Machine Learning Techniques for Cost Estimation of Engineer-to-Order Products

Filipa Manuela Carvalho Santos

Dissertação de Mestrado

Orientador na FEUP: Professor Américo Azevedo

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Mestrado em Engenharia e Gestão Industrial

2025-06-27

I learned that courage was not the absence of fear, but the triumph over it.
Nelson Mandela

Resumo

A estimativa de custos em ambientes *engineer-to-order* representa um desafio significativo, devido à elevada variabilidade dos projetos, à escassez de dados históricos e à forte dependência em conhecimento especializado. Neste contexto, as técnicas de *machine learning* surgem como uma alternativa promissora, com potencial para melhorar a precisão e a consistência das estimativas. Esta dissertação explora a aplicabilidade de modelos de *machine learning* à previsão de custos em ambientes *engineer-to-order*, com base em bases de dados sintéticas que simulam diferentes níveis de estrutura e complexidade.

A estratégia metodológica adotada envolveu a criação de quatro bases de dados sintéticas, combinando variáveis categóricas e numéricas, com diferentes graus de organização. Foram implementados quatro algoritmos: regressão linear, *random forest*, *XGBoost* e *artificial neural networks*. Estes algoritmos foram avaliados com e sem afinação de hiper parâmetros.

Os resultados demonstram que os modelos mais avançados, como *XGBoost* e a*rtificial neural networks*, alcançam desempenhos excecionais quando os dados são estruturados, enquanto modelos mais simples, como a regressão linear, revelam maior robustez em contextos mais aleatórios. A afinação de hiper parâmetros mostrou ser útil em cenários menos estruturados, embora com ganhos marginais quando a estrutura dos dados é elevada.

A nível académico, este trabalho contribui com evidência empírica sobre o desempenho de diferentes algoritmos em contextos industriais simulados. Do ponto de vista prático, propõe-se um percurso de adoção progressiva de *machine learning* para empresas com menor maturidade digital, começando com dados sintéticos e modelos simples, e evoluindo para abordagens mais sofisticadas. Esta investigação oferece, assim, uma base sólida para a adoção gradual e realista de técnicas de *machine learning* na estimativa de custos em ambientes *engineer-to-order*.

Machine Learning Techniques for Cost Estimation of Engineer-to-Order Products

Abstract

Cost estimation in engineer-to-order environments presents a significant challenge due to the high variability of projects, limited historical data, and strong reliance on expert judgement. In this context, machine learning techniques emerge as a promising alternative, with the potential to improve both the accuracy and consistency of estimates. This thesis explores the applicability of machine learning models to cost prediction in engineer-to-order settings, based on synthetic datasets designed to simulate varying levels of structure and complexity.

The methodological strategy involved the creation of four synthetic datasets, combining categorical and numerical variables with different degrees of organisation. Four algorithms were implemented: linear regression, random forest, XGBoost, and artificial neural networks. These models were evaluated both with and without hyperparameter tuning.

The results show that advanced models such as XGBoost and artificial neural networks achieved exceptional performance when applied to structured data, while simpler models like linear regression proved more robust in more randomised scenarios. Hyperparameter tuning was shown to be useful in less structured contexts, although it provided only marginal gains when data organisation was already high.

From an academic perspective, this work contributes empirical evidence on the performance of different algorithms in simulated industrial contexts. From a practical standpoint, it proposes a progressive adoption path for machine learning in companies with lower levels of digital maturity, starting with synthetic data and simple models, and evolving towards more sophisticated approaches. This research therefore provides a solid foundation for the gradual and realistic implementation of machine learning techniques in cost estimation for engineer-to-order environments.

Agradecimentos

Chegar até aqui não foi apenas um objetivo académico, mas também pessoal. Terminar esta dissertação representa muito mais do que concluir um ciclo de estudos. É, acima de tudo, uma conquista emocional, fruto de resiliência num percurso com desafios inesperados.

Agradeço ao meu orientador, Professor Américo Azevedo, pela orientação prestada ao longo deste trabalho, bem como pelos comentários e sugestões que contribuíram para o seu desenvolvimento.

Agradeço também à empresa parceira pela colaboração estabelecida, nomeadamente pela disponibilização das variáveis utilizadas na construção dos conjuntos de dados sintéticos. Esse contributo foi essencial para dar contexto prático ao trabalho desenvolvido.

Agradeço à Sílvia por ter caminhado comigo no início. Ao Diogo, agradeço por me ter ajudado a chegar ao fim. Contra todas as minhas expectativas, depender de mim funcionou.

Por fim, agradeço à minha família e a todos os que, de forma direta ou indireta, me apoiaram ao longo deste processo.

Table of Contents

List of Acronyms

ANN – Artificial Neural Networks;

ETO – Engineer-to-Order;

KH – Know-How Reuse;

LM – Labour to Material Ratio;

LR – Linear Regression;

MAE – Mean Absolute Error;

MAPE – Mean Absolute Percentage Error;

ML – Machine Learning;

PD – Project Duration;

PF – Project Familiarity;

RF – Random Forest;

RMSE – Root Mean Squared Error;

SF – Number of System Features;

SVM – Support Vector Machines;

TC – Technological Complexity;

XGBoost – Extreme Gradient Boosting.

List of Figures

List of Tables

1 Introduction

In recent years, cost estimation for engineer-to-order (ETO) products has become increasingly challenging. Unlike standardised manufacturing environments, ETO contexts involve high variability, customised specifications, and limited historical data, which make traditional estimation techniques less reliable. These methods often rely on expert judgement or past experience, introducing subjectivity and inconsistency into the process.

To address these issues, machine learning (ML) has gained attention as a promising alternative. ML models are capable of processing large datasets, identifying hidden patterns, and generating accurate cost predictions even in complex and variable settings. Several studies have explored how ML techniques can support early cost estimation and reduce dependence on manual input (Mandolini et al. 2024).

These findings suggest that ML could improve decision-making in ETO companies, allowing better resource planning, quoting, and pricing strategies. This thesis aligns with previous research by examining how different ML algorithms perform in various structured and unstructured data scenarios related to ETO product cost estimation.

1.1 Context

ETO manufacturing has become increasingly relevant as companies seek to deliver tailored solutions that respond to specific customer needs. Unlike standardised production environments, ETO settings are characterised by high variability, longer lead times, and non-repetitive design processes, all of which complicate both operational planning and cost estimation (Gosling and Naim 2009).

One of the key challenges in these environments is the reliable estimation of production costs. Traditional methods, such as expert judgement and historical analogies, may be effective in repetitive manufacturing contexts but often prove inadequate in ETO products, where uncertainty and uniqueness are dominant. Inaccurate estimates can result in mispricing, budget deviations, or lost business opportunities (Caputo and Pelagagge 2008; Elmousalami 2021).

To address these challenges, ML has gained attention as a potentially valuable tool. ML techniques are capable of uncovering patterns in complex datasets and generating cost predictions that are faster, more consistent, and potentially more accurate than traditional approaches (Elmousalami 2021; Mandolini et al. 2024).

This thesis focuses on the problem of cost estimation for ETO products, exploring whether ML techniques can offer a viable alternative to traditional methods. While the project was not conducted within a company, a collaboration was established with a Portuguese industrial automation firm that contributed a set of input variables typically used during early-stage cost estimation. These variables served as the basis for synthetic dataset creation. The work aims to provide insights that can help organisations operating in variable production environments make more informed, data-driven decisions.

1.2 Challenge and Objectives

Estimating costs in ETO environments is particularly difficult due to data scarcity and process variability, which limit the reliability of traditional methods. This thesis investigates whether ML techniques can provide a more accurate and consistent alternative.

The core research question that guides this work is:

"Can machine learning techniques be used effectively to estimate the cost of engineer-to-order products?"

The overall aim of this thesis is to examine whether ML can reduce uncertainty, improve estimation accuracy, and provide a data-driven alternative to manual cost estimation methods. To achieve this, the following specific objectives were defined:

- Conduct a critical review of the literature on cost estimation techniques in ETO settings;
- Construct synthetic datasets based on industry-provided input variables, using both structured (heuristic) and unstructured (random) generation approaches;
- Implement and train multiple ML algorithms on the constructed datasets;
- Evaluate and compare model performance using standard regression metrics;
- Analyse the impact of hyperparameter tuning on model accuracy;
- Discuss the practical implications of the findings for real-world ETO manufacturers.

In line with these objectives, this thesis seeks to address the following research questions:

- RQ1: How can synthetic datasets be constructed to simulate cost estimation scenarios in ETO environments?;
- RQ2: Which ML models are most appropriate for estimating costs in the context of ETO products, using synthetic data?;
- RQ3: What are the limitations and strengths of using ML models in this context, particularly in the absence of real historical cost data?

Together, these objectives and research questions aim to provide a structured assessment of the potential and limitations of ML techniques in a domain where traditional estimation methods often fall short.

1.3 Methodology

The methodology followed in this thesis was designed to evaluate the applicability of ML techniques to cost estimation for ETO products. Due to the lack of real-world historical data, the research relied on a fully synthetic dataset based on six variables provided by the partner company.

Four datasets were created by combining two design dimensions: structure (rule-based vs. random) and variable type (categorical vs. mixed). This setup enabled a comparison of model performance in structured versus unstructured conditions. For each dataset, a synthetic cost value was calculated using a weighted scoring function, followed by a non-linear transformation to reflect realistic cost scaling in complex projects.

Four ML models were selected and implemented: linear regression (LR), random forest (RF), extreme gradient boosting (XGBoost), and artificial neural networks (ANN). Each model was evaluated with and without hyperparameter tuning to assess the impact of optimisation strategies. Model performance was measured using three standard regression metrics: R^2, mean absolute error (MAE), and root mean squared error (RMSE).

This methodological framework was developed to systematically address the research questions, and to explore the extent to which ML can outperform traditional estimation methods under varying data conditions.

1.4 Thesis Structure

This thesis is divided into seven chapters, providing a structured analysis of the use of ML for cost estimation in ETO environments.

Chapter 1 introduces the topic, outlines the motivation for the research, and presents the main challenges, research questions, objectives, and a summary of the methodology.

Chapter 2 reviews the relevant literature, focusing on traditional cost estimation methods and recent ML applications. It highlights the limitations of existing approaches in ETO and supports the need for alternative solutions.

Chapter 3 outlines the industrial context and core estimation challenges in ETO production. It introduces the input variables supplied by the industrial partner and explains how synthetic datasets were designed to reflect real-world constraints.

Chapter 4 details the methodological approach, including the generation of synthetic datasets, the selection of four ML models, the tuning strategies applied, and the metrics used for performance evaluation.

Chapter 5 presents the experimental results across the four datasets. It compares model performance with and without hyperparameter tuning, showing how data structure and optimisation affect accuracy.

Chapter 6 discusses the main findings, compares algorithms, and explores their potential applicability to real-world ETO cost estimation tasks.

Finally, Chapter 7 concludes the thesis with a summary of contributions, a reflection on limitations, and suggestions for future research, particularly regarding the use of real industrial data.

2 State of the Art

This chapter looks at different ways to estimate costs in ETO environments. Section 2.1 explains traditional methods like expert judgement, parametric models, and analytical techniques, and shows their main problems in ETO. Section 2.2 talks about how ML can be used for cost estimation, including the types of models, their benefits, and what to think about when using them. Section 2.3 talks about data problems in ETO and how synthetic datasets can help when real data is not available. Finally, Section 2.4 sums up the main points and explains why this research was done.

2.1 Traditional Approaches to Cost Estimation

Cost estimation in ETO environments has traditionally relied on a combination of expert judgement and quantitative techniques. These methods include qualitative assessments, parametric models, and analytical approaches, each with distinct advantages and limitations. Although widely used in industry, traditional methods often struggle to produce consistent results when data is scarce, or project conditions vary. This section reviews the main traditional techniques applied in ETO cost estimation and discusses the practical challenges that have led to growing interest in ML as a more adaptable and data-driven alternative.

2.1.1 Classification of traditional techniques

Traditional cost estimation methods are usually divided into two groups: qualitative and quantitative. Qualitative methods include expert judgement, basic rules, and experience-based knowledge. These are useful at early stages of a project, when not much information is available, and decisions must be made quickly. But their results depend a lot on the person doing the estimation, which can lead to different outcomes in different situations (Matel et al. 2022; Özcan and Fığlalı 2014).

Quantitative methods use clear and measurable data. Some common examples are analogous, parametric, and analytical estimation. Analogous estimation means using costs from similar past projects to predict new ones (Caputo and Pelagagge 2008). However, this only works well if the old and new projects are truly similar. While these methods can be more consistent, they need historical data, which is often missing in ETO environments.

2.1.2 Parametric Cost Estimation

Parametric models estimate cost by linking measurable features of a project, called cost drivers, to the final cost. Common cost drivers include weight, size, or processing time. These models use mathematical formulas, like regression equations, to make predictions (Caputo and Pelagagge 2008; Mandolini et al. 2024).

Parametric methods are often preferred in early design stages because they can provide quick and reasonably accurate estimates without requiring detailed design drawings (Gunduz, Ugur, and Ozturk 2011). However, they depend on the availability of reliable historical data, which may not exist when developing new or innovative products (Anwar and Colliander 2023). Additionally, regression-based techniques typically require the relationship between variables to be defined in advance, which can limit their flexibility (Verlinden et al. 2008).

2.1.3 Analogous and Bottom-Up Methods

Analogous cost estimation relies on using data from previous, similar projects to predict the cost of a new one. This method assumes a functional or geometrical similarity between the current product and a past one, adjusting the cost based on observable differences (Caputo and

Pelagagge 2008; Özcan and Fığlalı 2014). It is often used in contexts where quick estimates are needed and detailed product data is not yet available. However, its accuracy depends heavily on correctly identifying a truly comparable reference project, which can be challenging in ETO environments where products are highly customized (Padala and Goyal 2025). Additionally, analogous methods do not explicitly model the manufacturing process or cost structure, making them less transparent and harder to generalize for novel or innovative designs (Caputo and Pelagagge 2008).

On the other hand, analytical or bottom-up approaches involve decomposing a product or system into smaller components, each of which is estimated individually based on its material, labour, and overhead requirements. The total cost is then obtained by summing the contributions of all parts (Caputo and Pelagagge 2008). These methods are often considered the most accurate when detailed product and process data is available.

2.1.4 Limitations and Industry Practice

Despite the development of more formal techniques, many companies still rely on expert-based or hybrid approaches. Caputo and Pelagagge (2008) pointed out that even when quantitative methods are available, they are often underused due to the lack of accessible tools and missing data. More recent studies confirm that these challenges persist (Matel et al. 2022).

For example, Özcan and Fığlalı (2014) observed that in stamping die production, companies revert to expert judgement when CAD data is incomplete. This practice illustrates a recurring issue in industry, where reliance on expert input persists despite the availability of advanced estimation tools. Matel et al. (2022) also noted that during early phases of planning, cost engineers often rely heavily on experience due to insufficient data, reinforcing the need for methods that can function under uncertainty.

While traditional techniques remain widely used, their limitations, such as subjectivity, poor scalability, and limited adaptability, have motivated researchers and practitioners to explore more flexible and data-driven alternatives, particularly those based on ML.

2.2 Machine Learning Techniques for Cost Estimation

In recent years, ML has emerged as a promising alternative to traditional cost estimation methods, particularly in ETO contexts where data is often limited, unstructured, or highly variable. The increasing demand for faster and more accurate estimates has driven research and industry interest towards data-driven techniques capable of capturing hidden relationships between inputs and cost outcomes. This section reviews key motivations behind the use of ML for cost estimation, the main types of models employed, and practical considerations such as feature selection, implementation challenges, and comparative performance.

2.2.1 Motivations for ML in Cost Estimation

Traditional techniques often struggle in environments characterised by data complexity and limited standardisation. In contrast, ML methods excel at identifying patterns in large, multi-dimensional datasets and can improve over time as new data becomes available (Mandolini et al. 2024). These capabilities make ML particularly attractive for early-stage estimation, where reliable predictions are needed despite limited available information.

Several studies have demonstrated that ML can outperform expert judgement in various domains, including automotive, construction, and manufacturing. For example, Hammann (2024) showed that gradient-boosted models outperformed human experts in estimating the total cost of complete passenger cars when applied to industrial-scale datasets. However, at

more detailed levels (e.g. assembly components), expert judgement remained more accurate, suggesting that ML offers strong advantages primarily in higher-level estimation tasks.

2.2.2 Overview of Machine Learning Models Used

A wide range of ML models have been explored in the context of cost estimation, including regression techniques, ANNs, and tree-based ensemble methods such as RF and XGBoost.

Among these, ANNs are frequently used due to their ability to approximate complex non-linear functions. They have been successfully applied in construction (Meharie and Shaik 2020), tooling for sheet metal forming (Özcan and Fığlalı 2014), and mechanical manufacturing (Ziegler, Dobhan, and Heusinger 2022). In ETO contexts, Ziegler et al. (2022) showed that ANNs can support tool selection and quotation costing by learning from historical production data. In their study, an ANN was implemented by a manufacturing firm to automate grinding tool selection, while the quotation costing model, still under evaluation, showed promising accuracy. These applications highlight the potential of ANNs to improve estimation and enable further process automation.

Tree-based ensemble methods, particularly RF and XGBoost, have demonstrated robust performance due to their ability to manage high-dimensional data and feature interactions. In a comparative study, Elmousalami (2021) evaluated 20 artificial intelligence models for conceptual cost estimation and identified XGBoost as the most accurate model based on mean absolute percentage error (MAPE) and adjusted R^2 scores. Similarly, in the construction domain, RF was found to outperform both ANNs and SVMs in highway project cost prediction, delivering higher accuracy across multiple performance metrics (Meharie and Shaik 2020).

2.2.3 Feature Selection and Data Processing

Feature selection is a common step in ML pipelines, aiming to reduce input dimensionality, improve model performance, and enhance interpretability. In the cost estimation domain, several studies have applied feature selection techniques to isolate key cost drivers and improve prediction accuracy. For instance, Rapaccini et al. (2023) implemented an iterative feature selection process to identify relevant variables during early ETO project stages.

In the present research, formal feature selection was not applied. This decision reflects the nature of the dataset used, which was synthetically generated based on a small set of predefined input variables provided by the industrial partner. These variables were already selected for their relevance to cost estimation, making additional selection unnecessary in this context.

2.2.4 Benefits and Challenges in ETO Environments

The application of ML in ETO cost estimation presents both opportunities and challenges. ETO environments often involve limited historical data, high variability, and loosely structured inputs, which pose difficulties for traditional and ML methods alike. To address these limitations, techniques such as data augmentation have been proposed to expand sparse datasets and support early-stage estimation efforts, even when input data is limited (Mandolini et al. 2024).

However, challenges remain. These include:

- The need for sufficient, clean, and representative training data;
- The difficulty of integrating ML models into existing engineering workflows;
- A lack of ML literacy among cost engineers and decision-makers;
- The “black-box” nature of some algorithms, which limits interpretability and trust.

These obstacles highlight the importance of developing robust, accessible, and transparent ML solutions tailored to the specific constraints of ETO settings.

2.2.5 Comparative Studies and Performance Insights

Comparative studies are essential to assess the practical effectiveness of ML models in cost estimation tasks. Elmousalami (2021), for example, found that ensemble-based methods consistently outperformed simpler models in scenarios involving non-linear relationships and uncertainty among cost drivers. These findings support the preference for more robust algorithms in complex estimation tasks.

In addition, Mandolini et al. (2024) presented a hybrid method that integrates parametric software with ML to automatically generate training datasets, achieving better accuracy and broader applicability in early cost estimation for axial compressors, a particularly relevant contribution to ETO contexts.

2.3 Data Challenges and Use of Synthetic Datasets

One of the most significant limitations when applying ML techniques to cost estimation in ETO environments is the availability and quality of data. Unlike mass production settings where large volumes of structured historical data can be extracted from enterprise systems, ETO companies often deal with one-off products and fragmented data repositories. This poses unique challenges for data-driven modelling and has driven some researchers and practitioners to explore the use of synthetic data generation.

2.3.1 Data Scarcity in ETO Contexts

In ETO settings, collecting sufficient, structured, and consistent data is often a challenge. Many firms rely on spreadsheets, handwritten notes, and fragmented databases, which makes it difficult to develop fully data-driven estimation models. As a result, ML applications in these environments frequently face data scarcity, both in terms of quantity and quality.

In response to these constraints, synthetic data generation has been explored in related domains as a practical workaround. For instance, Liu et al. (2025) developed a synthetic dataset to train an ANN for predicting soil liquefaction susceptibility, using parameter ranges derived from established geotechnical methods. While not about cost, this shows how fake data can be helpful when real data is missing.

2.3.2 Reflection on the Present Research

In the context of the present research, the lack of any historical dataset prompted the development of a fully synthetic dataset based solely on the variables provided by the industrial partner. These variables were structured into a format suitable for training regression models, and artificial values were generated to simulate realistic but fictional projects. Care was taken to design the dataset in a way that reflects plausible cost-driving behaviour, guided by engineering logic and practical constraints.

This approach, while limited, was deemed appropriate for testing the feasibility of ML techniques for cost estimation in ETO settings. However, the findings should be interpreted with caution, and future work will aim to validate the models against actual industrial data as it becomes available.

2.4 Summary and Research Motivation

This chapter has reviewed the main approaches to cost estimation, with an emphasis on both traditional techniques and ML-based methods, particularly in the context of ETO manufacturing. Traditional techniques continue to be widely used in industry. However, they

often face limitations when applied to highly customised projects, especially during early design stages where detailed information is scarce.

Recent research suggests that ML methods can offer improvements in predictive accuracy and efficiency, particularly in complex industrial environments. Algorithms such as ANNs, RFs, and gradient boosting machines have been applied successfully in a variety of engineering cost estimation tasks. In ETO settings, their ability to handle non-linear relationships and high-dimensional inputs makes them especially promising.

Nonetheless, the application of ML in real industrial ETO scenarios remains limited by one key factor: data availability. In many cases, including the present research, the companies involved are unable to provide a dataset suitable for training predictive models. While this limitation is common in ETO contexts, it nonetheless presents a significant obstacle to the development of reliable ML tools.

As a result, this research adopts a fully synthetic dataset, built from scratch using the variables identified by the company as relevant to cost estimation. The synthetic data is not intended to replicate real operations, but rather to allow the structured testing of ML models under controlled assumptions.

The central aim of this research is therefore to assess the effectiveness of ML techniques in estimating project costs within the context of ETO production for an industrial automation company. Even though the data is synthetic, the results give useful insights and can guide future research in settings with limited data.

3 The Cost Estimation Challenge in Custom Manufacturing

This chapter presents the industrial background, and the problem studied in this thesis. Section 3.1 explains the main cost estimation challenges in ETO environments and describes the industrial partner's activity. Section 3.2 presents the set of input variables used in this research and explains how synthetic datasets were constructed to simulate realistic project scenarios under varying data conditions. Finally, Section 3.3 reflects on the importance of this research for other companies with similar cost estimation problems.

3.1 Industrial Context and ETO Cost Estimation Challenges

The industrial partner that contributed to this research works in the field of customised automation systems. The company creates solutions like robotic platforms, automated transport systems, and specialised machinery made for each client's needs. These projects are usually unique, with different technical requirements, designs, and sizes. This matches the ETO production model. Some examples, such as robotic arms and automated guided vehicles, are shown in Figure 3.1.

Figure 3.1 - Examples of automation solutions developed by the industrial partner. Source: Unsplash, accessed on 11 June 2025.

In the broader industrial automation sector, the demand for highly customised equipment has grown significantly, driven by the increasing need for flexible, efficient, and client-specific production solutions. Companies operating in this space are often required to design and manufacture one-off systems tailored to unique operational contexts, making cost estimation a particularly complex task. Unlike mass production industries, where standardised components and repeatable processes allow for predictable cost structures, customised automation relies on variable inputs, custom engineering, and evolving technical requirements. This is especially true for organisations that operate under an ETO model, where quoting must occur before full technical specifications are available. As a result, cost estimation must often be performed under time pressure and with incomplete information, further amplifying the risk of under- or overestimation. As Fortes et al. (2023) observe, ETO environments are inherently complex and variable, requiring intense coordination across departments and close involvement of the customer throughout the process, which significantly complicates planning and cost forecasting.

Even though ETO products are becoming more complex, many companies still use informal methods based on experience. These include expert judgement, simple rules, or internal templates that are not standardised or properly tested. While these methods are quick, they often give different results depending on the cost engineer or product. Also, it is hard to use more analytical or data-driven methods because there is not enough structured data and clear cost

drivers. Depending too much on informal knowledge makes the process less transparent and harder to repeat in other situations.

It is within this industrial and methodological context that the present research is situated. The work investigates whether ML models can serve as viable tools for supporting early-stage cost estimation in ETO environments, where uncertainty, variability, and time constraints frequently limit the effectiveness of traditional approaches. The data given by the industrial partner were chosen to be used early in the project. They help with the first steps of cost estimation and support the decision on whether the estimate should be sent to other departments for more analysis. Although no real-world project data were available, the set of input variables provided by the industrial partner, reflect the type of information typically used internally during quotation processes. These variables formed the foundation for the construction of synthetic datasets, enabling a controlled evaluation of different ML techniques.

3.2 Input Variables and Synthetic Dataset Design

The variables provided by the industrial partner were used to build the synthetic datasets used in this research:

- Project Familiarity (PF) – Captures the degree of familiarity with the client or industry, considering domain expertise, expected demands, and overall comfort;
- Technological Complexity (TC) – Evaluates whether the technologies used are familiar and well mastered by the company, or present significant technical challenges;
- Number of System Features (SF) – Represents an internal assessment of the number of modules and components likely to be included in the final solution;
- Project Duration (PD) – Indicates the overall lead time of the project. Extended durations tend to increase the complexity and uncertainty of cost estimates;
- Know-How Reuse (KH) – Assesses whether the company has existing knowledge applicable to the project and the extent to which it can be reused;
- Labour to Material Ratio (LM) – Represents the balance between labour and material costs, serving as an indicator of the project's structural impact on the organisation.

Each variable captures a distinct dimension of project effort, risk, or technical uncertainty and together they form the foundation for generating synthetic datasets used throughout the research.

The goal of the data generation process was to simulate a variety of realistic project profiles and explore how different data characteristics influence ML performance. Four datasets were created by combining two design dimensions: structure (rule-based vs. random) and variable type (categorical vs. mixed). This setup enables the comparison of ML behaviour under idealised, structured scenarios and more chaotic, unstructured ones. In particular, the random datasets are designed to simulate poor data conditions, helping assess how well ML models perform when data quality is low or inconsistent, a common situation in many ETO environments. Further details regarding this process, including variable distributions and modelling assumptions, are provided in Chapter 4.

3.3 Relevance of the Research

This research was designed to explore the feasibility of using ML to support early-stage cost estimation in ETO environments. From the outset, it was anticipated that ML models would outperform traditional methods in capturing the underlying cost dynamics across datasets, particularly those generated with rule-based logic. These datasets were built to reflect internally consistent relationships between project variables, making them more amenable to pattern recognition by learning algorithms. In contrast, lower performance was expected in the random

datasets, which lack structure. Based on findings in the literature, it was also anticipated that models, such as XGBoost and ANNs, would deliver higher predictive accuracy.

The approach explored in this research is closely aligned with the real-world constraints faced by organisations working within ETO environments, making its findings particularly relevant. The difficulty of producing reliable early-stage cost estimates in such environments is well documented, especially when detailed technical specifications are not yet available and historical data are either unstructured or entirely absent.

By working with a variable set that mirrors real-world decision-making practices and generating synthetic data to evaluate multiple ML techniques, this research provides insights that may be transferable to other ETO contexts facing similar challenges. This approach demonstrates how controlled experimentation can help clarify the potential value of ML in supporting quotation decisions under uncertainty, even in the absence of company-specific historical data.

4 Dataset Construction and Model Implementation

This chapter explains how the data and ML models were set up for this research. Section 4.1 describes how the input variables were defined and structured for dataset generation. Section 4.2 presents the strategy used to generate four synthetic datasets. This section also explains how a synthetic cost function was designed to generate realistic cost estimates for each project, based on project complexity. Finally, Section 4.3 describes how the four ML models were implemented and trained using the datasets, including baseline and tuned configurations.

4.1 Definition of Variables

The variables used in this research were previously introduced and conceptually described in Section 3.2. This section focuses on how those variables were technically structured and adapted for dataset generation, including scale definitions and modelling assumptions.

All variables were originally defined using a four-level scale, as provided in internal company documents. For three of them, Number of System Features, Project Duration, and Labour to Material Ratio, continuous value ranges were also specified or derived, allowing for greater modelling flexibility while remaining aligned with the original categorisation. The exact scale definitions and value ranges used for each variable are shown in Tables 4.1 and 4.2.

Table 4.1 - Scale definitions and value ranges for variables that included both categorical levels and associated continuous intervals

Variable Name	**Scale**	**From**	**To**
Number of System Features	1	1 module	2 modules
	2	3 modules	11 modules
	3	12 modules	19 modules
	4	20 modules	30 modules
Project Duration	1	4 weeks	12 weeks
	2	13 weeks	24 weeks
	3	25 weeks	48 weeks
	4	49 weeks	72 weeks
Labour to Material Ratio	1	1%	10%
	2	11%	25%
	3	26%	50%
	4	51%	100%

Table 4.2 - Discrete scale definitions for variables with no associated continuous intervals

Variable Name	**Scale**
Project Familiarity	1 – familiar and trusted context
	2 – familiar but challenging context
	3 – demanding or unfamiliar context
	4 – completely new
Know-How Reuse	1 – no novelty, full reuse
	2 – no novelty, high reuse
	3 – novelty, some reuse
	4 – all new, no reuse
Technological Complexity	1 – no complexity
	2 – moderate complexity
	3 – high complexity
	4 – very high complexity

These variables serve as the input dimensions for the datasets constructed in the following sections, where two different generation approaches are explored.

4.2 Data Generation Strategy

This section presents the strategy used to generate the synthetic datasets employed in this research. Section 4.2.1 provides an overview of the four datasets created, which combine the two generation methods with different variable formats (categorical and mixed). Section 4.2.2 describes the logic behind the heuristic generation method, detailing how dependencies were defined between variables to introduce structure and realism. Section 4.2.3 explains the random generation strategy used to create unstructured baselines for testing. Finally, Section 4.2.4 introduces a synthetic cost function developed to calculate estimated costs for each project.

4.2.1 Overview of the Datasets Created

Four synthetic datasets were developed to support the training and evaluation of the ML models. These datasets combine two key design dimensions: the data generation strategy (heuristic or random) and the type of variable encoding (fully categorical or mixed). Table 4.3, below, summarises the main characteristics of each dataset.

Table 4.3 - Overview of dataset configurations, highlighting the generation logic, data format, and role in the modelling process

Dataset	Generation Strategy	Variable Type	Purpose
DB1	Heuristic	Categorical (1-4 scale)	To simulate structured logic using predefined rules and discrete levels
DB2	Random	Categorical (1-4 scale)	To create an unstructured baseline for comparison, that reflects poor data conditions
DB3	Heuristic	Mixed	To combine rule-based logic with increased realism and variability
DB4	Random	Mixed	To create an unstructured baseline with greater granularity

Each dataset includes 1 000 project instances and contains the same six input variables defined earlier: Project Familiarity, Technological Complexity, Number of System Features, Project Duration, Know-How Reuse, and Labour to Material Ratio. A sample of four project records from each dataset is presented in Appendix A.

4.2.2 Generation Method 1: Heuristic

The first data generation approach was built using a set of predefined rules designed to reflect plausible dependencies between project variables. These rules were developed to introduce structure and consistency into the datasets by simulating how one variable might logically influence another. The aim was to replicate the kind of reasoning that typically guides early-stage project assessment, without relying on real data. The overall structure of these dependencies is summarised in Figure 4.1.

This heuristic was applied in the creation of datasets DB1 and DB3, as detailed in Table 4.3 in the previous section.

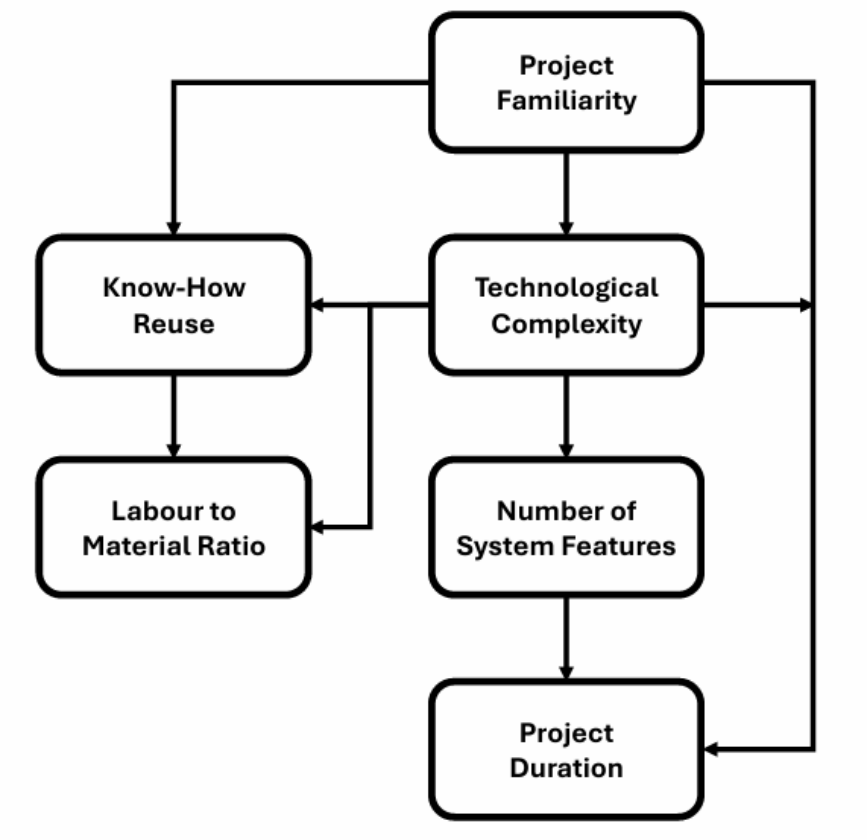

Figure 4.1. Dependency structure between variables used in heuristic-based data generation

Project Familiarity (PF)

The generation process begins with PF, which serves as the root of the dependency chain. This variable was chosen as the starting point because familiarity with the client and their industry can significantly shape the company's confidence, perceived risk, and estimation practices. If the organisation has worked with a client before, or operates frequently in their sector, it is likely to rely on well-established assumptions and reusable knowledge. In contrast, engaging with new clients, or in industries where the company has limited experience, typically introduces more uncertainty, making PF a logical anchor for downstream dependencies. In both DB1 and DB3, PF was generated as a random integer between 1 and 4.

Technological Complexity (TC)

Building on that, TC was defined as directly dependent on PF. The reasoning here is that projects for unfamiliar clients or industries are more likely to involve novel or untested technologies, increasing the perceived complexity. When a company is less familiar with the client, it may also be exposed to new technical standards, integration challenges, or constraints that push the technological boundaries of what it normally delivers. Conversely, when the client and their needs are well understood, the technologies involved tend to fall within the company's existing capabilities, reducing TC.

To reflect this logic, TC was sampled according to a probability distribution that varied depending on the value of PF. The structure ensures that high PF (a strong relationship with the client) leads to a greater likelihood of low TC, while low PF increases the probability of more complex scenarios. Table 4.4 shows how the value of TC was chosen depending on the level of PF.

This variable was generated the same way in both DB1 and DB3.

Table 4.4 - Probability distributions used to assign TC values based on the level of PF

Conditioning Variable	Value	Outcome Variable	TC=1	TC=2	TC=3	TC=4
Project Familiarity	1	Technological Complexity	60%	20%	10%	0%
	2		20%	40%	30%	10%
	3		10%	30%	40%	20%
	4		0%	10%	20%	60%

Number of System Features (SF)

Next, SF was modelled as dependent on TC. The idea is that more complex projects usually need more technical modules, subsystems, or custom options. When a project uses known or standard technology, it tends to need fewer features. But when the technology is more advanced or new, the project may require a wider range of features. In DB1, SF values were chosen using probabilities based on the level of TC. These distributions are shown in Table 4.5.

Table 4.5 - Probability distributions used to assign SF values based on the level of TC

Conditioning Variable	Value	Outcome Variable	SF=1	SF=2	SF=3	SF=4
Technological Complexity	1	Number of System Features	60%	20%	10%	0%
	2		20%	40%	30%	10%
	3		10%	30%	40%	20%
	4		0%	10%	20%	60%

In DB3, a different method was used. Instead of selecting a value from 1 to 4, real values were randomly picked from the value ranges defined in Table 4.1. Each range corresponded to a level of TC, allowing for a more detailed variation while keeping the same underlying logic. The value of SF was selected as follows:

- If TC = 1, the value for SF was randomly picked between 1 and 2 modules;
- If TC = 2, the value was picked between 3 and 11 modules;
- If TC = 3, the value was picked between 12 and 19 modules;
- If TC = 4, the value was picked between 20 and 30 modules.

Project Duration (PD)

PD depends on three other variables: PF, TC, and SF. The idea is that working with unfamiliar clients, using complex technologies, or having many features in a system usually leads to longer projects. These situations increase uncertainty and require more time for planning, designing, and testing. By using all three of these influences together, PD works as an overall indicator of project size and how much effort is needed.

To apply this logic, in DB1, a score was created by taking the values of PF, TC, and SF, subtracting one from each, and adding them together. This method ensures that the least demanding project scenario (PF = 1, TC = 1, SF = 1) is represented by the lowest score (0), while the most complex scenario (PF = 4, TC = 4, SF = 4) corresponds to the highest score (9). The formula used was:

$$score\_pd = (pf - 1) + (tc - 1) + (sf - 1) \quad (4.1)$$

Where:

score_pd is the normalised score for project duration;

pf is the score associated to project familiarity;

tc is the score associated to technological complexity;

sf is the score associated to number of system features.

The score was then mapped to four PD levels:

- If *score_pd* was 2 or less, then PD was set to 1;
- If *score_pd* was between 3 and 4, then PD was set to 2;
- If *score_pd* was between 5 and 6, then PD was set to 3;
- If *score_pd* was greater than 6, then PD was set to 4.

In DB3, the same scoring method was used, but the output values were continuous rather than categorical. The score was used to randomly assign a real value from one of the PD intervals listed in Table 4.1. The mapping was:

- If *score_pd* was 2 or less, PD was randomly selected between 4 and 12 weeks;
- If *score_pd* was between 3 and 4, PD was selected between 13 and 24 weeks;
- If *score_pd* was between 5 and 6, PD was selected between 25 and 48 weeks;
- If *score_pd* was greater than 6, PD was selected between 49 and 72 weeks.

This approach introduced greater granularity in DB3 while preserving the underlying logic used in DB1.

Know-How Reuse (KH)

KH depends on both PF and TC. The underlying assumption is that when the client is familiar and the technological demands are low, the company is more likely to reuse existing tools, engineering practices, and documentation. In contrast, if the client is new or the technology is novel, reuse becomes more difficult, and the company may have to develop solutions from scratch, thus reducing the ability to leverage past knowledge. KH was therefore placed at the intersection of these two influences, reflecting their combined effect on the potential for internal knowledge reuse.

To apply this logic, a score was created by taking the values of PF and TC, subtracting one from each, and adding them together, similar to what was done previously for PD. This ensured that the most familiar and least complex projects (PF = 1, TC = 1) were assigned the lowest score (0), while the highest score (6) represented a new client and highly complex technology (PF = 4, TC = 4). The formula used was:

$$score\_kh = (pf - 1) + (tc - 1) \tag{4.2}$$

Where:

score_kh is the normalised score for know-how reuse;

pf is the score associated to project familiarity;

tc is the score associated to technological complexity.

The score was then mapped to four KH levels:

- If *score_kh* was 1 or less, KH was set to 1;
- If *score_kh* was between 2 and 3, KH was set to 2;
- If *score_kh* was between 4 and 5, KH was set to 3;
- If *score_kh* was 6, KH was set to 4.

This logic was applied in both DB1 and DB3, where KH was always treated as a categorical variable.

Labour to Material Ratio (LM)

Finally, LM was defined as a function of KH and TC. This relationship reflects the idea that when little or no existing knowledge can be reused (KH = 4) and the technological demands are high (TC = 4), projects tend to be significantly more labour-intensive. In these cases, a greater share of the total cost is associated with internal engineering work rather than material procurement.

To apply this logic, in DB1, a score was calculated based on the values of KH and TC. Each input was adjusted by subtracting one, so that the scale started at zero. The formula used was:

$$score\_lm = (kh - 1) + (tc - 1) \quad (4.3)$$

Where:

score_lm is the normalised score for labour to material ratio;

kh is the score associated to know-how reuse;

tc is the score associated to technological complexity.

This score could range from 0 (full reuse and low complexity) to 6 (new solutions and very high complexity). The final LM value was assigned as follows:

- If *score_lm* was 1 or less, LM was set to 1;
- If *score_lm* was between 2 and 3, LM was set to 2;
- If *score_lm* was between 4 and 5, LM was set to 3;
- If *score_lm* was 6, LM was set to 4.

In DB3, the same logic was used, but the output values were generated as percentages to reflect continuous variation. The score was mapped to the continuous ranges listed in Table 4.1:

- If *score_lm* was 1 or less, LM was randomly selected between 1% and 10%;
- If *score_lm* was between 2 and 3, LM was selected between 11% and 25%;
- If *score_lm* was between 4 and 5, LM was selected between 26% and 50%;
- If *score_lm* was 6, LM was selected between 51% and 100%.

This logic ensures that combinations with high complexity and low reuse are mapped to higher LM values, reflecting a shift toward labour-driven cost structures.

In short, this heuristic model was designed to reflect realistic relationships between project variables and create internally consistent datasets for training and testing.

4.2.3 Generation Method 2: Random

To complement the heuristic datasets, a second set of datasets was created using random generation, DB2 and DB4. The goal was to build a baseline where there are no predefined relationships between variables and to simulate weak or unstructured data conditions. This allows for a clearer comparison with the structured datasets.

In DB2, each variable was generated independently and categorically. All variables were assigned random integer values between 1 and 4, with equal probability for each level. This dataset reflects a completely unstructured scenario where no dependencies exist between variables.

In DB4, the same logic was applied, but with continuous outputs for variables where value ranges had been defined. For example, PD was randomly selected between 4 and 72 weeks, and LM between 1% and 100%. SF values were also drawn from the full range of 1 to 30 modules. As in DB2, variables were generated independently and without any underlying structure.

The scales and value ranges used for the randomly generated data are the same as those defined in Section 4.1, ensuring consistency across all datasets and allowing for fair comparisons between generation strategies.

4.2.4 Cost Function Design

In the absence of real cost data, a synthetic cost estimation method was developed to generate realistic and scalable values for each project instance. The goal was to simulate a plausible way of estimating cost, based on the idea that higher project complexity leads to higher effort and, therefore, higher cost.

Step 1: Calculating the Project Score

For each dataset entry, a total project score was calculated by summing the values of all variables, each multiplied by a fixed weight. This score served as an abstract representation of the overall complexity or effort involved in delivering the project. The formula used was:

$$score_{all} = w * \sum_{i=1}^{n} v_i \tag{4.4}$$

Where:

$score_{all}$ is the total project score;

v_i is the value of variable i;

w is the associated weight (set to 1.3 for all variables);

n is the number of variables considered (6 in this research).

The weight of 1.3 was selected after testing different values and observing the resulting cost distribution. This value produced a more realistic spread of cost outcomes, helping to differentiate projects according to their complexity levels.

In the case of DB3 and DB4, where some variables were generated as continuous values, their values were converted to a 1–4 scale before being used in the score calculation. This was done by adjusting each continuous value proportionally within its original range, as defined in Table 4.1, to fit the same numerical interval used for the categorical variables. This step ensured that all variables had a comparable influence on the final score and maintained consistency across datasets with different formats.

Step 2: Mapping Score to Cost

To translate the abstract score into a monetary estimate, a non-linear transformation was applied. This transformation increases more rapidly for higher scores, reflecting the intuition that cost tends to rise disproportionately as projects become more demanding. The final cost was computed using the following formula:

$$estimated\_cost = \left(FC + \left(\frac{score_{all}}{score_{max}}\right)^{\alpha} * (MC - FC)\right) * noise \tag{4.5}$$

Where:

estimated_cost is the final predicted value for the project's total cost, calculated based on all input variables, scaled by their weighted contribution and adjusted for uncertainty;

$score_{all}$ is the total weighted score for the project, calculated through equation (4.4);

$score_{max}$ is the theoretical maximum score, used to normalise the result and control the effect of score scaling in the cost formula. It represents the scenario in which all variables reach their highest possible value, 24 in this case;

FC is the fixed cost, set at 10 000€, to represent a minimum project cost;

MC is the maximum possible cost. In this research, it was set at 1 000 000€ to simulate the upper bound of a highly complex and resource-intensive project;

noise introduces variability to mimic the uncertainty typically present in early-stage cost estimation. Values were randomly sampled between 0.95 and 1.05;

α is a sensitivity factor used to shape the cost scaling curve, amplifying or reducing the impact of project score on the estimated cost. For this research, the value of α was set to 1.3.

This synthetic cost function ensures a logical relationship between project complexity and estimated cost while introducing controlled variability. Although artificial, the function provides a realistic approximation of early-stage cost uncertainty, making it a robust foundation for testing ML prediction models in the absence of real industrial data.

4.3 Machine Learning Implementation

This section explains how ML models were used to estimate costs in ETO projects. Four algorithms were tested: linear regression, random forest, XGBoost, and artificial neural networks. These models were chosen to cover different types of learning methods, from simple linear models to more complex ones.

Section 4.3.1 introduces the four algorithms and explains why they were selected. Sections 4.3.2 to 4.3.5 describe how each model was trained and tested, including different tuning strategies.

4.3.1 Algorithms Selected

The ML algorithms used in this research were selected based on their reported success in the literature, particularly in the context of project cost estimation. In addition to commonly used models, linear regression was included to serve as a simple baseline. The selected algorithms represent a mix of statistical, ensemble-based, and neural approaches, offering a broad perspective on modelling performance. The final selection includes:

- Linear Regression – Included as a baseline for comparison. Frequently used in related studies to benchmark more advanced models;
- Random Forest – A robust ensemble method known for its good performance in regression tasks and its ability to model non-linear relationships without extensive parameter tuning;
- XGBoost – Chosen for its high accuracy and flexibility in structured data problems. Frequently outperforms other models in regression benchmarks;
- ANN – selected to assess whether increased model complexity translates into improved performance in this context. While previous studies have reported mixed results with ANNs due to small datasets, this project used a larger dataset, enabling a more robust test.

These choices reflect what was found in the literature review, where these algorithms showed good performance in similar problems.

4.3.2 Linear Regression

Linear regression was used as a baseline model to provide a simple performance benchmark for more complex algorithms. Its inclusion aimed to assess how much value advanced models could add beyond a standard linear approach.

The implementation followed a consistent process across all four datasets:

- The target variable was *estimated_cost*;
- The model used six predictor variables: PF, TC, SF, PD, LM, and KH;

- Each dataset was split into training (80%) and testing (20%) subsets using *train_test_split* from the scikit-learn library with a fixed random seed for reproducibility;
- A *LinearRegression* model from scikit-learn was trained on the training data;
- No hyperparameter tuning was performed, consistent with the model's role as a simple benchmark;
- Model performance was evaluated on the testing set using three metrics: R^2, MAE, and RMSE;

This procedure enabled a consistent and interpretable baseline, allowing for a clear comparison with more complex models.

4.3.3 Random Forest

Random forest was selected as a core algorithm in this research due to its flexibility, robustness against overfitting, and proven performance in various cost estimation problems. Its ability to model complex, non-linear relationships made it particularly well-suited for the type of data generated in this project. Three different implementations were tested to evaluate its behaviour under different tuning strategies: default configuration (no tuning), randomised search, and grid search.

Default configuration (no tuning): The first implementation served as a baseline to evaluate the model's performance with default parameters:

- The target variable was *estimated_cost*;
- The features used were: PF, TC, SF, PD, LM, and KH;
- Datasets were split into training (80%) and testing (20%) subsets using scikit-learn's *train_test_split*, with a fixed random seed for reproducibility;
- A *RandomForestRegressor* was trained with n_estimators=100, and all other parameters left as default;
- Model performance was evaluated using R^2, MAE, and RMSE.

This setup established a reference point for later comparison with tuned models.

With Randomised Search: To improve model performance, randomised search was used to explore a wider hyperparameter space efficiently:

- A *RandomisedSearchCV* object was configured with 50 iterations and 3-fold cross-validation (cv=3), optimising for the r^2 score;
- The best estimator was extracted and evaluated on the testing set using R^2, MAE, and RMSE.

This method enabled the model to test a wide range of configurations with manageable runtime.

With Grid Search: grid search was also applied to allow for a more exhaustive search over a smaller, focused hyperparameter space:

- The same train-test split and variables were used;
- A *GridSearchCV* object was run using 3-fold cross-validation and r^2 as the scoring function.
- The selected model was then evaluated using R^2, MAE, and RMSE on the testing set.

The grid was deliberately restricted to reduce computational time, since each configuration had to be evaluated multiple times and across four datasets.

Table 4.6 summarises the hyperparameter ranges used in both tuning strategies, which were defined pragmatically based on early test results and available computational resources.

Table 4.6 - Hyperparameter values tested in Random Forest tuning (Randomised and Grid Search)

Parameter	Randomised Search Values	Grid Search Values
n_estimators	[50, 100, 150, 200, 250, 300]	[100, 200]
max_depth	[none, 5, 10, 15, 20, 30]	[10, 20, none]
min_samples_split	[2, 3, 5, 10]	[2, 5]
min_samples_leaf	[1, 2, 4, 6]	[1, 2]
max_features	[“auto”, “sqrt”, “log2”]	[“auto”, “sqrt”]
bootstrap	[true, false]	[true, false]

All three implementations were applied to each of the four datasets (DB1 through DB4), allowing for a comparative analysis of model behaviour across different tuning strategies and data generation methods.

4.3.4 XGBoost

XGBoost was selected for its strong track record in regression tasks and its ability to capture non-linear relationships through gradient boosting. Widely used in both academic and industrial settings, it offers a flexible yet efficient modelling approach. As with random forest, three variants were tested in this research: default configuration (no tuning), randomised search, and grid search.

Default configuration (no tuning): The first implementation used XGBoost with a default configuration to establish a baseline performance:

- The target variable was *estimated_cost*;
- The features included: PF, TC, SF, PD, LM, and KH;
- Each dataset was split into training (80%) and testing (20%) using *train_test_split* with a fixed random seed;
- The model used was *XGBRegressor* with n_estimators=100, learning_rate=0.1, and all other parameters set to default;
- Performance was evaluated using R^2, MAE, and RMSE on the testing set.

This basic configuration provided an initial view of the algorithm’s capability without optimisation.

With Randomised Search: To explore a wider hyperparameter space more efficiently, randomised search was applied:

- A *RandomisedSearchCV* object was configured with 50 iterations and 3-fold cross-validation (cv=3), optimising for the r^2 score;
- The selected model was then evaluated using R^2, MAE, and RMSE on the testing set.

This approach allowed the model to test many configurations in a shorter time than a full grid search, providing a good balance between performance and efficiency.

With Grid Search: A more exhaustive but targeted tuning was also conducted using grid search:

- The same training and test splits were applied;
- A *GridSearchCV* object was run with 3-fold cross-validation and R^2 as the evaluation metric;
- The search space was narrower, focusing on values that had shown promise in the randomised search.
- The best model was selected and evaluated using R^2, MAE, and RMSE on the test set.

Although more computationally demanding, this method aimed to fine-tune the model further. The full list of hyperparameters tested for both tuning strategies is presented in Table 4.7 below.

Table 4.7 - Hyperparameter values tested in XGBoost tuning (Randomised and Grid Search)

Parameter	Randomised Search Values	Grid Search Values
n_estimators	[100, 200, 300, 400]	[100, 300, 400]
learning_rate	[0.001, 0.005, 0.01, 0.05, 0.1]	[0.01, 0.05, 0.1]
max_depth	[3, 4, 5, 6, 8, 10]	[4, 6, 8]
subsample	[0.5, 0.6, 0.8, 1.0]	[0.8, 1.0]
colsample_bytree	[0.5, 0.6, 0.8, 1.0]	[0.8, 1.0]
gamma	[0, 0.1, 0.3, 0.5]	[0, 0.3]
reg_alpha	[0, 0.01, 0.1, 1]	[0, 0.1]
reg_lambda	[0.1, 1, 5, 10]	[1, 10]

As with random forest, all three XGBoost configurations were applied to each of the four datasets (DB1 to DB4), supporting a robust comparison across different model setups and data generation strategies.

4.3.5 Artificial Neural Networks

ANNs were included in this research to explore the potential of more complex, non-linear models. While ANNs have been used in previous cost estimation research, several studies noted challenges such as limited dataset size or training instability. This project benefited from access to a sufficiently large dataset, making it possible to apply ANN models more confidently and test their performance under controlled conditions.

The implementation required additional considerations, such as feature scaling and more extensive hyperparameter tuning. Three approaches were tested: default configuration (no tuning), randomised search, and grid search. Two solvers were explored: *adam* and *lbfgs*.

Default configuration (no tuning): This setup established a baseline ANN model using a fixed architecture and default parameters:

- The target variable was *estimated_cost*;
- The input features included: PF, TC, SF, PD, LM, and KH;
- Data was split into training (80%) and testing (20%) using *train_test_split* with a fixed random seed;
- Both input and output variables were standardised using *StandardScaler*;
- A *MLPRegressor* was trained with the following configuration:
  - hidden_layer_sizes=(50, 20)
  - activation='relu'
  - solver='adam'
  - max_iter=1000
  - early_stopping=True
- After training, predictions were inverse-transformed to original scale and evaluated using R^2, MAE, and RMSE.

With Randomised Search: To improve model performance, randomised search was used to explore a broader space of architectures and optimisation settings:

- Two sets of hyperparameters were defined: one for *adam* and one for *lbfgs* (excluding learning rate options);

- The search was configured with 50 iterations using *RandomizedSearchCV*, with 3-fold cross-validation and r^2 as the scoring function;
- For each solver:
    - The dataset was scaled in the same manner as above;
    - A base *MLPRegressor* was used as the estimator with max_iter=1000 and early_stopping=True;
    - The best model from the search was evaluated on the testing set using R^2, MAE, and RMSE (after inverse-transforming the predictions to the original scale).

With Grid Search: grid search was also applied to enable a more exhaustive and structured evaluation across the same parameter space used in the randomised search:

- The same data split and scaling procedures were followed;
- Two separate grids were defined:
    - For *adam*, including combinations of learning rates and activation functions;
    - For *lbfgs*, focusing on weight regularisation and hidden layer sizes (excluding learning rate);
- A *GridSearchCV* object was used with 3-fold cross-validation and r^2 as the scoring metric;
- Each dataset was tested with both solvers, and the best model was selected for evaluation using R^2, MAE, and RMSE.

Both *adam* and *lbfgs* solvers were tested independently using the same hyperparameter ranges, as shown in Table 4.8. These ranges were applied identically for both randomised and grid search, with only minor adjustments where required.

Table 4.8 - Hyperparameter search space used in ANN tuning

Parameter	**Search Values**
hidden_layer_sizes	[(20,), (50,), (100,), (100,20), (50,20), (100,50), (128,64,32), (64, 32, 16)]
Activation	[“relu”, “tanh”, “logistic”]
Alpha	[1e-5, 0.0001, 0.001, 0.01, 0.1]
learning_rate	[“constant”, “adaptive”] (only for *adam*)
learning_rate_init	[0.001, 0.005, 0.01, 0.05] (only for *adam*)

This approach enabled the ANN to adapt to various dataset characteristics and configurations. All three strategies were applied to each of the four datasets, providing a broad view of ANN performance under different optimisation regimes and solver types.

In the following chapter, the results of this research are presented.

5 Results

This chapter presents the results obtained from applying the ML models described in Chapter 4 to the generated datasets.

The results are organised by dataset to provide a clear view of how different models and tuning strategies performed under varying data conditions. Only the best configuration for each model is shown. No interpretation or ranking of models is provided here. A comparative discussion follows in Chapter 6.

5.1 Performance on DB1

This section presents the performance of all ML models on DB1, the heuristic dataset with categorical inputs. Table 5.1 shows the best results for each model and tuning strategy. Most models achieved high accuracy, with only small differences across configurations.

Table 5.1. Performance of machine learning models on DB1

Algorithm	Tuning Strategy	R^2	MAE	RMSE
Linear Regression	None	0.9941	16364	19880
Random Forest	None	0.9955	13545	17333
Random Forest	Randomised Search	0.9956	13422	17102
Random Forest	Grid Search	0.9957	13399	17041
XGBoost	None	0.9956	13435	17162
XGBoost	Randomised Search	0.9958	13295	16803
XGBoost	Grid Search	0.9957	13382	16960
ANN	None	0.9951	13862	18049
ANN (lbfgs)	Randomised Search	0.9959	13252	16511
ANN (lbfgs)	Grid Search	0.9960	13252	16511
ANN (adam)	Randomised Search	0.9958	13444	16773
ANN (adam)	Grid Search	0.9957	13726	17032

5.2 Performance on DB2

This section presents the performance of all ML models on DB2, the randomly generated dataset with categorical inputs. Table 5.2 shows the best results for each model and tuning strategy. While overall performance remained strong, results were slightly lower than on DB1, reflecting the unstructured nature of this dataset.

Table 5.2. Performance of machine learning models on DB2

Algorithm	**Tuning Strategy**	**R^2**	**MAE**	**RMSE**
Linear Regression	None	0.9821	14848	17984
Random Forest	None	0.9488	24269	30416
Random Forest	Randomised Search	0.9449	25091	31550
Random Forest	Grid Search	0.9449	25048	31529
XGBoost	None	0.9680	18755	24046
XGBoost	Randomised Search	0.9806	15335	18713
XGBoost	Grid Search	0.9774	16195	20208
ANN	None	0.9788	16357	19581
ANN (lbfgs)	Randomised Search	0.9842	14216	16904
ANN (lbfgs)	Grid Search	0.9835	14431	17278
ANN (adam)	Randomised Search	0.9829	14691	17550
ANN (adam)	Grid Search	0.9837	14340	17146

5.3 Performance on DB3

This section presents the performance of all ML models on DB3, the heuristic dataset with mixed input types. Table 5.3 shows the best results for each model and tuning strategy. Performance remained strong across configurations.

Table 5.3. Performance of machine learning models on DB3

Algorithm	**Tuning Strategy**	**R^2**	**MAE**	**RMSE**
Linear Regression	None	0.9930	16539	21300
Random Forest	None	0.9922	16944	22537
Random Forest	Randomised Search	0.9931	16110	21159
Random Forest	Grid Search	0.9933	15807	20908
XGBoost	None	0.9919	17000	22926
XGBoost	Randomised Search	0.9936	15676	20313
XGBoost	Grid Search	0.9931	16258	21208
ANN	None	0.9944	15205	19030
ANN (lbfgs)	Randomised Search	0.9948	14091	18281
ANN (lbfgs)	Grid Search	0.9950	13874	18060
ANN (adam)	Randomised Search	0.9946	14294	18743
ANN (adam)	Grid Search	0.9948	14001	18301

5.4 Performance on DB4

This section presents the performance of all ML models on DB4, the randomly generated dataset with mixed inputs. Table 5.4 shows the best results for each model and tuning strategy. As expected, the lack of structure led to greater variability in performance across algorithms.

Table 5.4. Performance of machine learning models on DB4

Algorithm	**Tuning Strategy**	**R^2**	**MAE**	**RMSE**
Linear Regression	None	0.9807	14178	16687
Random Forest	None	0.9071	28585	36558
Random Forest	Randomised Search	0.9172	26798	34525
Random Forest	Grid Search	0.9157	27007	34826
XGBoost	None	0.9379	23829	29886
XGBoost	Randomised Search	0.9750	15467	18977
XGBoost	Grid Search	0.9744	15195	19210
ANN	None	0.9750	15459	18960
ANN (lbfgs)	Randomised Search	0.9821	13289	16030
ANN (lbfgs)	Grid Search	0.9822	13303	16009
ANN (adam)	Randomised Search	0.9807	13929	16646
ANN (adam)	Grid Search	0.9807	13955	16645

6 Discussion

This chapter discusses how each ML model performed across the four datasets. The analysis focuses on baseline accuracy, the effect of tuning, and how data structure influenced results. The final sections compare algorithms, explore practical implications, and reflect on the research's limitations.

6.1 Discussion of DB1 results

6.1.1 Baseline Comparison: Untuned Models

In DB1, all untuned models delivered excellent results, with R^2 values above 0.99 across the board. This strong performance reflects the structured and rule-based nature of the dataset, which offered clear patterns for the models to learn from. XGBoost and random forest showed the highest predictive accuracy, suggesting that ensemble methods are well suited to environments where the data follows deterministic logic.

ANNs also performed well, with results comparable to the tree-based models despite their different architecture. Their ability to model non-linear relationships appears to have helped capture subtle patterns in the data. Linear regression, while slightly behind the others, still provided a robust baseline with an R^2 of 0.9941 and an RMSE under 20 000€. Its simplicity made it less sensitive to more complex interactions, but its performance remained strong due to the linear tendencies present in the dataset.

Overall, these results indicate that in highly structured scenarios like DB1, where inputs are clearly related to outputs, all models can achieve high accuracy without tuning. However, more flexible algorithms such as XGBoost and ANN still hold a slight edge, likely due to their ability to fine-tune decision boundaries beyond what linear models can offer.

6.1.2 Impact of Hyperparameter Tuning

Tuning had the biggest impact on ANN and XGBoost. In both cases, it helped reduce prediction errors compared to their untuned versions. This shows that even though these models already performed well, careful tuning made them more accurate. For ANN, the solver also made a difference, *lbfgs* gave better results than *adam*. The improvements were not huge, but in settings where accuracy matters, they could be useful.

Random forest, on the other hand, only improved slightly with tuning. This suggests that in structured datasets like DB1, it works well without needing many adjustments. Its default settings were already close to the best it could do.

In short, tuning is most helpful for models that can still adapt and improve. It is not always necessary, but when used on the right models, it can bring clear benefits without changing the overall structure of the dataset.

6.2 Discussion of DB2 results

6.2.1 Baseline Comparison: Untuned Models

In DB2, performance dropped noticeably across all models compared to the structured scenario in DB1. The most stable results came from linear regression, which achieved the highest R^2 and the lowest MAE and RMSE among untuned models. This suggests that in unstructured and random data environments, simpler models may generalise better by avoiding overfitting to noise.

ANNs also adapted relatively well. While they did not outperform linear regression, they delivered the second-best results across all three metrics. This shows that even in noisy data, the ANN architecture can still find weak patterns and generalise effectively without overfitting.

XGBoost showed a more significant performance drop relative to DB1. The model struggled to maintain the same level of accuracy when confronted with random, unstructured input. Its R^2 was notably lower, and both MAE and RMSE increased, indicating difficulties in separating signal from noise without underlying data structure.

Random forest was the weakest performer in this setting. Its reliance on split-based logic appears to have been particularly vulnerable to randomness in the input features, leading to poor generalisation and the highest error values of all models.

Overall, these results confirm that unstructured data presents serious challenges for model performance. While complex models may excel in structured environments, their tendency to overfit becomes a liability when clear patterns are absent. In contrast, simpler or more adaptable models, like linear regression and ANN, tend to offer more reliable baselines in noisy or weakly structured scenarios.

6.2.2 Impact of Hyperparameter Tuning

In the randomised context of DB2, hyperparameter tuning had mixed effects across models, with improvements observed only in those algorithms that were inherently flexible and able to cope with noisy or weakly structured inputs.

ANNs responded particularly well to tuning. The *lbfgs* solver achieved the best overall performance in this dataset, reducing both MAE and RMSE to the lowest values across all configurations. While the *adam* solver also benefited from tuning, the gains were slightly smaller. These improvements suggest that ANNs, when properly configured, can adapt their internal representations to capture even subtle or partial patterns present in the data, something that untuned versions only achieved to a limited extent. The fact that both solvers improved consistently indicates that ANNs are not just flexible but also tunable, a key trait when dealing with unpredictable inputs.

XGBoost also showed meaningful improvements from tuning. The best results were obtained through randomised search, which led to noticeable gains across all metrics. This suggests that although the untuned model struggled with the unstructured nature of DB2, fine-tuning allowed it to better regulate its complexity and focus on more generalisable patterns. Still, its final performance remained slightly behind the best ANN configurations, highlighting that even well-regularised tree-based models are not immune to the effects of noise.

In contrast, random forest did not benefit from tuning. In fact, performance deteriorated slightly after both tuning strategies. This reinforces the notion that random forest's effectiveness is tightly coupled to the presence of structure in the data, without clear splits or feature hierarchies to exploit, the model appears to learn from randomness, and tuning cannot fix this fundamental mismatch. These results also confirm that tuning cannot compensate for a poor fit between model assumptions and data characteristics.

Taken together, these findings illustrate a critical distinction: tuning is only helpful when the model has the underlying capacity to adapt. In DB2, models like ANN and XGBoost had that capacity, and the optimisation process helped refine their learning strategies. On the other hand, models that lack flexibility did not benefit from additional tuning effort.

These observations also align with the broader conclusion that in highly variable, noisy environments, model selection and adaptability matter more than tuning alone. The ability to respond to weak signals, rather than amplify noise, determines whether a model can improve with tuning or not.

6.3 Discussion of DB3 results

6.3.1 Baseline Comparison: Untuned Models

In DB3, all untuned models performed well, with R^2 values above 0.99. This dataset introduced more granularity than DB1 by combining categorical and continuous inputs with structured relationships. The added detail created more complex patterns in the data, which benefitted models capable of learning subtle interactions.

Among the untuned models, ANNs achieved the best performance. Their architecture is particularly well suited to this type of dataset, as they can capture both non-linear relationships and fine-grained patterns. The increased granularity in DB3 appears to have worked in their favour, providing more nuanced signals for the network to learn from.

XGBoost and random forest also produced strong results, although their errors were slightly higher than those of the ANN. These models handled the data structure well but may not have captured the same level of subtlety without tuning.

Linear regression remained consistent, but its relative performance declined slightly. This is likely because the more complex relationships in DB3 went beyond the linear trends it can model.

Overall, DB3 highlights how greater data richness can shift the advantage toward more flexible and expressive models. While all algorithms performed well, ANNs stood out as the most capable of taking full advantage of the dataset's mixed and detailed structure.

6.3.2 Impact of Hyperparameter Tuning

Tuning improved results across all models in DB3, but the extent of the improvement varied by algorithm. The most noticeable gains came from ANN and XGBoost, which already performed well without tuning and were able to improve further with careful parameter adjustment.

ANN achieved the best performance after tuning, particularly with the *lbfgs* solver. This configuration consistently delivered the lowest error values, confirming that tuning helped the model better align with the structure and complexity of the dataset. The *adam* solver also showed gains, although slightly smaller. These results confirm that ANN is not only flexible but also highly tunable, making it an excellent fit for datasets that include a mix of categorical and continuous variables.

XGBoost also responded well to tuning. The best result was obtained using randomised search, showing clear improvement in all metrics compared to its untuned version. These results suggest that even when the data is already well structured, tuning helps this model fine-tune its decisions and make better use of the available information.

Random forest improved slightly with tuning. Both tuning methods delivered modest gains, but the improvements were smaller than those seen with ANN and XGBoost. This may be because random forest's default settings already align well with the structure of the dataset, or because the model reaches its performance ceiling more quickly.

Together, these results reinforce that tuning is most beneficial for models that already show flexibility and adaptability. While the gains were not dramatic in absolute terms, they show that when working with structured, mixed-type data, performance can still be improved with thoughtful configuration. ANN, in particular, stood out again as the most responsive model, confirming its strong performance across different scenarios.

6.4 Discussion of DB4 results

6.4.1 Baseline Comparison: Untuned Models

In DB4, which combined continuous inputs with random generation, model performance varied more than in previous datasets. Linear regression gave the most stable results among the untuned models. It achieved the highest R^2 and the lowest MAE and RMSE. Its simplicity seemed to work as an advantage in this setting, allowing it to focus on general trends without getting distracted by noise.

ANN also performed well, with results close to linear regression. Even in a noisy and unstructured dataset, its flexible architecture allowed it to detect weak patterns. This suggests that ANN can remain competitive when structure is limited, even if its full potential is not reached.

Tree-based models struggled more. XGBoost showed a clear drop in performance compared to structured datasets. Despite having regularisation, it found it hard to separate signal from noise. Its results were noticeably weaker than those of ANN and linear regression.

Random forest had the lowest scores overall. Its logic based on data splits did not work well in this context. Without patterns to follow, it likely learned from noise, which led to poor generalisation and the highest error values.

These results confirm the challenges of learning from random data. Simpler models like linear regression and flexible models like ANN handled the situation better. More complex methods, such as random forest and XGBoost, were more sensitive to the lack of structure and performed worse without careful tuning.

6.4.2 Impact of Hyperparameter Tuning

As with DB2, the effect of hyperparameter tuning in DB4 depended on each model's flexibility and ability to deal with noisy inputs.

ANN showed the most benefit. With the *lbfgs* solver and grid search, it reached the best performance in this dataset. The *adam* solver also improved, though slightly less. These results show that, even in noisy environments, ANN can adapt if tuned well. Its internal structure allows it to explore weak or hidden patterns that other models may miss.

XGBoost also improved clearly with tuning. Its best configuration, found using randomised search, reduced the prediction errors and increased the R^2 to a level close to the best ANN results. This shows that XGBoost can adapt when properly configured, even in random data, by focusing more on general trends and less on overfitting.

Random forest, however, remained limited. Tuning led to only small improvements, and the model still performed worse than all others. This confirms that if a model is not suited to the data, adjusting the parameters does not solve the problem.

In summary, tuning helped ANN and XGBoost make better use of the weak patterns in DB4. Random forest did not show the same potential. These findings confirm that tuning is most useful when the model already has some capacity to learn from difficult data.

6.5 Performance Comparison by Algorithm

This section compares the performance of the four ML algorithms tested across all datasets. The goal is to briefly evaluate how each model adapted to different data structures, how tuning affected results, and whether the computational effort associated with model optimisation was

justified. By shifting focus from datasets to algorithms, this analysis aims to provide clearer guidance on the practical utility of each approach for ETO project cost prediction.

6.5.1 Linear Regression: Consistent, Interpretable, and Surprisingly Strong

Linear regression consistently delivered strong results, even in unstructured or noisy settings. It achieved the best performance in both DB2 and DB4, datasets with randomised and weakly structured inputs. This indicates that in highly variable environments, the simplicity and robustness of linear models are advantageous. While it performed slightly below more complex models in structured datasets like DB1 and DB3, its results remained competitive.

Importantly, linear regression was not subject to hyperparameter tuning, making it the most efficient model in terms of deployment. Its ease of interpretation, combined with low computational overhead and stable performance, suggests it is a highly suitable baseline for cost estimation in ETO settings, especially where model transparency is important or data is limited in quality.

6.5.2 Random Forest: Solid on Structure, Weak on Noise

Random forest performed well on the highly structured datasets where its ensemble nature helped capture rule-based relationships. However, its performance declined sharply in the unstructured scenarios. In DB4, for example, it recorded the lowest R^2 (0.9071) and highest RMSE (36 558 €) of all models, even after tuning.

Tuning led to only marginal improvements across all datasets. Gains in R^2 were generally below 0.01, and error reductions were small. This suggests that random forest's default settings are already quite robust, and its limitations in noisy contexts are more structural than parametric. In practice, this model may not be ideal for ETO contexts where data is unstructured or includes weak patterns. However, in environments where categorical rules or historical consistency exist, it remains a valid and interpretable choice.

6.5.3 XGBoost: High Potential, High Sensitivity

XGBoost delivered excellent results in structured scenarios like DB1, where it achieved top performance with default parameters ($R^2 = 0.9956$). Its strength lies in its ability to model complex relationships with built-in regularisation, which helps prevent overfitting in well-structured datasets.

However, its performance dropped in DB2 and DB4, where randomness was dominant. The untuned model struggled more than expected, with notable increases in MAE and RMSE. Tuning had a clear positive impact in those cases. For example, in DB4, randomised search improved the R^2 from 0.9379 to 0.9750 and reduced the RMSE by over 10 000 €, confirming that XGBoost can adapt if appropriately configured.

These findings highlight XGBoost's high sensitivity to data structure. It performs exceptionally when the data is structured but becomes vulnerable to noise unless carefully tuned. While tuning is computationally expensive, the gains in unstructured settings make it a worthwhile effort when model flexibility and accuracy are required.

6.5.4 Artificial Neural Networks: Adaptability with Tuning Payoff

ANNs demonstrated strong and consistent performance across all datasets, particularly when tuned. In DB3 and DB4, both involving continuous and categorical variables, ANNs achieved the best overall results. These results show that ANNs benefit from increased data granularity. The top configuration in DB3 (grid search with the *lbfgs* solver) reached an R^2 of 0.9950 and

the lowest RMSE across all models (18 060€). In DB4, ANN again led all models in RMSE and MAE after tuning.

The benefits of hyperparameter optimisation were clear: improvements in MAE of over 1 300€ were common, and performance gains were consistent across different tuning strategies. While the tuning process was time-consuming, retraining a known ANN architecture is relatively fast once the optimal configuration is found. This makes the model a practical option in scenarios where both prediction quality and flexibility are valued.

The ANN's capacity to model non-linear interactions and its adaptability to mixed data types explain its consistent top-tier results. However, the model's lower interpretability may be a limitation in some ETO contexts, especially where stakeholders require explainable predictions.

6.6 Data Requirements and Organisational Readiness

The successful use of ML for ETO cost estimation depends strongly on the quality and consistency of the data available. As the results show, model performance is not determined by algorithm complexity alone, but also by the presence of meaningful patterns in the data. Models like ANN and XGBoost only outperformed simpler alternatives when they had structured data to learn from. In the absence of that structure, even the most powerful models struggled.

To achieve accurate predictions, ETO companies must ensure that their historical data is structured, complete, and reliable. This involves standardising how data is collected and recorded. Many companies rely on informal tools such as spreadsheets, which often lack consistent formats and contain embedded comments or manual notes. Others use internal forms or systems with text fields that allow long, unstructured responses written in free form. These practices make it difficult to automate learning or apply ML methods reliably, as they hinder consistent data extraction and interpretation.

Integrating ML into the quotation process also requires organisational readiness. Models should not aim to replace cost engineers, but to assist them. For example, an ML model could provide a first estimate or cost range based on past projects, giving human experts a starting point and improving both speed and consistency. This hybrid approach respects the value of human expertise while adding structure and automation to the estimation process.

The most successful integration happens when ML tools are aligned with company processes. That means not only building the model but also maintaining it over time. As customer needs or project types change, the data must be updated and the model retrained. Without this support, performance will degrade over time.

6.7 Practical Implementation and Integration

For companies with limited or poor-quality data, adopting ML may seem out of reach. However, a phased strategy can make implementation more manageable and effective. One possible approach starts with the use of synthetic data, combined with a simple model like linear regression.

If the company can define the key variables that influence cost and establish a repeatable method for calculating those costs using a logic that aligns with its operations, it can simulate a small dataset that mimics realistic projects. Linear regression can then be applied to this synthetic data as a starting point. This has two advantages: it allows the company to experiment with ML in a low-risk setting, and it sets up the technical infrastructure for future improvements.

In parallel, the company should create a process to collect structured data from real projects going forward. This includes digitising inputs, enforcing consistent formats, and removing

ambiguous or free-text fields. Over time, these real records will replace the synthetic entries. Once the majority of the dataset consists of real project data, the company can begin to test more advanced models like ANN or XGBoost.

This progressive strategy reduces implementation risk and avoids placing unrealistic demands on the data at early stages. It also helps build trust within the organisation by starting with transparent models and gradually increasing complexity as confidence and data maturity grow.

6.8 Closing Reflections and Trade-offs

The findings in this research underline a key principle: there is no single best model for all situations, but ML clearly brings value across different contexts. Performance depends on the structure and quality of the data, as well as the goals and constraints of the organisation. Simple models like linear regression can outperform complex ones in noisy environments, while models like ANN and XGBoost show their strengths when the data is rich and well organised.

Tree-based models, especially random forest, proved to be highly dependent on structured input. Their generalisation suffered significantly when the data lacked clear patterns. This highlights the fact that high predictive performance is not only a matter of algorithmic power but also of data suitability. Companies must therefore see data quality and engineering not as a technical detail, but as a strategic investment.

At the same time, trade-offs must be considered. More complex models tend to offer better accuracy in structured environments, but they also come with greater training time, tuning effort, and lower interpretability. In ETO settings, where cost estimates carry financial risk, model transparency and ease of use are often just as important as accuracy.

In conclusion, the success of ML in ETO cost estimation will not come from selecting the best algorithm in isolation, but from aligning the model, the data, and the organisation’s capabilities. With the right data and a clear deployment strategy, companies can make informed use of ML, gradually moving toward more advanced tools as their data improves and confidence grows.

7 Conclusion

This thesis explored the application of ML to cost estimation in ETO products. By testing four algorithms across datasets of varying structure and complexity, this research aimed to evaluate the feasibility, reliability, and practical value of ML in predicting project costs. The following sections summarise the key findings, address the research questions, and reflect on the implications, limitations, and future opportunities identified throughout the research. In addition, the chapter discusses how the findings contribute to broader sustainability goals.

7.1 Summary of Findings and Contribution

This thesis set out to investigate whether ML could support accurate and consistent cost estimation in ETO products. Using four different algorithms applied to synthetic datasets with varying levels of structure and complexity, this research explored both technical performance and practical relevance.

The results show that ML can indeed offer significant value in ETO cost estimation. In structured and well-organised datasets, advanced models such as ANNs and XGBoost achieved very high levels of accuracy, often with R^2 values above 0.99. This indicates that, when data quality is high, ML models are capable of producing reliable and precise estimates with minimal tuning. In contrast, in noisy or randomised datasets, simpler models like linear regression performed better, showing more resilience to poor data conditions.

The research also examined the role of hyperparameter tuning. In structured datasets, tuning provided only small improvements. However, in less structured cases, especially for ANN and XGBoost, tuning led to clear performance gains, showing that optimisation is most useful when data presents more uncertainty. Still, this comes with a trade-off in terms of increased training time and reduced interpretability.

In addressing its main objectives, the research:

- Demonstrated that ML models can be effectively applied to ETO cost estimation;
- Showed that tuning is useful in some contexts but not essential in all;
- Highlighted the strong link between model performance and data quality, structure, and consistency.

This work adds to current academic and industrial discussions by offering both empirical evidence and practical guidance. It shows that ML has the potential to be used not just in theory but also in practice, especially when companies take steps to prepare and structure their data. The findings also provide a possible path forward for companies with limited digital maturity, offering a realistic way to adopt ML tools gradually.

7.2 Answers to Research Questions

To ensure alignment between the research objectives and the outcomes, this section provides direct answers to the three research questions defined in the introduction and the main research question. Each point below addresses one question individually, summarising the relevant findings and highlighting how they contribute to a deeper understanding of the applicability of ML to cost estimation in ETO environments.

Can machine learning techniques be used effectively to estimate the cost of engineer-to-order products?

The results of this research support a positive answer to the main research question. ML techniques can be used effectively to estimate the cost of ETO products, especially when data is structured and relevant cost drivers are well defined. Across all experiments, ML models showed strong predictive performance under a variety of data conditions.

Importantly, this research also demonstrated that ML can be useful even in the absence of real historical data. Through the construction and use of synthetic datasets, it was possible to simulate realistic estimation scenarios and evaluate how different algorithms respond to varying data challenges. The results suggest that, with thoughtful design and the right modelling choices, ML can support early-stage cost estimation in ETO contexts.

While real-world validation remains a step for future work, this research shows that ML is not just theoretically applicable to ETO environments. It can offer practical value, provided data limitations are addressed, and appropriate strategies are followed.

RQ1: How can synthetic datasets be constructed to simulate cost estimation scenarios in ETO environments?

In ETO contexts, reliable historical cost data is often lacking, which limits the use of traditional data-driven approaches. Synthetic datasets provide a practical solution by simulating realistic project data. This begins by identifying key input variables commonly used in quotations and generating artificial values using rule-based logic or randomisation.

This approach allows full control over data structure, variability, and complexity, making it easier to test how different ML models perform under varied conditions. It helps assess model robustness and practical suitability in scenarios that reflect industrial challenges.

Because synthetic data is fully transparent, it also enables targeted experimentation. Researchers can analyse how specific variables affect model behaviour, something that is harder with real-world datasets where hidden biases and missing entries are common.

Finally, synthetic data can help prototype ML workflows before real data is available. For companies with limited digital maturity, this creates a low-risk path to adoption and positions synthetic datasets as useful tools to explore and prepare for ML-based cost estimation in ETO settings.

RQ2: Which ML models are most appropriate for estimating costs in the context of ETO products, using synthetic data?

The research shows that the suitability of ML models for ETO cost estimation depends strongly on data structure and consistency. In structured datasets with well-defined relationships (DB1 and DB3), ANNs and XGBoost achieved the highest accuracy, capturing complex, non-linear patterns effectively. These models are well suited to contexts where input data is clean, complete, and stable.

In unstructured or randomised datasets (DB2 and DB4), XGBoost performed best overall. It showed resilience to noise, flexibility under different data conditions, and significant improvements when tuned. Across all scenarios, XGBoost proved to be the most balanced model, offering strong predictive power in both structured and less predictable settings.

RQ3: What are the limitations and strengths of using ML models in this context, particularly in the absence of real historical cost data?

ML models are a flexible alternative to traditional cost estimation in ETO contexts. They can model complex patterns and yield accurate predictions.

One of the main strengths of using ML in the absence of historical data is the ability to simulate realistic estimation conditions through synthetic datasets. In this research, models like XGBoost and ANN showed strong predictive performance not only in structured, rule-based scenarios but also in more randomised and unstructured datasets. This adaptability highlights their robustness to different data conditions.

Moreover, ML enables companies to move forward with estimation even without recorded project histories, as long as the cost drivers and a way to approximate cost are known. This can significantly reduce dependence on manual effort and expert judgement. Once trained, ML

models can produce estimates rapidly, freeing staff for higher-value tasks. The wide availability of ML algorithms also offers flexibility: organisations can test multiple models in parallel and select those best aligned with their operational needs.

However, these advantages come with trade-offs. ML models often function as black boxes, offering limited transparency into how predictions are made. This lack of interpretability can pose challenges in environments where traceability and justification are important. Additionally, companies with limited digital infrastructure may face barriers to implementation, including data preparation, integration, and cultural resistance to automation.

7.3 Contribution to Sustainable Development Goals

Although this thesis is focused on a technical problem, it also contributes to broader global priorities, particularly those outlined in the United Nations Sustainable Development Goals (SDGs). The work supports two SDGs in particular: Goal 9 (Industry, Innovation and Infrastructure) and Goal 12 (Responsible Consumption and Production).

Goal 9 emphasises the importance of building resilient infrastructure, promoting inclusive and sustainable industrialisation, and fostering innovation. This research contributes to that goal by exploring the application of ML as a means of modernising and improving cost estimation practices in industrial settings. Traditional estimation methods in ETO manufacturing often depend on informal knowledge, subjective judgement, or historical analogies that may not be reliable. By introducing and evaluating data-driven techniques, this thesis helps lay the groundwork for more structured, scalable, and efficient decision-making processes in a context where variability and uncertainty are high. This can support the digital transformation of manufacturing environments, especially in small and medium enterprises that may lack the resources to implement large-scale enterprise systems.

Goal 12 is addressed through the potential practical impact of more accurate and transparent cost estimation. When companies can estimate costs more reliably, they are better equipped to allocate resources, reduce overengineering, avoid excessive procurement, and plan production processes more efficiently. This minimises waste, both in terms of materials and time, and encourages more sustainable operational planning. The use of synthetic datasets, in particular, shows that it is possible to create structured approximations of real industrial conditions and develop useful predictive models even in the absence of historical data. This lowers the entry barrier for companies with limited digital maturity and enables incremental improvements towards data-driven sustainability.

In short, while the main contribution of this work is technical, it also opens avenues for more sustainable, efficient, and inclusive industrial practices. These connections reinforce the value of applied research not only in addressing specific operational problems, but also in contributing to global development goals.

7.4 Research Limitations

While this research offers promising results, several limitations must be acknowledged.

First, the research focused exclusively on supervised regression models. Other modelling approaches, including those designed to handle incomplete or uncertain data, were not explored but may be relevant in future research.

Second, the research did not include direct input from industry professionals or test the models using real company data. This limits how easily the results can be generalised or applied in real-world business situations, particularly with respect to organisational practices and operational constraints.

Finally, although the research outlines a possible implementation strategy for companies with limited data, it does not include empirical testing of this approach in industrial environments. Practical aspects such as system integration, scalability, and the actual resources required for training and deployment were not evaluated. These factors are often critical for successful adoption in practice and should be addressed in future work.

7.5 Opportunities for Future Work

This research opens several avenues for further investigation.

One clear next step is to test the developed models on real ETO company data. This would allow validation of the results in practical settings and provide deeper insights into the challenges of implementation, particularly regarding data availability, structure, and consistency.

Another opportunity lies in expanding the range of algorithms and techniques. For example, future work could explore classification models instead of regression, particularly in cases where the goal is to categorise projects into cost brackets or risk levels rather than predict exact values.

Future research could also focus on developing hybrid systems, combining ML predictions with rule-based logic or expert feedback to balance accuracy with transparency. In ETO environments, such approaches may be more acceptable than purely data-driven models.

These directions not only extend the technical scope of this work but also offer a practical path forward for making ML tools more useful and acceptable in real-world ETO environments.

References

Anwar, Lawin, and Sandra Barr\ıa Colliander. 2023. 'The Road to an Accurate Cost Estimation in Engineer-to-Order Manufacturing-A Case Study of Cost Estimation Accuracy within Advanced Technology Development in a Defence and Security Company'.

Caputo, Antonio C., and Pacifico M. Pelagagge. 2008. 'Parametric and Neural Methods for Cost Estimation of Process Vessels'. *International Journal of Production Economics* 112 (2): 934–54. https://doi.org/10.1016/j.ijpe.2007.08.002.

Elmousalami, Haytham H. 2021. 'Comparison of Artificial Intelligence Techniques for Project Conceptual Cost Prediction: A Case Study and Comparative Analysis'. *IEEE Transactions on Engineering Management* 68 (1): 183–96. https://doi.org/10.1109/TEM.2020.2972078.

Fortes, Carlos S., Alexandra B. Tenera, and Pedro F. Cunha. 2023. 'Engineer-to-Order Challenges and Issues: A Systematic Literature Review of the Manufacturing Industry'. *Procedia Computer Science* 219:1727–34. https://doi.org/10.1016/j.procs.2023.01.467.

Gosling, Jonathan, and Mohamed M. Naim. 2009. 'Engineer-to-Order Supply Chain Management: A Literature Review and Research Agenda'. *International Journal of Production Economics* 122 (2): 741–54. https://doi.org/10.1016/j.ijpe.2009.07.002.

Gunduz, Murat, Latif Onur Ugur, and Erhan Ozturk. 2011. 'Parametric Cost Estimation System for Light Rail Transit and Metro Trackworks'. *Expert Systems with Applications* 38 (3): 2873–77. https://doi.org/10.1016/j.eswa.2010.08.080.

Hammann, Dominik. 2024. 'Big Data and Machine Learning in Cost Estimation: An Automotive Case Study'. *International Journal of Production Economics* 269 (March):109137. https://doi.org/10.1016/j.ijpe.2023.109137.

Liu, Chih-Yu, Cheng-Yu Ku, Ting-Yuan Wu, Yu-Jia Chiu, and Cheng-Wei Chang. 2025. 'Liquefaction Susceptibility Mapping Using Artificial Neural Network for Offshore Wind Farms in Taiwan'. *Engineering Geology* 351 (May):108013. https://doi.org/10.1016/j.enggeo.2025.108013.

Mandolini, Marco, Luca Manuguerra, Mikhailo Sartini, Giulio Marcello Lo Presti, and Francesco Pescatori. 2024. 'A Cost Modelling Methodology Based on Machine Learning for Engineered-to-Order Products'. *Engineering Applications of Artificial Intelligence* 136 (October):108957. https://doi.org/10.1016/j.engappai.2024.108957.

Matel, Erik, Faridaddin Vahdatikhaki, Siavash Hosseinyalamdary, Thijs Evers, and Hans Voordijk. 2022. 'An Artificial Neural Network Approach for Cost Estimation of Engineering Services'. *International Journal of Construction Management* 22 (7): 1274–87. https://doi.org/10.1080/15623599.2019.1692400.

Meharie, Meseret, and Nagaraju Shaik. 2020. 'Predicting Highway Construction Costs: Comparison of the Performance of Random Forest, Neural Network and Support Vector Machine Models'. *Journal of Soft Computing in Civil Engineering* 4 (2). https://doi.org/10.22115/scce.2020.226883.1205.

Özcan, Burcu, and Alpaslan Fığlalı. 2014. 'Artificial Neural Networks for the Cost Estimation of Stamping Dies'. *Neural Computing and Applications* 25 (3–4): 717–26. https://doi.org/10.1007/s00521-014-1546-8.

Padala, S.P. Sreenivas, and Anshul Goyal. 2025. 'Early Stage Cost Prediction Model for Indian Building Construction Projects Using Artificial Neural Networks'. *Journal of Financial Management of Property and Construction*, February. https://doi.org/10.1108/JFMPC-12-2023-0085.

Rapaccini, Mario, Veronica Loew Cadonna, Leonardo Leoni, and Filippo De Carlo. 2023. 'Application of Machine Learning Techniques for Cost Estimation of Engineer to Order Products'. *International Journal of Production Research* 61 (20): 6978–7000. https://doi.org/10.1080/00207543.2022.2141907.

Verlinden, B., J.R. Duflou, P. Collin, and D. Cattrysse. 2008. 'Cost Estimation for Sheet Metal Parts Using Multiple Regression and Artificial Neural Networks: A Case Study'. *International Journal of Production Economics* 111 (2): 484–92. https://doi.org/10.1016/j.ijpe.2007.02.004.

Ziegler, Cedric C., Alexander Dobhan, and Moritz Heusinger. 2022. 'Applications of Neural Networks in Engineer-to-Order Environment'. *Procedia CIRP* 112:140–45. https://doi.org/10.1016/j.procir.2022.09.052.

APPENDIX A: Sample Project Instances from Synthetic Datasets

This appendix presents a sample of four project instances from each of the four synthetic datasets developed for this research (DB1 to DB4). The tables include the six input variables used throughout the modelling process, along with the corresponding estimated cost for each project. These examples are intended to offer the reader a clearer view of the data used in the experiments, complementing the methodological description provided in Chapter 4. They serve as a practical illustration of the dataset contents without requiring reference to the full data files.

Table A.1 presents four project instances from DB1, a dataset generated using predefined rules and fully categorical variables. In this case, the values follow a structured dependency chain. For example, Project Familiarity (PF) influences Technological Complexity (TC), which in turn affects Number of System Features (SF). The second row illustrates a minimal configuration across all variables, resulting in the lowest estimated cost. In contrast, the remaining records show more typical mid-complexity scenarios, where the interplay between variables like Know-How Reuse (KH) and TC leads to consistent patterns in Project Duration (PD) and Labour to Material Ratio (LM). These examples reflect the internal logic imposed by the heuristic strategy used in DB1.

Table A.1 - Sample of project instances from DB1 (heuristic, categorical)

Project Familiarity	Technological Complexity	Number of System Features	Know How-Reuse	Project Duration	Labour to Material Ratio	Estimated Cost (€)
3	3	2	3	3	3	660 295
1	1	1	1	1	1	165 631
4	3	2	3	3	3	701 534
1	3	4	2	3	2	531 631

Table A.2 presents four project instances from DB2, a dataset generated using random logic and fully categorical variables. In this dataset, all inputs were assigned independently and randomly. For example, the first record combines high values of TC and SF with a minimal PD, a combination unlikely under structured logic. These inconsistencies reflect the intentionally unstructured nature of DB2 and serve as a contrast to the more coherent patterns found in datasets like DB1.

Table A.2 - Sample of project instances from DB2 (random, categorical)

Project Familiarity	Technological Complexity	Number of System Features	Know-How Reuse	Project Duration	Labour to Material Ratio	Estimated Cost (€)
4	3	4	3	1	3	697 924
1	4	4	2	4	3	723 581
3	4	1	3	3	1	529 342
2	3	2	4	4	1	612 739

Table A.3 presents four records from DB3, the dataset generated using predefined rules and including both categorical and continuous variables. For example, the first project combines high values of PF and TC, resulting in a large number of SF (30), a long PD (56 weeks), and a high LM (41%). In contrast, the final row shows a low-complexity case with minimal features and effort. These examples help illustrate how structured dependencies and numeric variation coexist within DB3.

Table A.3 - Sample of project instances from DB3 (heuristic, mixed)

Project Familiarity	Technological Complexity	Number of System Features (units)	Know-How Reuse	Project Duration (weeks)	Labour to Material Ratio (%)	Estimated Cost (€)
3	4	30	3	56	41	739 709
3	2	10	2	45	20	480 097
2	1	2	1	9	1	232 527
1	1	2	1	12	4	194 971

Table A.4 presents four project instances from DB4, a dataset created using random generation and mixed variable types. In this case, all values were assigned independently, without any predefined relationships. For instance, the second record shows a highly complex project (PF = 4, TC = 4, SF = 30) with an unexpectedly short duration of just 5 weeks. Such cases highlight the absence of internal structure in DB4 and help contrast it with the more coherent patterns observed in DB3.

Table A.4 - Sample of project instances from DB4 (random, mixed)

Project Familiarity	Technological Complexity	Number of System Features (units)	Know-How Reuse	Project Duration (weeks)	Labour to Material Ratio (%)	Estimated Cost (€)
3	4	29	3	64	60	803 718
4	4	30	2	5	19	602 847
2	2	9	2	56	24	436 293
4	3	12	1	11	4	454 034