalization, and Randomly Expanded Training Sets. *Institute of Electrical and Electronics Engineers Transactions on Neural Networks*, *11* (5), 1050-1057.

Khalid, S., Khalil, T., & Nasreen, S. (2014). A Survey of Feature Selection and Feature Extraction Techniques in Machine Learning. 2014 Science and Information Conference,

Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised Machine Learning: A Review of Classification Techniques. *Emerging Artificial Intelligence Applications in Computer Engineering*, *160* (1), 3-24.

L'heureux, A., Grolinger, K., Elyamany, H. F., & Capretz, M. A. (2017). Machine Learning With Big Daa: Challenges and Aproaches. Institute of Electrical and Electronics Engineers *Access*, *5*, 7776-7797.

Lan, H., & Ding, Y. (2007). Price Quotation Methodology for Stereolithography Parts Based on STL Model. *Computers & Industrial Engineering, 52* (2), 241-256.

Layer, A., Brinke, E. T., Houten, F. V., Kals, H., & Haasis, S. (2002). Recent and Future Trends in Cost Estimation. *International Journal of Computer Integrated Manufacturing*, *15* (6), 499-510.

Lee, W.-M. (2019). *Python Machine Learning*. John Wiley & Sons, NY.

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., & Liu, H. (2017). Feature Selection: A Data Perspective. *Association for Computing Machinery Computing Surveys*, *50* (6), 1-45.

Li, L., Jamieson, K., Rostamizadeh, A., Gonina, K., Hardt, M., Recht, B., & Talwalkar, A. (2018). Massively Parallel Hyperparameter Tuning. *International conference on Learning Representations*.

Li, Z., Fotheringham, A. S., Oshan, T. M., & Wolf, L. J. (2020). Measuring Bandwidth Uncertainty in Multiscale Geographically Weighted Regression Using Akaike Weights. *Annals of the American Association of Geographers*, *110* (5), 1500-1520.

Loh, W. Y. (2011). Classification and Regression Trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *1* (1), 14-23.

Mahesh, B. (2020). Machine learning Algorithms - A Review. *International Journal of Science and Research*, *9* (1), 381-386.

Maxwell, A. E., Warner, T. A., & Fang, F. (2018). Implementation of Machine-Learning Classification in Remote Sensing: An Applied Review. *International Journal of Remote Sensing*, *39* (9), 2784-2817.

Meilink B.V. (2024). *Industriële Verpakkingsoplossingen*. Meilink B.V. https://meilink.com

Meyer, D., & Wien, F. (2001). Support Vector Machines. *R News*, *1* (3), 23-26.

Miotto, R., Wang, F., Wang, S., Jiang, X., & Dudley, J. T. (2018). Deep Learning for Healthcare: Review, Opportunities and Challenges. *Briefings in bioinformatics*, *19* (6), 1236-1246.

Mountrakis, G., Im, J., & Ogole, C. (2011). Support Vector Machines in Remote Sensing: A Review. *International Society for Photogrammetry and Remote Sensing Journal of Photogrammetry and Remote Sensing*, *66* (3), 247-259.

Muraina, I. (2022). Ideal Dataset Splitting Ratios in Machine Learning Algorithms: General Concerns for Data Scientists and Data Analysts. 7th International Mardin Artuklu Scientific Research Conference, 496-504.

Nakagawa, S., Johnson, P. C., & Schielzeth, H. (2017). The Coefficient of Determination R2 and Intra-Class Correlation Coefficient from Generalized Linear Mixed-Effects Models Revisited and Expanded. *Journal of the Royal Society Interface*, *14* (134).

Narong, D. K. (2009). Activity-Based Costing and Management Solutions to Traditional Shortcomings of Cost Accounting. *Cost engineering*, *51* (8), 11-22.

Nguyen, Q. H., Ly, H.-B., Ho, L. S., Al-Ansari, N., Le, H. V., Tran, V. Q., Prakash, I., & Pham, B. T. (2021). Influence of Data Splitting on Performance of Machine Learning Models in Prediction of Shear Strength of Soil. *Mathematical Problems in Engineering,* 1-15.

Nieuwenhuis, P. W. (2014). *Hoe een Borculose Stoomzagerij en Houthandel Uitgroeide Tot Een Belangrijke Speler in de Internationale Verpakkingsindustrie* (Vol. 13). Historische Vereniging Borculo.

Noble, W. S. (2006). What is a Support Vector Machine? *Nature biotechnology*, *24* (12), 1565-1567.

Nohara, Y., Matsumoto, K., Soejima, H., & Nakashima, N. (2022). Explanation of Machine Learning Models Using Shapley Additive Explanation and Application for Real Data in Hospital. *Computer Methods and Programs in Biomedicine*, *214*

Oluwagbemiga, O. E., Olugbenga, O. M., & Zaccheaus, S. A. (2014). Cost Management Practices and Firm's Performance of Manufacturing Organizations. *International Journal of Economics and Finance*, *6* (6), 234-239.

Oshan, T. M., Li, Z., Kang, W., Wolf, L. J., & Fotheringham, A. S. (2019). MGWR: A Python Implementation of Multiscale Geographically Weighted Regression for Investigating Process Spatial Heterogeneity and Scale. *International Society for Photogrammetry and Remote Sensing International Journal of Geo-Information*, *8* (6).

Park, Y.-S., & Lek, S. (2016). Artificial Neural Networks: Multilayer Perceptron for Ecological Modeling. In *Developments in Environmental Modelling* (Vol. 28) pp. 123-140. Elsevier, Amsterdam.

Popescu, M.-C., Balas, V. E., Perescu-Popescu, L., & Mastorakis, N. (2009). Multilayer Perceptron and Neural Networks. *World Scientific and Engineering Academy and Society Transactions on Circuits and Systems*, *8* (7), 579-588.

Rodriguez-Galiano, V., Sanchez-Castillo, M., Chica-Olmo, M., & Chica-Rivas, M. (2015). Machine Learning Pedictive models for Mineral Pospectivity: An Ealuation of Neural Networks, Random Forest, Regression Trees and Support Vector Machines. *Ore Geology Reviews*, *71,* 804-818.

Rodríguez, P., Bautista, M. A., Gonzalez, J., & Escalera, S. (2018). Beyond One-hot Encoding: Lower Dimensional Target Embedding. *Image and Vision Computing*, *75,* 21-31.

Ruffo, M., Tuck, C., & Hague, R. (2006). Cost Estimation for Rapid Manufacturing-Laser Sintering Production for Low to Medium Volumes. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, *220* (9), 1417-1427.

Schuh, G., & Schmidt, C. (2014). *Produktionsmanagement*. Springer, New York.

Schwertman, N. C., Owens, M. A., & Adnan, R. (2004). A Simple More General Boxplot Method for Identifying Outliers. *Computational statistics & data analysis*, *47* (1), 165-174.

Segal, M. R. (2004). Machine Learning Benchmarks and Random Forest Regression Technical Report, Center for Bioinformatics & Molecular Biostatistics, University of California, San Francisco.

Sekhar, C. R., & Madhu, E. (2016). Mode Choice Analysis Using Random Forrest Decision Trees. *Transportation Research Procedia*, *17,* 644-652.

Shehab, E., & Abdalla, H. (2002). An Intelligent Knowledge-Based System for Product Cost Modelling. *The International Journal of Advanced Manufacturing Technology* (19), 49-65.

Sontakke, S. A., Lohokare, J., Dani, R., & Shivagaje, P. (2019). Classification of Cardiotocography Signals Using Machine Learning. Intelligent Systems and Applications: Proceedings of the 2018 Intelligent Systems Conference, 2.

Srinath, K. (2017). Python – The Fastest Growing Grogramming Language. *International Research Journal of Engineering and Technology*, *4* (12), 354-357.

Sun, Y., Wang, X., & Tang, X. (2014). Deep Learning Face Representation From Predicting 10,000 Classes. Proceedings of the Institute of Electrical and Electronics Engineers Conference on Computer Vision and Pattern Recognition.

Szepesvári, C. (2022). *Algorithms for Reinforcement Learning*. Springer Nature, NY.

Tkáč, M., & Verner, R. (2016). Artificial Neural Networks in Business: Two Decades of Research. *Applied Soft Computing*, *38*, 788-804.

Uyanık, G. K., & Güler, N. (2013). A Study on Multiple Linear Regression Analysis. *Procedia-Social and Behavioral Sciences*, *106,* 234-240.

Walfish, S. (2006). A Review of Statistical Outlier Methods. *Pharmaceutical technology*, *30* (11), 82-86.
Wong, T.-T., & Yeh, P.-Y. (2019). Reliable Accuracy Estimates From K-fold Cross Validation. *Institute of Electrical and Electronics Engineers Transactions on Knowledge and Data Engineering*, *32* (8), 1586-1594.
Yang, X., Hou, L., Zhou, Y., Wang, W., & Yan, J. (2021). Dense Label Encoding for Boundary Discontinuity Free Rotation Detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 15819-15829.
Yegnanarayana, B. (2009). *Artificial Neural Networks*. PHI Learning Pvt. Ltd.
Ying, X. (2019). An Overview of Overfitting and its Solutions. *Journal of Physics: Conference Series*.
Yip, K. Y., & Gerstein, M. (2009). Training Set Expansion: an Approach to Improving the Reconstruction of Biological Networks From Limited and Uneven Reliable Interactions. *Bioinformatics*, *25* (2), 243-250.
Zhou, Z.-H. (2021). *Machine learning*. Springer Nature, New York.
Zou, K. H., Tuncali, K., & Silverman, S. G. (2003). Correlation and Simple Linear Regression. *Radiology*, *227* (3), 617-628.

# APPENDICES

## APPENDIX A - Systematic Literature Review

We used the systematic literature review method for the literature review conducted in this study. We identified relevant literature by systematically searching scientific databases with a search string. We limited the search string to title, abstract and keywords. Contents of the search string consist of keywords, publication year, subject area, and language.

```
(TITLE-ABS-KEY ( "Manufacturing" OR "Production" OR "Shop Floor" )
        AND
TITLE-ABS-KEY ( "Pricing" OR "Cost Calculation" OR "Pricing Strategy" )
        AND
TITLE-ABS-KEY ( "Time Analysis" OR "Time Discrepancy" OR "Scheduling"))
        AND
PUBYEAR > 2009 AND PUBYEAR < 2024
        AND
        ( LIMIT-TO ( SUBJAREA , "ENGI" ) OR LIMIT-TO ( SUBJAREA , "COMP" )
        OR LIMIT-TO ( SUBJAREA , "DECI" ) OR LIMIT-TO ( SUBJAREA , "MATH" )
        OR LIMIT-TO ( SUBJAREA , "BUSI" ) OR LIMIT-TO ( SUBJAREA , "CENG" )
        OR LIMIT-TO ( SUBJAREA , "ECON" ) OR LIMIT-TO ( SUBJAREA , "SOCI" )
        OR LIMIT-TO ( SUBJAREA , "MULT" ) )
        AND
( LIMIT-TO ( LANGUAGE , "English" ) )
```

Results from the search string are preferably English. Furthermore, we limited the literature review to articles published after 2010 and before 2024 due to the recent developments of this field of research.

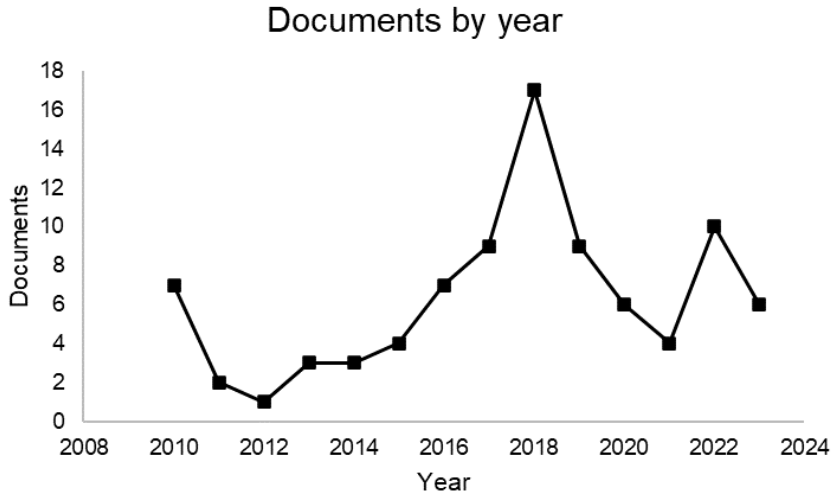



***Figure 8.1: Literature review documents by year.***

We limited subject areas to “Engineering”, “Computer Science”, “Decision Making”, “Mathematics”, “Business, Management and Accounting”, “Chemical engineering”, “Economics, Econometrics and Finance”, “Social Sciences”, and “Multidisciplinary“ due to two arguments. Firstly, to narrow down the number of results and remain specific. Secondly, to exclude subject areas such as “Medicine”, “Neuroscience”, “Biochemistry”, and other irrelevant fields. We limited search engines to Scopus, Mendeley, and Google Scholar to find literature and identify the knowledge gap. The literature selection process is illustrated in Figure 8.2.

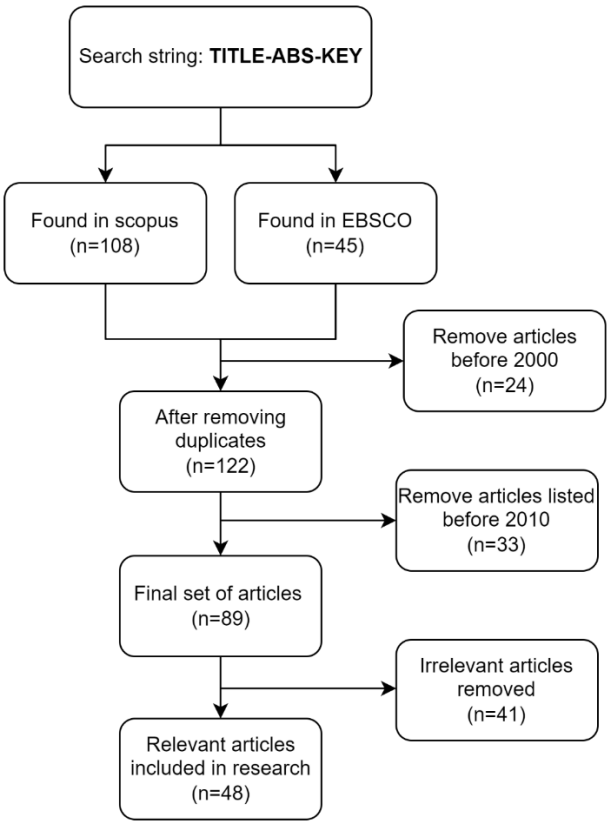



***Figure 8.2: Literature selection process.***

## APPENDIX B – Data Preparation Process

Labour times were tracked since July 2023. Every single CTO is linked to a set of features and a unique labour time. However, these two aspects must be retrieved from separate sources. In other words, the dataset first needed to be ‘mined’ as al records have to be linked to their labour time from separate databases. The two could be matched by the unique product number attached to the data. The data was gathered and combined in a macro-enabled Excel workbook. Here it could be cleaned and prepared:

1) Mine the features from the .xml file for each labour time.
2) Remove columns which hold only zeros or no values.
3) Check and replace correct delimiter (dots and commas).
4) Remove products with fewer than fifty samples.

The previously described process is illustrated in Figure 8.3. Each node of the figure contains the name of the considered file, the format, and the shape of the dataset.

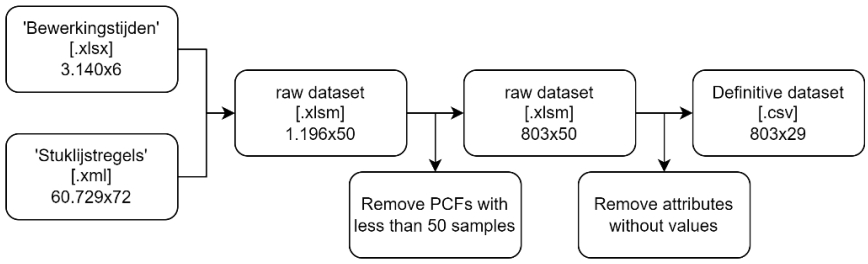



***Figure 8.3: Composition of the dataset.***

The result of the data preparation is the definitive dataset. This definitive dataset holds the samples for six PCFs manufactured from July 18th, 2023, until March 14th, 2024.

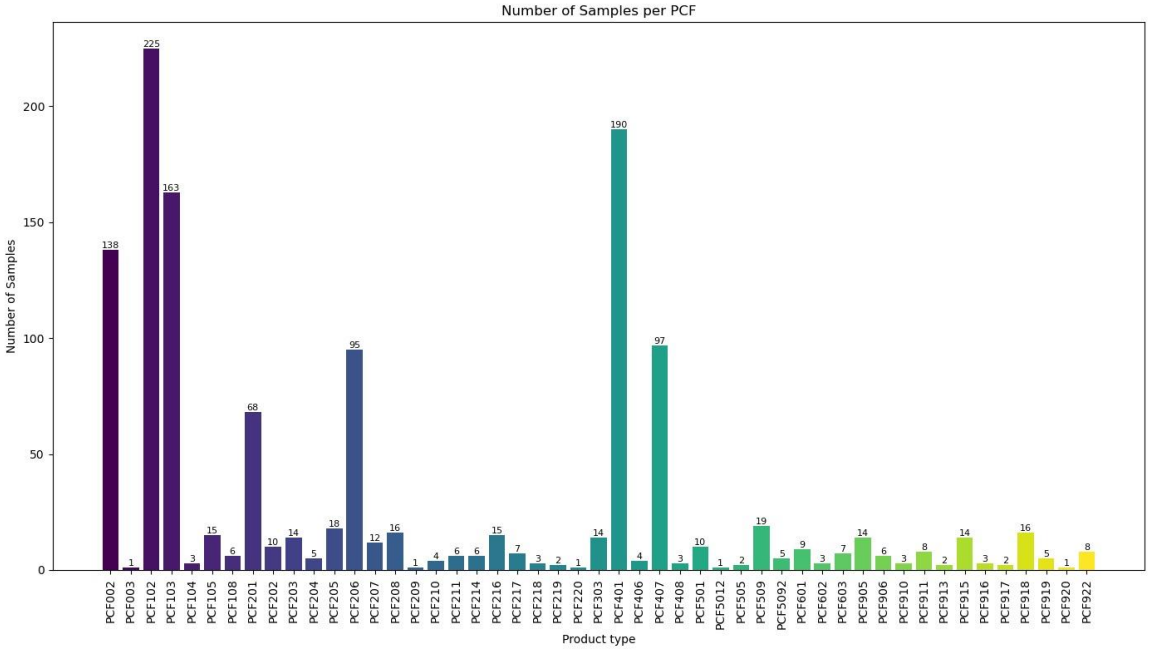



***Figure 8.4: Sample size per product type.***

All PCFs with fewer than fifty samples were removed. The presence of the features in the dataset is plotted in Figure 8.5. It indicates how common a feature is for configurated products. All Features with a presence of 0 (i.e., only contains zeros) are removed.

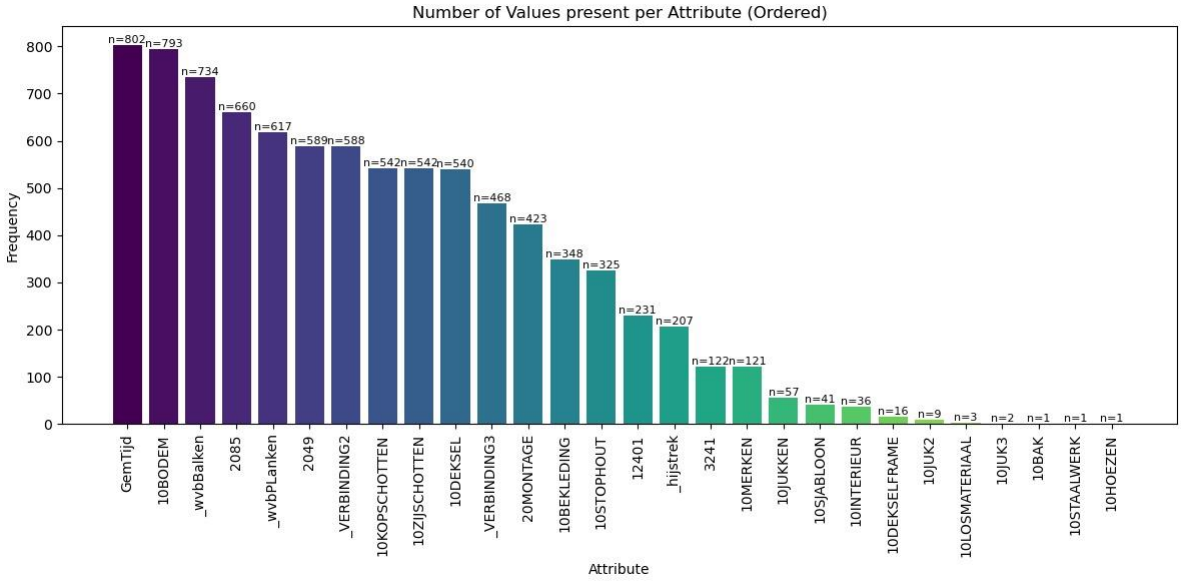



***Figure 8.5: Presence of features in the dataset.***

Separating the dataset into subsets for each PCF results in six datasets, where each sample holds twenty-seven features linked to a unique labour time. The sample size per dataset is plotted in Figure 8.6.

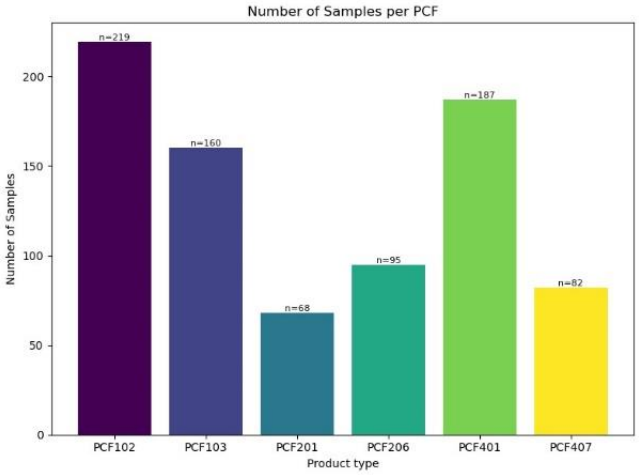



***Figure 8.6: Sample size per PCF dataset.***

After the model selection phase, the relative importance of each feature can be determined. The systematic feature engineering approach is to add two artificial features to the PCF dataset. The values of artificial features are the result of multiplying the most important features. The relative importances of each feature are attached in Appendix H. The new features are named ‘FE01’ and ‘FE02’, their values are calculated with:

$$FE01 = f_1 \times f_2$$
$$FE02 = f_1 \times f_3$$

Where:

$f_1$ = the number one important feature

$f_2$ = the two one important feature

$f_3$ = the three one important feature

## APPENDIX C - Managerial Problem Solving Method

The managerial problem solving method (MPSM) by Heerkens and Winden (2017) is one of the systematic problem solving methods developed by University of Twente. It is a flexible framework and does not require complicated prior knowledge. The authors describe the method as a combination of systematic and creative, one that looks for a practicable solution in a stepwise manner. Additionally, it considers that every problem is unique, and no problem is isolated. A problem is embedded in the context of an organisation. The MPSM is based on multiple problem-solving methods and consists of the following seven phases:

1. Defining the problem
2. Formulating the approach
3. Analysing the problem
4. Formulating (alternative) solutions
5. Choosing a solution
6. Implementing the solution
7. Evaluating the solution.

Although the phases have a sequential order, it does not exclude occasional backtracking. It is likely that the problem established in the first phase does not turn out as expected in phase three. In this case, return to the first phase and review the initial diagnosis. Furthermore, the MPSM expresses problems in terms variables to structure problems clearly and it allows for a more comprehensive visualisation. however, the author claims that the MPSM sets itself apart from the other methods by simplicity, one does not have to know the seven phases by heart.

Problems are divided into two broad categories: action problems and knowledge problems. The first is a discrepancy between the norm and reality, as perceived by the problem owner. In other words, when things are not turning out as planned or expected. The problem owner is the entity that experiences the problem, for example, a person, a group, or even an organisation. In the context of an action problem, both the norm and the reality are expressed as variables. Consequently, an action problem rarely occurs on its own, it is usually connected to other problems. Therefore, developing a problem cluster visualises the connections between problems in terms of cause and effect. The selected problem becomes the ‘core problem’.

Knowledge problems, on the other hand, deals with situations where information is missing. In this case, the problem owner does not know or understand something. Or more detailed; a knowledge problem describes a research population, the variables, and the relationships that require investigation. An example of a knowledge problem is identifying relationships between variables. Research populations consist of the person(s), group(s), or organisation(s) that are subject to investigation. Furthermore, research problems can be further divided into two subcategories: descriptive or explanatory. The first subcategory aims to know *what,* while the latter describes the *why*. Knowledge is gathered through research, the procedure recommended by the authors is the research cycle, which consists of eight phases:

1. Formulating the research goal
2. Formulating the problem statement
3. Formulating the research questions
4. Formulating the research design
5. Formulating the operationalisation
6. Performing the measurements
7. Processing the data

8. Drawing conclusions.

While the MPSM mainly focusses on action problems, it recognises that knowledge problems are unavoidable. Whenever knowledge is needed while moving through the seven phases of problem solving, move away from the MPSM and enter the research cycle. Once the research cycle is completed (and the investigation), re-enter into the MPSM at the phase that was interrupted by the knowledge problem. Proceed with the action problem with the knowledge obtained.

## APPENDIX D – Model Design

We describe the blueprint for the functionality of the model in this chapter. This blueprint contains the structure of the mathematics and computational strategies. We present the working mechanism of the model in the form of pseudo code, which is a method to describe the essential underlying actions of the programming code in regular words.

## *Pseudo Code of Models*

We divide pseudo code into multiple algorithms, Algorithm 1 represents all models except MLP. Furthermore, Algorithm 2 is the blueprint for the MLP model. Algorithm 1 is the blueprint for *Linear Regression* (LR), *Gaussian Process Regression* (GPR), *Random Forest Regression* (RGR), *Support Vector Machines* (SVM), *Decision Tree Regression* (DTR), *Gradient Boosting Regression* (GBR), *K-Nearest Neighbours* (KNN), and Extreme Gradient Boosting (XGB).

Table 8.1 contains the pseudo code for Algorithm 1. The first four steps and last two steps are the formal declarations around the mathematical application. For instance, the required libraries, definition of dictionaries and the importation of the datasets (lines 1-4), and the output of the computations are structured in a systematic way to allow for easy interpretation and comparison of results (lines 16-17).

***Table 8.1: Pseudo code for regression models.***

| | **Algorithm 1:** Pseudo code for each model = [LR, GPR, RGR, SVM, DTR, GBR, KNN, XGB] |
|---|---|
| 1: | Import necessary libraries. |
| 2: | Define an empty dictionary **performance_metrics** to store performance metrics. |
| 3: | Load all datasets and store them in a dictionary **pcf_datasets**. |
| 4: | For each PCF dataset pcf_number in **pcf_datasets**: |
| 5: | Split the dataset into features X and target variable Y. |
| 6: | Split the dataset into training and testing sets. |
| 7: | Initialise an algorithm (***model***). |
| 8: | Train the model using the training sets. |
| 9: | Make predictions using the testing set. |
| 10: | Calculate residuals. |
| 11: | Exclude outliers using the **IQR** method. |
| 12: | Calculate evaluation metrics. |
| 13: | Store the performance metrics in the **performance_metrics** dictionary. |
| 14: | Plot a scatter plot of Actual vs Predicted Build Times excluding outliers. |
| 15: | Plot a bar chart of Actual vs Predicted Build Times. |
| 16: | Adjust layout and display the plots. |
| 17: | Display the performance metrics table. |

The key elements from the loop over the six PCF datasets (indented lines 5-15) are the split in line 5, where features are distinguished from the label. Subsequently, the dataset is

split into training and testing data in line 6. The model is fitted, trained, and makes predictions in lines 7-9. Lines 10-15 are dedicated to treatment and visualisation of the results.

Algorithm 2, in Table 8.2, fundamentally differs from Algorithm 1. The initial four steps and last two steps of the Algorithms 1 and 2 are identical, the loop over the datasets distinguishes the two. The key differences are: 1) Outlier treatment, negative values are treated as outliers. 2) Data preprocessing, features are standardised by removing the mean and scaling the values to unit variance. 3) Model compilation, the model is compiled using the Adam optimiser to minimise the MSE loss function. 4) Model training, the model is trained with the training data for 100 *epochs* (iterations through all data).

***Table 8.2: Pseudo code for the MLP Artificial Neural Network.***

| | **Algorithm 2:** Pseudo code for MLP |
|---|---|
| 1: | Import necessary libraries. |
| 2: | Define an empty dictionary **performance_metrics** to store performance metrics. |
| 3: | Load all datasets and store them in a dictionary **pcf_datasets**. |
| 4: | For each PCF dataset pcf_number in **pcf_datasets**: |
| 5: | Treat negative values as outliers. |
| 6: | Split the dataset into features (X) and the target variable (Y). |
| 7: | Split the dataset into training and testing sets. |
| 8: | Standardise the features by removing the mean and scaling to unit variance. |
| 9: | Build a sequential neural network model with input layer, hidden layers, and output layer. |
| 10: | Compile the model using **Adam optimiser** and **mean squared error** loss function. |
| 11: | Train the model for 100 epochs with a batch size of 32 and 20% validation split. |
| 12: | Evaluate the model on both training and testing sets. |
| 13: | Calculate evaluation metrics including **MSE**, **RMSE**, **MAE**, **R-squared**, and **AIC**. |
| 14: | Store the performance metrics in the **performance_metrics** dictionary. |
| 16: | Plot the scatter plot of actual vs predicted values with outlier treatment. |
| 17: | Plot the trendline of the scatter plot without outliers. |
| 18: | Adjust layout and display the plots. |
| 19: | Display the performance metrics table. |

Important to note is that the outlier treatment means that the outliers are excluded from both the performance metrics and the visualisations. We attached the specific model parameters in Appendix K. The models are validated with K-fold cross validation as we described in Section 4.5. Table 8.3 contains the key steps of the code that allows us to critically assess the validity and accuracy of the models. Validation takes place after the models are selected for each of the products.

## *Pseudo Code of Model Validation*

We discussed the core principle of K-fold cross validation in Section 4.5.1. To summarise, the code iterates a 90-10 training-test split ten times. Each iteration, the dataset is divided into 90% training data and 10% test data, in such a way that every part of the dataset

is once that 10% test data. The performance is then assessed based on three parameters: MSE, AIC and the accuracy. Accuracy refers to the percentage of predictions, where the true value lies within its 95%-confidence interval. Lastly, the performance metrics are listed for each of the 10 folds along with the overall averages and the relative feature importances.

***Table 8.3: Pseudo code for K-fold cross validation.***

| **Algorithm 3:** Pseudo code for K-fold cross validation | |
|---|---|
| 1: | Load dataset for **PCF**. |
| 2: | Split dataset into features and target variable. |
| 3: | Initialise corresponding model. |
| 4: | Initialise **K-Fold** with 10 folds. |
| 5: | Initialise lists to store predictions, true values, MSE, AIC, and fold results. |
| 6: | For each fold in **K-Fold**: |
| 7: | Split dataset into training and testing sets. |
| 8: | Train the model using the training sets. |
| 9: | Extract the feature importances. |
| 10: | Make predictions using the testing set. |
| 11: | Calculate residuals. |
| 12: | Calculate evaluation metrics (MSE, AIC). |
| 13: | Store the performance metrics and fold results. |
| 14: | Calculate overall performance metrics (Overall MSE, Overall AIC, Accuracy). |
| 16: | Plot AIC and MSE for each fold. |
| 17: | Plot scatter plot of Actual vs Predicted Build Times. |
| 18: | Display performance metrics table and fold results. |
| 19: | Display list of feature importances |

In this chapter, we explained the computational construction of the models, by which we aim to predict labour cost of products. In the following chapter, we apply this model to the dataset, present the findings, and discuss the implications.

## APPENDIX E – Performance Metrics

Table 8.4 displays the complete results obtained from the experiments with PCF102, PCF103, PCF201, PCF206, PCF401, PCF407 on the machine learning models: Linear Regression (LR), Multi-Layer Perceptron (MLP), Gaussian Process Regression (GPR), Random Forest Regression (RGR), Support Vector Machines (SVM), Decision Tree Regression (DTR), Gradient Boosting Regression (GBR), K-Nearest Neighbours (KNN), and Extreme Gradient Boosting (XGB). The results from each product-model combination experiment consists of the Mean Squared error (MSE), the R-squared ($R^2$), and the Akaike Information Criterion (AIC).

***Table 8.4: Complete performance metrics.***

| **PCF102** | **LR** | **GPR** | **RFR** | **MLP** | **SVM** | **DTR** | **GBR** | **KNN** | **XGB** |
|---|---|---|---|---|---|---|---|---|---|
| MSE | 2824.88 | 2473.95 | 3694.53 | 3673 | 3245.72 | 10543 | 2413.33 | 2281.79 | 5102.57 |
| $R^2$ | 0.81 | 0.47 | -0.13 | 0.32 | 0.78 | -0.74 | 0.34 | 0.84 | -0.21 |
| AIC | 395.69 | 358.73 | 374.37 | 390.56 | 393.57 | 452.32 | 357.76 | 371.04 | 395.5 |

| **PCF103** | **LR** | **GPR** | **RFR** | **MLP** | **SVM** | **DTR** | **GBR** | **KNN** | **XGB** |
|---|---|---|---|---|---|---|---|---|---|
| MSE | 1343.02 | 1713.93 | 1000.58 | 1383.08 | 1110.22 | 1456.28 | 960.1 | 973.23 | 843.56 |
| R2 | 0.26 | 0.06 | 0.45 | 0.24 | 0.39 | -0.1 | 0.47 | 0.31 | 0.36 |
| AIC | 277.28 | 284.84 | 268.16 | 270.96 | 264.37 | 272.51 | 266.88 | 260.42 | 256.13 |
| **PCF201** | **LR** | **GPR** | **RFR** | **MLP** | **SVM** | **DTR** | **GBR** | **KNN** | **XGB** |
| MSE | 34.08 | 104.51 | 35.4 | 40.81 | 18.14 | 73.32 | 112.32 | 24.55 | 75.13 |
| $R^2$ | -0.31 | -6.14 | -0.36 | -0.68 | -0.24 | -2.25 | -3.98 | 0.06 | -2.33 |
| AIC | 96.34 | 114.44 | 96.8 | 102.21 | 91.67 | 114.13 | 120.1 | 92.41 | 114.47 |

| **PCF206** | **LR** | **GPR** | **RFR** | **MLP** | **SVM** | **DTR** | **GBR** | **KNN** | **XGB** |
|---|---|---|---|---|---|---|---|---|---|
| MSE | 522.81 | 607.88 | 332.81 | 861.79 | 660.5 | 328.26 | 770.18 | 891.73 | 342.53 |
| $R^2$ | -0.74 | -0.27 | -0.18 | -0.84 | -0.41 | -0.16 | -0.81 | -0.41 | -0.21 |
| AIC | 452.32 | 175.79 | 146.7 | 175.66 | 170.87 | 146.7 | 173.64 | 183.07 | 147.38 |

| **PCF401** | **LR** | **GPR** | **RFR** | **MLP** | **SVM** | **DTR** | **GBR** | **KNN** | **XGB** |
|---|---|---|---|---|---|---|---|---|---|
| MSE | 589.71 | 229.57 | 221.38 | 4027.7 | 656.82 | 268.45 | 278.43 | 150.88 | 285.1 |
| $R^2$ | 0.65 | 0.86 | 0.86 | -0.73 | 0.13 | 0.84 | 0.79 | 0.91 | 0.82 |
| AIC | 264.53 | 227.96 | 232.2 | 336.23 | 261.6 | 227.37 | 234.13 | 214.53 | 234.89 |

| **PCF407** | LR | GPR | RFR | MLP | SVM | DTR | GBR | KNN | XGB |
|---|---|---|---|---|---|---|---|---|---|
| MSE | 252.48 | 321.02 | 373.5 | 484.81 | 469.22 | 373.63 | 257.29 | 530.91 | 452.07 |
| $R^2$ | -0.27 | -0.2 | -0.39 | -0.81 | -0.75 | -0.88 | -0.29 | -0.98 | -0.69 |
| AIC | 136.97 | 146.34 | 148.77 | 152.94 | 152.42 | 142.85 | 137.25 | 154.39 | 151.82 |

## APPENDIX F – Normalised Performance Metrics

The performance metrics need to be normalised with min-max normalisation to calculate our composite score described in Section 4.5. it is not representative to directly calculate the weighted average due to the difference in order of magnitude in the performance metrics. The minimum and maximum metrics in Table 8.5 are represented by values of 0.000 and 1.000, respectively.

***Table 8.5: Normalised Performance Metrics.***

| **PCF102** | **LR** | **GPR** | **RFR** | **MLP** | **SVM** | **DTR** | **GBR** | **KNN** | **XGBoost** |
|---|---|---|---|---|---|---|---|---|---|
| **MSE** | 0.398 | 0.371 | 0.466 | 0.464 | 0.431 | 1.000 | 0.366 | 0.356 | 0.576 |
| **R2** | 0.981 | 0.766 | 0.386 | 0.671 | 0.962 | 0.000 | 0.684 | 1.000 | 0.335 |
| **AIC** | 0.930 | 0.884 | 0.904 | 0.924 | 0.927 | 1.000 | 0.883 | 0.900 | 0.930 |

| **PCF103** | **LR** | **GPR** | **RFR** | **MLP** | **SVM** | **DTR** | **GBR** | **KNN** | **XGBoost** |
|---|---|---|---|---|---|---|---|---|---|
| **MSE** | 0.855 | 1.000 | 0.721 | 0.871 | 0.764 | 0.899 | 0.705 | 0.710 | 0.660 |
| **R2** | 0.632 | 0.281 | 0.965 | 0.596 | 0.860 | 0.000 | 1.000 | 0.719 | 0.807 |
| **AIC** | 0.986 | 1.000 | 0.969 | 0.974 | 0.962 | 0.977 | 0.967 | 0.955 | 0.947 |

| **PCF201** | **LR** | **GPR** | **RFR** | **MLP** | **SVM** | **DTR** | **GBR** | **KNN** | **XGBoost** |
|---|---|---|---|---|---|---|---|---|---|
| **MSE** | 0.400 | 0.940 | 0.410 | 0.452 | 0.278 | 0.701 | 1.000 | 0.327 | 0.715 |
| **R2** | 0.940 | 0.000 | 0.932 | 0.881 | 0.952 | 0.627 | 0.348 | 1.000 | 0.615 |
| **AIC** | 0.888 | 0.973 | 0.890 | 0.916 | 0.866 | 0.972 | 1.000 | 0.869 | 0.973 |

| **PCF206** | **LR** | **GPR** | **RFR** | **MLP** | **SVM** | **DTR** | **GBR** | **KNN** | **XGBoost** |
|---|---|---|---|---|---|---|---|---|---|
| **MSE** | 0.698 | 0.767 | 0.542 | 0.975 | 0.810 | 0.538 | 0.900 | 1.000 | 0.550 |
| **R2** | 0.147 | 0.838 | 0.971 | 0.000 | 0.632 | 1.000 | 0.044 | 0.632 | 0.926 |
| **AIC** | 1.000 | 0.538 | 0.490 | 0.538 | 0.530 | 0.490 | 0.535 | 0.551 | 0.491 |

| **PCF401** | **LR** | **GPR** | **RFR** | **MLP** | **SVM** | **DTR** | **GBR** | **KNN** | **XGBoost** |
|---|---|---|---|---|---|---|---|---|---|
| **MSE** | 0.177 | 0.091 | 0.089 | 1.000 | 0.193 | 0.100 | 0.103 | 0.072 | 0.104 |
| **R2** | 0.841 | 0.970 | 0.970 | 0.000 | 0.524 | 0.957 | 0.927 | 1.000 | 0.945 |
| **AIC** | 0.870 | 0.803 | 0.811 | 1.000 | 0.864 | 0.802 | 0.815 | 0.779 | 0.816 |

| **PCF407** | **LR** | **GPR** | **RFR** | **MLP** | **SVM** | **DTR** | **GBR** | **KNN** | **XGBoost** |
|---|---|---|---|---|---|---|---|---|---|
| **MSE** | 0.645 | 0.732 | 0.799 | 0.941 | 0.921 | 0.799 | 0.651 | 1.000 | 0.899 |
| **R2** | 0.910 | 1.000 | 0.756 | 0.218 | 0.295 | 0.128 | 0.885 | 0.000 | 0.372 |
| **AIC** | 0.940 | 0.972 | 0.981 | 0.995 | 0.993 | 0.960 | 0.941 | 1.000 | 0.991 |

## APPENDIX G – K-fold Cross Validation Results

Table 8.6 presents our complete K-fold cross validation results and distribution of the MSE over ten folds for each of the selected models.

***Table 8.6: K-fold cross validation results.***

**PCF102 (GBR)**

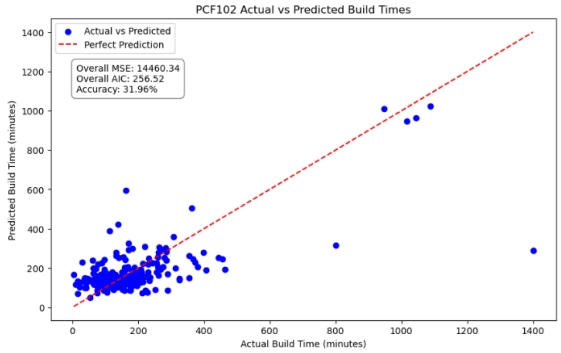
Fold Results:

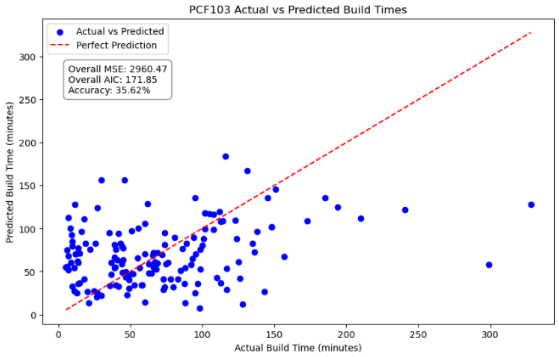
| | Fold | MSE | AIC | Accuracy |
|---|---|---|---|---|
| 0 | 1 | 13538.011955 | 267.291648 | 40.909091 |
| 1 | 2 | 4164.300866 | 241.354681 | 86.363636 |
| 2 | 3 | 11317.350633 | 263.350030 | 31.818182 |
| 3 | 4 | 11489.077393 | 263.681346 | 77.272727 |
| 4 | 5 | 18108.195354 | 273.690638 | 45.454545 |
| 5 | 6 | 11915.331709 | 264.482787 | 45.454545 |
| 6 | 7 | 62585.381276 | 300.974314 | 86.363636 |
| 7 | 8 | 6145.472165 | 249.916359 | 27.272727 |
| 8 | 9 | 7581.266318 | 254.535582 | 22.727273 |
| 9 | 10 | 4720.695852 | 235.653941 | 42.857143 |

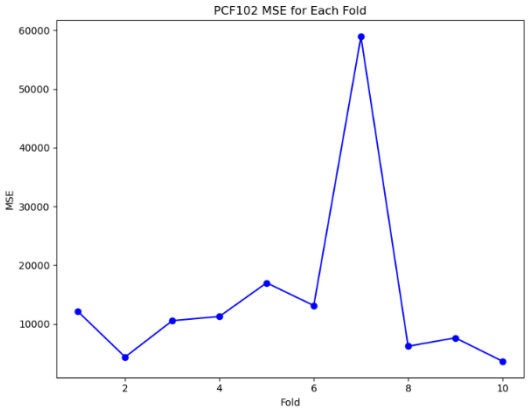
**PCF103 (XGB)**

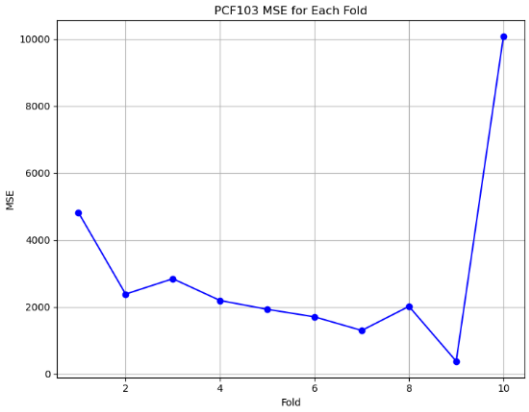
Fold Results:

| | Fold | MSE | AIC | Accuracy |
|---|---|---|---|---|
| 0 | 1 | 4817.176795 | 193.679093 | 37.50 |
| 1 | 2 | 2380.418955 | 182.400509 | 25.00 |
| 2 | 3 | 2839.231493 | 185.220619 | 43.75 |
| 3 | 4 | 2189.171187 | 181.060453 | 25.00 |
| 4 | 5 | 1925.371699 | 171.443115 | 43.75 |
| 5 | 6 | 1701.317195 | 162.148213 | 56.25 |
| 6 | 7 | 1294.415896 | 165.487222 | 56.25 |
| 7 | 8 | 2014.193966 | 179.727590 | 50.00 |
| 8 | 9 | 376.782933 | 141.043369 | 50.00 |
| 9 | 10 | 10066.572332 | 196.254633 | 6.25 |



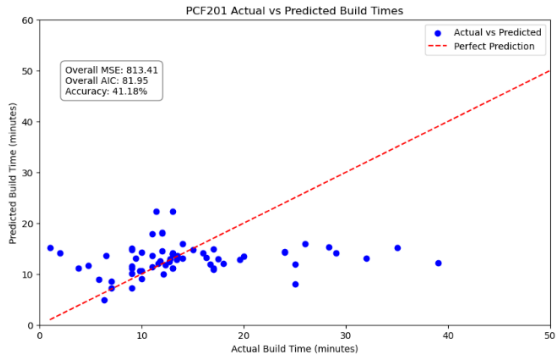



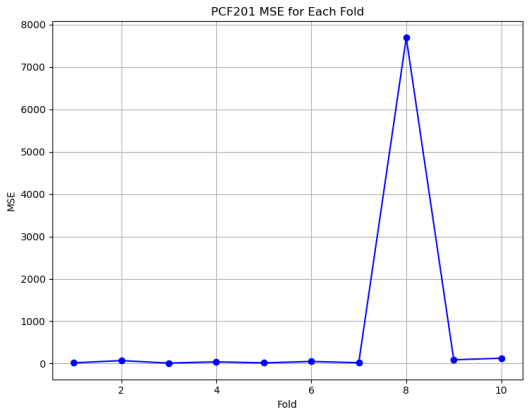



## PCF201 (SVM)

Fold Results:

| | Fold | MSE | AIC | Accuracy |
|---|---|---|---|---|
| 0 | 1 | 16.548582 | 77.644103 | 57.142857 |
| 1 | 2 | 66.932327 | 87.425774 | 57.142857 |
| 2 | 3 | 12.707157 | 75.795157 | 71.428571 |
| 3 | 4 | 40.594289 | 83.925392 | 28.571429 |
| 4 | 5 | 17.749639 | 78.134556 | 71.428571 |
| 5 | 6 | 5850.117646 | 118.719519 | 85.714286 |
| 6 | 7 | 21.843018 | 79.587169 | 42.857143 |
| 7 | 8 | 7694.091400 | 120.637456 | 42.857143 |
| 8 | 9 | 91.819725 | 85.118963 | 0.000000 |
| 9 | 10 | 135.590128 | 87.457819 | 33.333333 |

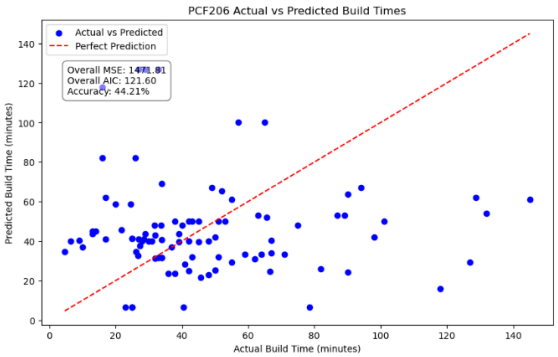


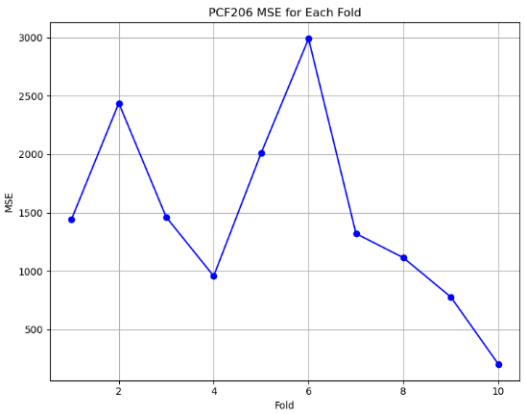




## PCF206 (DTR)

Fold Results:

| | Fold | MSE | AIC | Accuracy |
|---|---|---|---|---|
| 0 | 1 | 1441.468056 | 126.734174 | 60.000000 |
| 1 | 2 | 5359.336457 | 139.865955 | 10.000000 |
| 2 | 3 | 1461.961443 | 126.875343 | 10.000000 |
| 3 | 4 | 951.259315 | 122.577867 | 20.000000 |
| 4 | 5 | 2010.256361 | 130.060175 | 100.000000 |
| 5 | 6 | 3145.792800 | 126.484391 | 44.444444 |
| 6 | 7 | 1321.417464 | 118.678142 | 0.000000 |
| 7 | 8 | 1696.674383 | 120.927828 | 55.555556 |
| 8 | 9 | 673.222387 | 112.608681 | 55.555556 |
| 9 | 10 | 204.844136 | 101.900244 | 33.333333 |

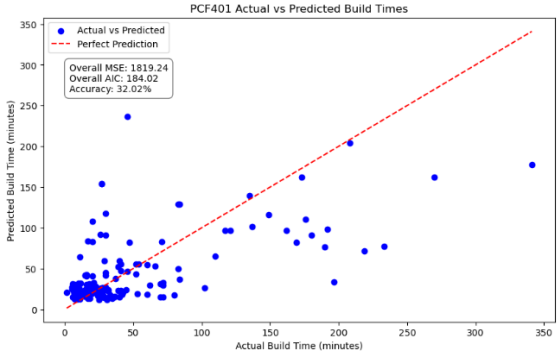


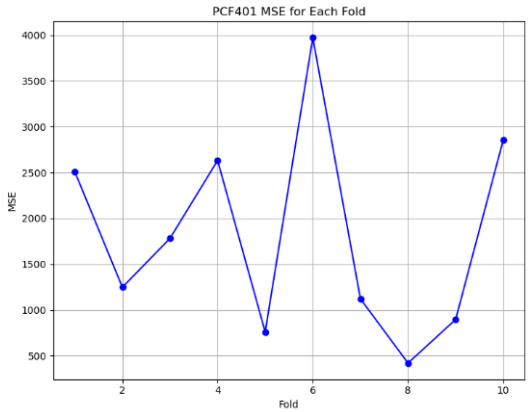



## PCF401 (KNN)

Fold Results:

| | Fold | MSE | AIC | Accuracy |
|---|---|---|---|---|
| 0 | 1 | 2223.388333 | 200.722177 | 55.555556 |
| 1 | 2 | 1423.512222 | 192.695885 | 55.555556 |
| 2 | 3 | 1794.140833 | 196.861068 | 11.111111 |
| 3 | 4 | 2666.912238 | 203.996179 | 83.333333 |
| 4 | 5 | 827.461373 | 182.930524 | 44.444444 |
| 5 | 6 | 2927.601127 | 205.674895 | 72.222222 |
| 6 | 7 | 998.779683 | 186.317616 | 44.444444 |
| 7 | 8 | 427.300680 | 171.034783 | 77.777778 |
| 8 | 9 | 773.568246 | 175.067236 | 41.176471 |
| 9 | 10 | 2892.188496 | 197.486069 | 64.705882 |

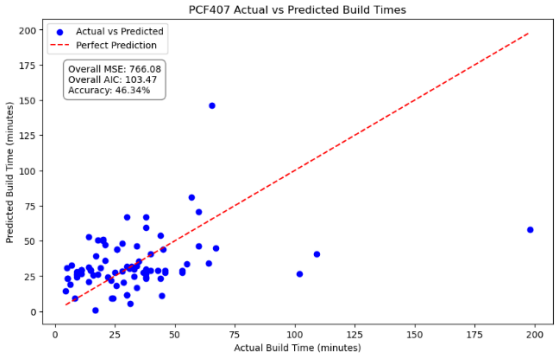


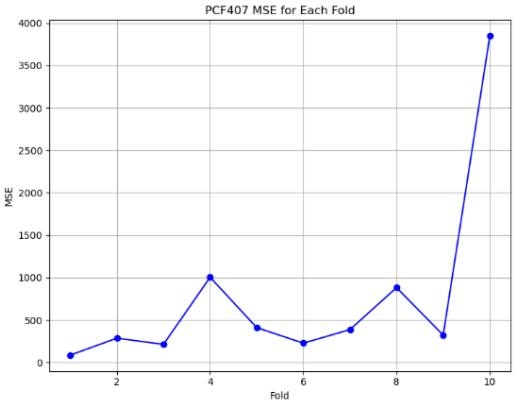




## PCF407 (LR)

Fold Results:

| | Fold | MSE | AIC | Accuracy |
|---|---|---|---|---|
| 0 | 1 | 85.650189 | 94.052443 | 66.666667 |
| 1 | 2 | 285.330822 | 104.882844 | 55.555556 |
| 2 | 3 | 212.223299 | 96.861112 | 37.500000 |
| 3 | 4 | 1003.549560 | 109.290388 | 100.000000 |
| 4 | 5 | 410.126819 | 102.131731 | 75.000000 |
| 5 | 6 | 226.533601 | 97.383146 | 12.500000 |
| 6 | 7 | 386.526207 | 101.657597 | 0.000000 |
| 7 | 8 | 881.376532 | 108.251879 | 12.500000 |
| 8 | 9 | 321.063290 | 100.173106 | 62.500000 |
| 9 | 10 | 3848.405151 | 120.043313 | 37.500000 |



## APPENDIX H – Feature Importances

Table 8.7 shows the relative feature importances per product-model combination. Each list includes the feature ID number, the feature name and its relative score.

***Table 8.7: Relative feature importances per product.***

### PCF102 (GBR)

Feature Importances:

| | Feature | Importance |
|---|---|---|
| 1 | _D2085 | 0.270895 |
| 6 | _VERBINDING3 | 0.224093 |
| 8 | _wvbPLanken | 0.102933 |
| 5 | _VERBINDING2 | 0.096765 |
| 13 | 10DEKSELFRAME | 0.071924 |
| 0 | _D2049 | 0.055061 |
| 7 | _wvbBalken | 0.034049 |
| 24 | 10STOPHOUT | 0.032048 |
| 3 | _D12401 | 0.022226 |
| 26 | 20MONTAGE | 0.018193 |
| 19 | 10KOPSCHOTTEN | 0.013895 |
| 4 | _hijstrek | 0.013657 |
| 25 | 10ZIJSCHOTTEN | 0.008061 |
| 15 | 10INTERIEUR | 0.007973 |
| 2 | _D3241 | 0.007474 |
| 12 | 10DEKSEL | 0.005087 |
| 18 | 10JUKKEN | 0.003836 |
| 20 | 10LOSMATERIAAL | 0.003795 |
| 21 | 10MERKEN | 0.003490 |
| 22 | 10SJABLOON | 0.002642 |
| 11 | 10BODEM | 0.001406 |
| 10 | 10BEKLEDING | 0.000469 |
| 16 | 10JUK2 | 0.000023 |
| 23 | 10STAALWERK | 0.000005 |
| 14 | 10HOEZEN | 0.000000 |
| 17 | 10JUK3 | 0.000000 |
| 9 | 10BAK | 0.000000 |

### PCF103 (XGB)

Feature Importances:

| | Feature | Importance |
|---|---|---|
| 19 | 10KOPSCHOTTEN | 0.350150 |
| 18 | 10JUKKEN | 0.241663 |
| 11 | 10BODEM | 0.106932 |
| 10 | 10BEKLEDING | 0.053034 |
| 12 | 10DEKSEL | 0.050318 |
| 24 | 10STOPHOUT | 0.041214 |
| 7 | _wvbBalken | 0.029838 |
| 8 | _wvbPLanken | 0.029193 |
| 1 | _D2085 | 0.028102 |
| 5 | _VERBINDING2 | 0.018997 |
| 15 | 10INTERIEUR | 0.017880 |
| 0 | _D2049 | 0.016589 |
| 22 | 10SJABLOON | 0.007178 |
| 21 | 10MERKEN | 0.006037 |
| 16 | 10JUK2 | 0.002878 |
| 23 | 10STAALWERK | 0.000000 |
| 20 | 10LOSMATERIAAL | 0.000000 |
| 25 | 10ZIJSCHOTTEN | 0.000000 |
| 13 | 10DEKSELFRAME | 0.000000 |
| 17 | 10JUK3 | 0.000000 |
| 14 | 10HOEZEN | 0.000000 |
| 9 | 10BAK | 0.000000 |
| 6 | _VERBINDING3 | 0.000000 |
| 4 | _hijstrek | 0.000000 |
| 3 | _D12401 | 0.000000 |
| 2 | _D3241 | 0.000000 |
| 26 | 20MONTAGE | 0.000000 |

### PCF201 (SVM)

Most Important Parameters:

| | Feature | Coefficient |
|---|---|---|
| 0 | _D2049 | 3.000000 |
| 7 | _wvbBalken | 0.343273 |
| 25 | 10ZIJSCHOTTEN | 0.306451 |
| 19 | 10KOPSCHOTTEN | 0.306451 |
| 16 | 10JUK2 | 0.018685 |
| 15 | 10INTERIEUR | 0.004800 |
| 26 | 20MONTAGE | 0.004000 |
| 21 | 10MERKEN | 0.002657 |
| 11 | 10BODEM | 0.002348 |
| 12 | 10DEKSEL | 0.002348 |
| 24 | 10STOPHOUT | 0.000543 |
| 18 | 10JUKKEN | 0.000201 |
| 6 | _VERBINDING3 | 0.000000 |
| 2 | _D3241 | 0.000000 |
| 23 | 10STAALWERK | 0.000000 |
| 22 | 10SJABLOON | 0.000000 |
| 20 | 10LOSMATERIAAL | 0.000000 |
| 3 | _D12401 | 0.000000 |
| 17 | 10JUK3 | 0.000000 |
| 8 | _wvbPLanken | 0.000000 |
| 4 | _hijstrek | 0.000000 |
| 5 | _VERBINDING2 | 0.000000 |
| 14 | 10HOEZEN | 0.000000 |
| 1 | _D2085 | 0.000000 |
| 10 | 10BEKLEDING | 0.000000 |
| 9 | 10BAK | 0.000000 |
| 13 | 10DEKSELFRAME | 0.000000 |

### PCF206 (DTR)

Feature Importances:

| | Feature | Importance |
|---|---|---|
| 7 | _wvbBalken | 0.261850 |
| 1 | _D2085 | 0.207438 |
| 3 | _D12401 | 0.140070 |
| 12 | 10DEKSEL | 0.125743 |
| 26 | 20MONTAGE | 0.084931 |
| 15 | 10INTERIEUR | 0.083819 |
| 8 | _wvbPLanken | 0.024865 |
| 24 | 10STOPHOUT | 0.020260 |
| 9 | 10BAK | 0.013674 |
| 25 | 10ZIJSCHOTTEN | 0.009686 |
| 21 | 10MERKEN | 0.008782 |
| 11 | 10BODEM | 0.008438 |
| 0 | _D2049 | 0.007364 |
| 19 | 10KOPSCHOTTEN | 0.001913 |
| 20 | 10LOSMATERIAAL | 0.001167 |
| 10 | 10BEKLEDING | 0.000000 |
| 14 | 10HOEZEN | 0.000000 |
| 6 | _VERBINDING3 | 0.000000 |
| 16 | 10JUK2 | 0.000000 |
| 17 | 10JUK3 | 0.000000 |
| 18 | 10JUKKEN | 0.000000 |
| 5 | _VERBINDING2 | 0.000000 |
| 22 | 10SJABLOON | 0.000000 |
| 23 | 10STAALWERK | 0.000000 |
| 4 | _hijstrek | 0.000000 |
| 2 | _D3241 | 0.000000 |
| 13 | 10DEKSELFRAME | 0.000000 |

### PCF401 (KNN SelectKBest)

Most Important Features:

| | Feature | Score |
|---|---|---|
| 0 | _D2049 | 0.07795079716742218 |
| 1 | _D2085 | 0.11013416721835462 |
| 2 | _D3241 | 0.09500905467764328 |
| 3 | _D12401 | 0.03931339110505773 |
| 4 | _hijstrek | 0.037705802970711924 |
| 5 | _VERBINDING2 | 0.1419700817993028 |
| 6 | _VERBINDING3 | 0.3707928625790211 |
| 7 | _wvbBalken | 0.05383760652012761 |
| 8 | 10BODEM | 4.7100015881142314e-05 |
| 9 | 10STOPHOUT | 0.0 |

### PCF407 (LR)

Most Important Parameters:

| | Feature | Coefficient |
|---|---|---|
| 4 | _hijstrek | 9.146945e+00 |
| 7 | _wvbBalken | 4.771402e-01 |
| 1 | _D2085 | 1.787150e-01 |
| 6 | _VERBINDING3 | 1.543366e-01 |
| 8 | _wvbPLanken | 4.479960e-02 |
| 24 | 10STOPHOUT | 9.994738e-03 |
| 11 | 10BODEM | 6.490359e-03 |
| 9 | 10BAK | 1.776357e-15 |
| 19 | 10KOPSCHOTTEN | 0.000000e+00 |
| 20 | 10LOSMATERIAAL | 0.000000e+00 |
| 21 | 10MERKEN | 0.000000e+00 |
| 13 | 10DEKSELFRAME | 0.000000e+00 |
| 22 | 10SJABLOON | 0.000000e+00 |
| 17 | 10JUK3 | 0.000000e+00 |
| 23 | 10STAALWERK | 0.000000e+00 |
| 25 | 10ZIJSCHOTTEN | 0.000000e+00 |
| 18 | 10JUKKEN | 0.000000e+00 |
| 26 | 20MONTAGE | 0.000000e+00 |
| 16 | 10JUK2 | 0.000000e+00 |
| 15 | 10INTERIEUR | 0.000000e+00 |
| 14 | 10HOEZEN | 0.000000e+00 |
| 12 | 10DEKSEL | 0.000000e+00 |
| 10 | 10BEKLEDING | 0.000000e+00 |
| 5 | _VERBINDING2 | -4.446569e-02 |
| 2 | _D3241 | -1.503962e-01 |
| 0 | _D2049 | -1.004754e+00 |
| 3 | _D12401 | -5.644708e+00 |

We condensed the feature importances from Table 8.7 to a single table in Table 8.8.

***Table 8.8: Relative feature importance per method.***

| Feature | LR | SVM | DTR | GBR | XGB |
|---|---|---|---|---|---|
| _hijstrek | 9.146945 | 0.000000 | 0.000000 | 0.013657 | 0.013657 |
| _wvbBalken | 0.477140 | 0.343273 | 0.261850 | 0.034049 | 0.034049 |
| _D2085 | 0.178715 | 0.000000 | 0.207438 | 0.270895 | 0.270895 |
| _VERBINDING3 | 0.154337 | 0.000000 | 0.224093 | 0.224093 | 0.224093 |
| _wvbPLanken | 0.044800 | 0.000000 | 0.024865 | 0.102933 | 0.102933 |
| 10STOPHOUT | 0.009995 | 0.000000 | 0.020260 | 0.032048 | 0.032048 |
| 10BODEM | 0.006490 | 0.002348 | 0.008438 | 0.001406 | 0.001406 |
| 10BAK | 0.000000 | 0.000000 | 0.013674 | 0.000000 | 0.000000 |
| 10KOPSCHOTTEN | 0.000000 | 0.306451 | 0.001913 | 0.013895 | 0.013895 |
| 10LOSMATERIAAL | 0.000000 | 0.306451 | 0.001167 | 0.003795 | 0.003795 |
| 10MERKEN | 0.000000 | 0.002657 | 0.008782 | 0.003490 | 0.003490 |
| 10DEKSELFRAME | 0.000000 | 0.002348 | 0.000000 | 0.071924 | 0.071924 |
| 10SJABLOON | 0.000000 | 0.000000 | 0.000000 | 0.002642 | 0.002642 |
| 10JUK3 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 10STAALWERK | 0.000000 | 0.000000 | 0.000000 | 0.000005 | 0.000005 |
| 10ZIJSCHOTTEN | 0.000000 | 0.000543 | 0.009686 | 0.008061 | 0.008061 |
| 10JUK2 | 0.000000 | 0.018685 | 0.000000 | 0.000023 | 0.000023 |
| 10INTERIEUR | 0.000000 | 0.004800 | 0.083819 | 0.007973 | 0.007973 |
| 10HOEZEN | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 10DEKSEL | 0.000000 | 0.002348 | 0.125743 | 0.005087 | 0.005087 |
| 10BEKLEDING | 0.000000 | 0.000000 | 0.000000 | 0.000469 | 0.000469 |
| _VERBINDING2 | -0.044466 | 0.000000 | 0.000000 | 0.096765 | 0.096765 |
| _D3241 | -0.150396 | 0.000000 | 0.000000 | 0.007474 | 0.007474 |
| _D2049 | -1.004754 | 3.000000 | 0.007364 | 0.055061 | 0.055061 |
| _D12401 | -5.644708 | 0.000000 | 0.140070 | 0.022226 | 0.022226 |

## APPENDIX I – Outlier Data

Table 8.9 presents the number of outliers removed from the predictions for each model and product with the IQR method described in Section 4.4.

***Table 8.9: Number of outliers per product and model.***

| Product | LR | GPR | RFR | MLP | SVM | DTR | GBR | KNN | XGB |
|---|---|---|---|---|---|---|---|---|---|
| **PCF102** | 1 | 5 | 5 | 4 | 2 | 2 | 6 | 3 | 4 |
| **PCF103** | 1 | 1 | 1 | 1 | 2 | 2 | 1 | 2 | 2 |
| **PCF201** | 2 | 1 | 2 | 3 | 1 | 0 | 0 | 2 | 0 |
| **PCF206** | 2 | 0 | 3 | 1 | 1 | 3 | 1 | 0 | 3 |
| **PCF401** | 3 | 4 | 3 | 6 | 4 | 3 | 4 | 4 | 4 |
| **PCF407** | 2 | 1 | 1 | 1 | 1 | 3 | 2 | 1 | 1 |

Table 8.10 presents the percentage of outliers removed from the predictions. The colour scales indicate the relatively high (yellow) and low (purple) percentages.

***Table 8.10: Percentage of outliers in predictions.***

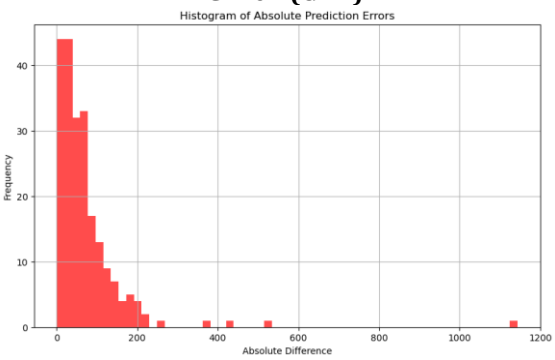
| Product | LR | GPR | RFR | MLP | SVM | DTR | GBR | KNN | XGB |
|---|---|---|---|---|---|---|---|---|---|
| **PCF102** | 2.33% | 12.82% | 12.82% | 10.00% | 4.76% | 4.76% | **15.79%** | 7.32% | 10.00% |
| **PCF103** | 3.23% | 3.23% | 3.23% | 3.23% | 6.67% | 6.67% | 3.23% | 6.67% | **6.67%** |
| **PCF201** | 16.67% | 7.69% | 16.67% | 27.27% | **7.69%** | 0.00% | 0.00% | 16.67% | 0.00% |
| **PCF206** | 11.76% | 0.00% | 18.75% | 5.56% | 5.56% | **15.79%** | 5.56% | 0.00% | 18.75% |
| **PCF401** | 9.09% | 12.50% | 9.09% | 20.00% | 12.50% | 9.09% | 12.50% | **12.50%** | 12.50% |
| **PCF407** | **13.33%** | 6.25% | 6.25% | 6.25% | 6.25% | 21.43% | 13.33% | 6.25% | 6.25% |

## APPENDIX J – Residual Analysis

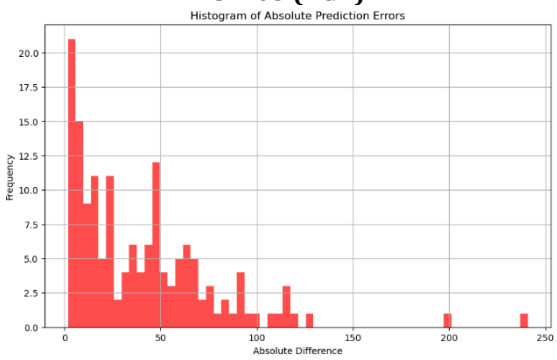
Table 5.6 in our error analysis plots the prediction's deviation from the actual value. Table 8.11 plots the absolute and the percentual errors.

***Table 8.11: Absolute error distribution in residuals.***

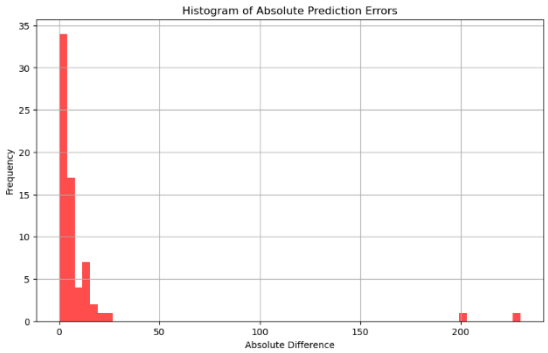
**PCF102 (GBR)**



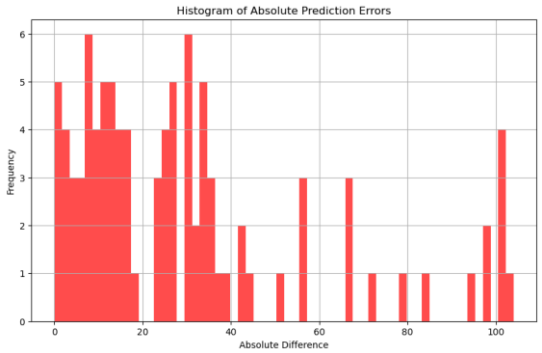

**PCF103 (XGB)**



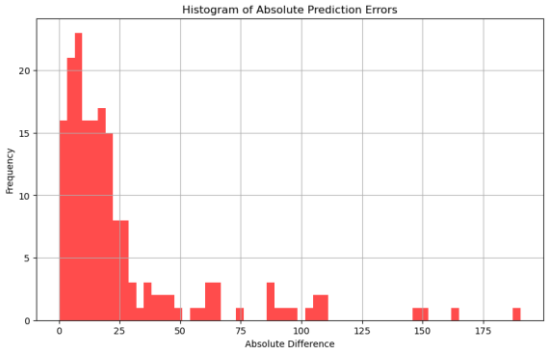

**PCF201 (SVM)**



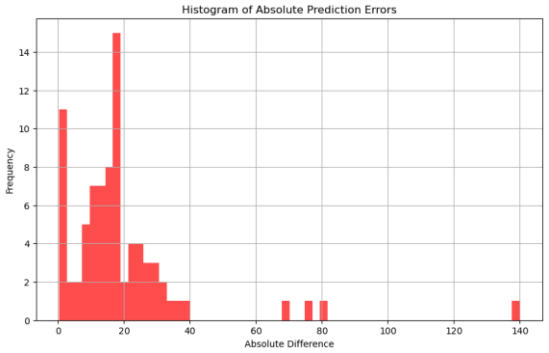

**PCF206 (DTR)**



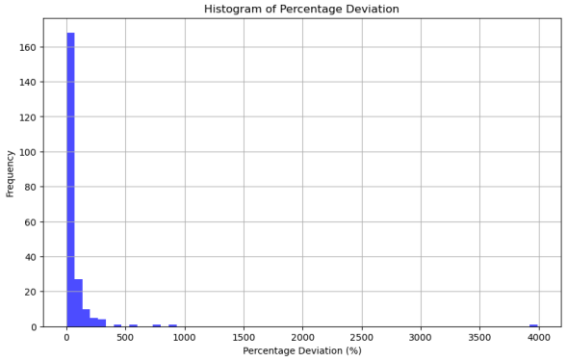

**PCF401 (KNN)**



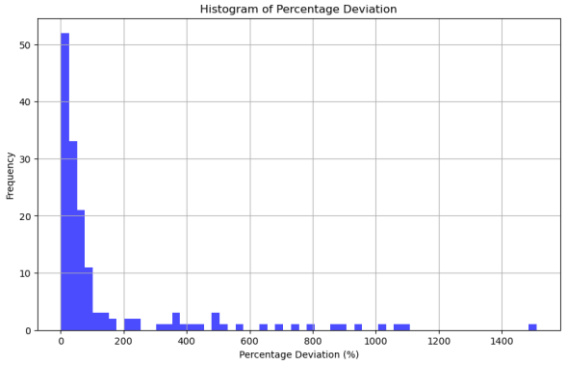

**PCF407 (LR)**



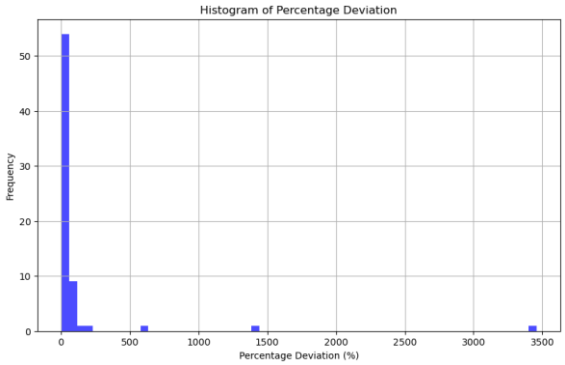
**PCF102 (GBR)**



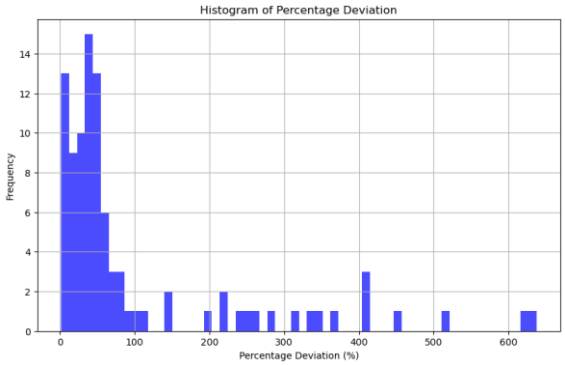

**PCF103 (XGB)**



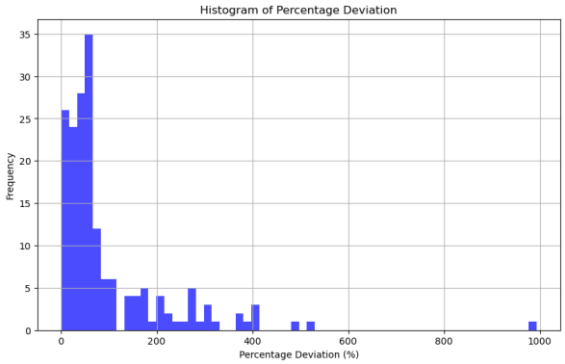

**PCF201 (SVM)**



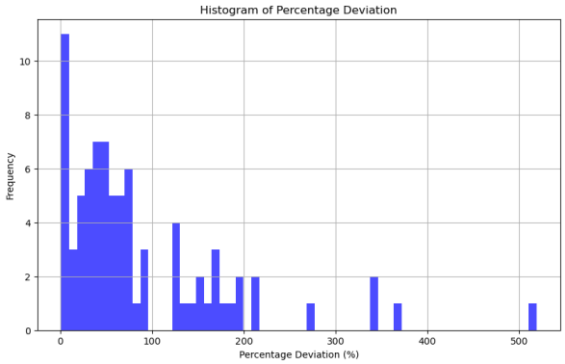

**PCF206 (DTR)**




**PCF401 (KNN)**




**PCF407 (LR)**



## APPENDIX K – Model Parameters:

Specific model parameters are set to their default values due to the comparison of the model performance. The evaluation metrics are MSE and AIC. Dataset is split into 80-20 ratio of training and test data. Outliers are excluded using the IQR method.

**Linear Regression:**

Linear relationships between independent and dependent variables are assumed, independence of errors, and constant variance of errors. No regularization techniques are applied.

**Gaussian Process Regression:**

Radial Basis Function (RBF) with length scale of 1.0 indicates that nearby points in the model have a similar effect on the prediction. White Kernel with noise level of 1e-5 which means that the noise level is relatively low. The random State is set to 42 to ensure the random numbers are equal in every run of the code.

**Random Forest Regression:**

Uses ensemble of n=100 decision trees to make predictions. Each decision tree is bult on a random subset of the dataset. The random State is set to 42 to ensure the random numbers are equal in every run of the code.

**Multi-Layer Perceptron - Artificial Neural Networks:**

Built and trained using Keras library. Standardisation is performed using Standardscaler from scikit-learn before training the model. The neural network design consists of an input layer with 64 neurons, a 32-neuron hidden layer, and an output layer. The model is trained for 100 epochs with a batch size of 32, using the Adam optimiser and MSE loss function.

**Support Vector Machines:**

A linear kernel is used as the base model. The default values are used for the parameters. Regularisation (C)=1.0, epsilon ($\varepsilon$)=0,1, and tolerance (tol)=1e-3.

**Decision Tree Regression:**

The Max Depth determines the maximum number of levels in the tree. The tree is expanded to all leaves are pure or until all leaves contain less than the minimum samples, if the parameters are not specified. The Minimal Sample Split in this case is the default value of 2. Min Samples leaves is set to the default value of 1 and determines the sample size required to be a leaf node.

**Gradient Boosting Regression:**

The loss function is optimised by least squares (LS) regression. The number of boosting stages to be performed is, by default, set to 100. The learning rate is set to 0.1 by default and shrinks the contribution of each tree. The maximum depth of the individual regression estimators is set to 3 by default. The minimum sample size required to split an internal node is set to 2 by default. The minimum sample size required to be at a leaf node is

set to 1 by default. The number of features to consider when looking for the best split considers all features by default.

**K-Nearest Neighbours:**

The number of neighbours used for prediction is set to the default value of 5. The weight function used in prediction is set to 'uniform' by default, therefore, all points in each neighbourhood are weighted equally. The algorithm used to compute the nearest neighbours is set to 'auto'. It selects the most the best fitting algorithm based on the requirements.

**Extreme Gradient Boosting:**

The *objective function* is the loss function that is minimised, by default is it set to 'reg:squarederror' for regression problems. The *step size shrinkage* is a parameter used to prevent overfitting. This parameter is set to its default value of 0.3. *Number of Trees* is set to the default value of 100. The *maximum depth of each tree* is set to 6, which is the default value. The minimum *sum of weights* of all observations required in a '*child'* is set to the default value of 1. Subsample is set to the default of 1. This is the fraction of observations to be randomly sampled for each tree. *Colsample Bytree* is the parameter that indicates the fraction of features to be randomly sampled for each tree. It is set to the default value of 1, which means that all features are used.

**Summary of hyperparameters for remaining models after model selection:**

| PCF | METHOD | Hyperparameters |
|---|---|---|
| 102 | GBR | Boosting stages = 100 |
| | | Learning rate = 0.1 |
| | | maximum depth of the individual regression estimators = 3 |
| | | minimum number of samples split node = 2 |
| | | minimum number of samples required for leaf node = 1. |
| 103 | XGB | step size shrinkage = 0.3 |
| | | Number of Trees = 100 |
| | | maximum depth of each tree = 6 |
| | | minimum sum of weights = 1 |
| | | Subsample = 1 |
| | | Colsample Bytree = 1 |
| 201 | SVM | Regularisation (C) = 1.0 |
| | | epsilon (ε) = 0,1 |
| | | tolerance (tol) = 1e-3. |
| 206 | DTR | The Minimal Sample Split = 2 |
| | | Min Samples leaves = 1 |
| 401 | KNN | number of neighbours = 5 |
| | | The weight function = uniform |
| 407 | LR | n/a |