Aalto University
School of Business

STRATEGIC PROJECT PORTFOLIO MANAGEMENT BY PREDICTING PROJECT PERFORMANCE AND ESTIMATING STRATEGIC FIT

Joona Åström

Master's Thesis
Department of Information and Service Management

Instructors: Juuso Liesiö & Eeva Vilkkumaa
Year: 2020

Aalto University
School of Business

Aalto University, P.O. BOX 11000, 00076 AALTO
www.aalto.fi

Abstract of master's thesis

Author Joona Åström

Title of thesis Strategic Project Portfolio Management by Predicting Project Performance and Estimating Strategic Fit

Degree Master of Science in Economics and Business Administration

Degree programme Information and Service Management

Thesis advisor(s) Juuso Liesiö & Eeva Vilkkumaa

Year of approval 2020 **Number of pages** 106 **Language** English

Abstract

Candidate project selections are extremely crucial for infrastructure construction companies. First, they determine how well the planned strategy will be realized during the following years. If the selected projects do not align with the competences of the organization major losses can occur during the projects' execution phase. Second, participating in tendering competitions is costly manual labour and losing the bid directly increase the overhead costs of the organization. Still, contractors rarely utilize statistical methods to select projects that are more likely to be successful. In response to these two issues, a tool for project portfolio selection phase was developed based on existing literature about strategic fit estimation and project performance prediction.

One way to define the strategic fit of a project is to evaluate the alignment between the characteristics of a project to the strategic objectives of an organisation. Project performance on the other-hand can be measured with various financial, technical, production, risk or human-resource related criteria. Depending on which measure is highlighted, the likelihood of succeeding with regards to a performance measure can be predicted with numerous machine learning methods of which decision trees were used in this study. By combining the strategic fit and likelihood of success measures, a two-by-two matrix was formed. The matrix can be used to categorize the project opportunities into four categories, *ignore, analyse, cash-in* and *focus,* that can guide candidate project selections.

To test and demonstrate the performance of the matrix, the case company's CRM data was used to estimate strategic fit and likelihood of succeeding in tendering competitions. First, the projects were plotted on the matrix and their position and accuracy was analysed per quartile. Afterwards, the project selections were simulated and compared against the case company's real selections during a six-month period.

The first implication after plotting the projects on the matrix was that only a handful of projects were positioned in the *focus* category of the matrix, which indicates a discrepancy between the planned strategy and the competences of the case company in tendering competitions. Second, the tendering competition outcomes were easier to predict in the low strategic fit quartiles as the project selections in them were more accurate than in the high strategic fit categories. Finally, the matrix also quite accurately filtered the worst low strategic fit projects out from the market.

The simulation was done in two stages. First, by emphasizing the likelihood of success predictions the matrix increased the hit rate and average strategic fit of the selected project portfolio. When strategic fit values were emphasized on the other hand, the simulation did not yield useful results.

The study contributes to the project portfolio management literature by developing a practice-oriented tool that emphasizes the strategical and statistical perspectives of the candidate project selection phase.

Keywords Project Portfolio Management, Strategic Fit, Project Performance Prediction

Aalto-yliopisto
Kauppakorkeakoulu

Aalto-yliopisto, PL 11000, 00076 AALTO
www.aalto.fi

Maisterintutkinnon tutkielman tiivistelmä

Tekijä Joona Åström

Työn nimi Strategista projektiportfolion hallintaa ennustamalla projektin suoriutumista ja strategista sopivuutta

Tutkinto Kauppatieteiden maisteri

Koulutusohjelma Tieto- ja palvelujohtaminen

Työn ohjaaja(t) Juuso Liesiö & Eeva Vilkkumaa

Hyväksymisvuosi 2020 **Sivumäärä** 106 **Kieli** englanti

Tiivistelmä

Kandidaattiprojektien valinta on äärimmäisen kriittistä infrarakentamisen palveluita tarjoaville yrityksille. Se määrittää kuinka hyvin yritysten suunniteltu strategia toteutuu seuraavien vuosien aikana. Jos valinnat eivät ole linjassa organisaatioiden voimavarojen kanssa projektien toteutusvaiheessa saattaa realisoitua suuria tappioita. Tarjouskilpailuihin osallistuminen myös vaatii kallista manuaalista työtä, jolloin tappiot tarjouskilpailuissa kasvattavat suoraan yritysten välillisiä kustannuksia. Silti urakoitsijat harvoin hyödyntävät tilastotieteellisiä menetelmiä todennäköisten tarjouskilpailuvoittojen ennustamiseen. Työssä kehitettiin projektiportfoliohallinnan työkalu näiden ongelmien ratkaisemiseksi pohjautuen kirjallisuuteen projektien strategisen sopivuuden arvioinnista ja projektien suoriutumisen ennustamisesta.

Projektin strateginen sopivuus voidaan määrittää sen perusteella, kuinka hyvin sen ominaisuudet vastaavat organisaation strategisia tavoitteita. Projektin suoriutumista taas voidaan mitata erilaisilla taloudellisilla, teknisillä, tuotannollisilla ja riskeihin tai henkilöstöön liittyvillä kriteereillä. Riippuen siitä mitä kriteereitä tarkastellaan, projektin onnistumista voidaan ennustaa koneoppimismenetelmillä, joista päätöspuita hyödynnettiin tässä tutkimuksessa. Yhdistämällä projektin strategisen sopivuuden arviot ja todennäköisyys sen onnistumiselle tutkimuksessa muodostettiin matriisi, jolla voidaan ohjata kandidaattiprojektivalintoja luokittelemalla projektit neljään kategoriaan: *vältä, analysoi, rahasta* ja *keskity*.

Tapausyrityksen asiakashallintajärjestelmän tietoa käytettiin projektien strategisten sopivuuksien mittaamiseen ja tarjouskilpailun onnistumisen todennäköisyyden määrittämiseen, sekä matriisin muodostamiseen ja testaamiseen. Ensiksi projektien asemia ja tarkkuuksia matriisin jokaisessa neljänneksessä analysoitiin. Sitä seuranneessa simulaatiossa matriisin annettiin tehdä projektivalinnat kuuden kuukauden ajalle, jota verrattiin tapausyrityksen projektivalintoihin.

Ensimmäinen tulos oli, että matriisin *keskity*-kategoriaan osui vain muutama projekti. Tämä viittaa epäjohdonmukaisuuteen tapausyrityksen suunnitellun strategian ja sen voimavarojen välillä. Toisena ilmeni, että projektien onnistumisen ennusteet olivat huomattavasti tarkemmat alhaisten strategisten sopivuuksien kategorioissa. Matriisi oli myös melko tarkka suodattamaan projektit, jotka olivat epätodennäköisiä onnistumaan ja omasivat alhaisen strategisen sopivuuden.

Seuraavaksi simulaatio toteutettiin kahdessa vaiheessa. Ensimmäisessä iteraatiossa annettiin suurempi painoarvo tarjouskilpailun onnistumisen todennäköisyysarvoille, jolloin matriisin valinnat nostivat, sekä tarjouskilpailujen onnistumisen todennäköisyyttä, että projektiportfolion keskimääräistä strategista sopivuutta. Toisessa suurempi painoarvo annettiin strategisille sopivuuksille, jolloin matriisista ei valitettavasti saatu hyödyllisiä tuloksia.

Tutkielma edistää projektiportfoliohallinnan tutkimusta kehittämällä käytäntöön pohjautuvan työkalun, joka korostaa strategista ja tilastotieteellistä näkökulmaa kandidaattiprojektienvalinta vaiheessa.

Avainsanat projektiportfoliohallinta, strateginen sopivuus, projektin suoriutumisen ennustus

CONTENTS

LISTS OF TABLES AND FIGURES

LIST OF ABBREVIATIONS

AdaBoost	*Adaptive boosting*
AUC	*Area under the curve*
Bagging	*Bootstrap aggregation*
CART	*Classification and regression trees*
CHAID	*Chi-square automatic interaction detector*
CRM	*Customer relationship management*
Dom	*Domain*
DSR	*Design science research*
DTC	*Decision tree classifier*
FN	*False negative*
FP	*False positive*
GDP	*Gross domestic product*
IC	*NCEC's Infrastructure construction segment*
IRR	*Internal rate of return*
IS	*Information Systems*
KPI	*Key performance indicator*
LoS	*Likelihood of Success*
M&A	*Mergers and acquisitions*
MoM	*Mean of Maxima*
NBAL	*NCEC’s Baltic operations division*
NBAL-EE	*NBAL's Estonian business unit*
NBAL-LT	*NBAL's Lithuanian business unit*
NBAL-LV	*NBAL's Latvian business unit*
NCEC	*Nordic Construction Engineering Company*
NISE	*NCEC’s Industrial, structural and energy engineering division*
NNOR	*NCEC’s Norwegian operations division*
NNOR-TS	*NNOR's Traffic and Special Engineering business unit*
NPV	*Net present value*
NPS	*Net promoter score*
NRFE	*NCEC’s Rock tunnelling and foundation engineering division*
NRFE-FS	*NRFE's Foundation and Special Engineering business unit*
NSRB	*NCEC’s Street, railroad and bridge engineering division*
NSWE	*NCEC’s Swedish operations division*
PPM	*Project portfolio management*
ROC	*Receiving operating characterstic*
ROI	*Return on investment*
SF	*Strategic Fit*
TN	*True negative*
TP	*True positive*

NOTATIONS

Upper case letters denote constants and lower-case letters dummy indexes. Throughout the study **boldface** is used to denote vectors and matrices.

Arrays and vectors

a	a scalar (or a variable)
$\boldsymbol{a}$	a vector with n number of values $(a_1, a_2, \dots, a_n)$
$\boldsymbol{A}$	a matrix with the size of $m \times n$

Indexing

x_i	element i of the feature vector $\boldsymbol{x}$
$\boldsymbol{x}_i$	element i from the feature matrix $\boldsymbol{X}$
$\boldsymbol{X}_{i,j}$	element i, j of the feature matrix $\boldsymbol{X}$

Functions

$dom(y)$	the domain of target variable y, which contains all the possible classes of $y \in \{c_1, c_2, \dots, c_n\}$
$\lvert\mathbb{X}\rvert$	cardinality of a set i.e. the number of (non-distinct) elements
$\mathcal{L}$	a loss function

Specific machine learning notations

C	number of classes of target variable y i.e. its cardinality
D	dimensionality of a data vector
N	number of instances
T	a tree T (or the data set in tree T)
t	node t (or the data set in node t) such that $\mathbb{X} = \bigcup_{t \in T} t$

$\mathcal{X}$	a feature space
$\mathcal{Y}$	a label space
$\mathcal{U}$	the universal instance space defined as $\mathcal{X} \times \mathcal{Y}$
$\mathcal{H}$	a hypothesis space
$\mathscr{h}$	a classifier, which can produce predictions with a feature vector $\boldsymbol{x}$ as $\mathscr{h}(\boldsymbol{x}) = \hat{y}$

$\omega_T^{(i)}$	weight of i-th instance in tree T
ε	generalization error (or misclassification rate)

Sets and datasets

$\boldsymbol{x}^{(i)}$	a feature vector of the i-th instance from a dataset
$y^{(i)}$	the target label of the i-th instance from a dataset
$\hat{y}$	a predicted label

$\mathbb{X}$	a set of instances
$\mathbb{X}^{train}$	a set of training instances $\{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$ with N number of instances

Probabilities and distributions

$P(\boldsymbol{x}, y)$	a joint probability distribution for $\boldsymbol{x}$ and y
$p(c)$	proportion of class c

1 INTRODUCTION

This research will present a case study about project portfolio selection process from the perspective of a large Nordic construction contractor and specifically its infrastructure project segment. This first section will cover the main motives and the context behind the thesis as they will provide the foundation, which the rest of the thesis is built upon. Based on this setting the research questions are then derived and finally, the structure for the rest of the thesis will be presented as a guide for the reader for the remainder of the thesis.

1.1 MOTIVATION AND BACKGROUND

Construction contractors that compete for complex infrastructure construction projects must find a balance between the quality and the price of the proposal as having the lowest bid does not guarantee the win in a tendering competition. More complex the project more the other factors than price are emphasized. The difficulty of selecting the right projects to be tendered causes contractors to make bidding choices that first, do not align with the intended strategic objectives and second, are unlikely to be successful (Martinsuo, 2013; Mintzberg, 1992). Hence, the candidate project selection phase is a crucial stage for infrastructure construction contractors.

Various factors contribute to the complexity and unpredictability of tendering competitions. First of all, multiple parties will often be involved in the decision-making process on the client's side and especially the larger projects can be comprised of multiple rounds of workshops and negotiations even before the contractors receive an invitation to tender the actual project (Finlex, 2016). If the contractor wishes to proceed in the tendering competition after receiving the invitation to tender, they have to prepare the required documents manually. Often it includes multiple certificates, proof of client references and a demanding cost calculation phase that can last many months if the project is extensive. Furthermore, in some cases the final design of the project does not come from the client, but instead must be provided by the contractor. And even if the client provides the design, there are often changes that the contractor has to propose to make the design executable. All of this work has to be done and covered by the contractor. The sum of the quality of design, tendering documents, previous references projects and price among other factors will decide the winner. See e.g. Lahdenperä (2009) for a good summary of the process used in alliance projects in Finland.

As demonstrated, taking part in a tendering competition is a complex and costly undertaking, which emphasizes the importance of candidate project selection phase for a construction company. Still often managers do not utilise statistical tools in alleviating the uncertainty related to tender competitions and every so often rely on intuition instead (Martinsuo, 2013).

The case company, NCEC's infrastructure segment is one of the oldest infrastructure construction companies in the Nordics. Infrastructure segment has operations in six different countries around the Nordics and Eastern Europe and are part of the larger NCEC group that operates in over 10 different countries and conducts projects varying from high-rise office buildings to long and deep underground tunnels. Even though, they have an established process for candidate project selection, they are lacking a clear framework to categorize and segment different opportunities in the market in this crucial step of the project portfolio management process. This is not an uncommon issue in the project portfolio management area as there are no silver-bullet solutions in the literature for selecting the optimal candidate projects. Without proper tools in place, managers can be influenced by internal power play, gutfeel and subjective opinions of themselves and others when deciding on which projects to undertake. If the project selections are made based on these grounds, the organisation might easily drift in an unintended direction as the implemented projects may not align with the strategic objectives set by the management or be necessarily aligned with what the organisation usually succeeds on (Martinsuo, 2013; Mintzberg, 1992).

1.2 Design Objectives and Questions

The goal of the thesis is to create a framework, which can be used in the candidate project selection phase. To tackle the previously mentioned common flaws in the candidate project selection process, the framework should help the case company to select projects that are both aligned with their strategic objectives and predicted to be favourable for them. It should thus, capture two separate aspects: first, the strategic alignment of the projects and second, the probability of selecting the most suitable projects with regards to the capabilities of the organisation. In other words, the three distinct research questions are:

1) *How to reliably estimate the strategic fit of a candidate project with the company's strategic objectives?*

2) *How to reliably estimate the probability of winning the tendering competition for a given candidate project?*
3) *How can these estimates be used to guide the process of selecting which candidate projects to pursue?*

In order to evaluate the framework, its performance will be tested in a simulation study by comparing the candidate project selections made with the framework based on a scoring model to the actual selections made by the management of the organisation. If the model's project selections are better than the management team's selections in terms of the strategic alignment and/or winning rate of projects, it accomplishes its purpose.

With this premise, the thesis will be contributing to the project portfolio management literature by developing a framework for prioritising candidate projects. This is done under the assumption that project-oriented organisations should select the projects that align with their strategic objectives and that are most likely to be won as suggested by scholars within the respective disciplines. In order to demonstrate the functionality and performance of the framework, it will be developed and implemented in a software, which is able to measure strategic fit and predict the likelihood of succeeding in a tendering competition. The projects will be visualized along the two measures and the software can, to an extent, automize and prioritise projects according to the framework with the provided scoring model. The development and implementation are done in cooperation with an established player within the industry to support real-world project portfolio management practices and processes.

1.3 Structure of the Thesis

This kind of a practical, yet experimental setting is suitable for a design science research methodology suggested in Peffers et al. (2006). Design science research in the Information Systems (IS) discipline is an applied research method, which aims to utilize theory, often from other research fields, in order to solve an actual problem found from the real-world. It is appropriate for research problems that are exploratory in nature as the output of the study cannot often be determined beforehand. Its end-product is a design artefact (referred to as the framework) that has its foundation in the academic literature, but has been applied and evaluated in a real-world setting (Peffers, et al., 2006). The framework can then be modified (if needed) and applied in novel contexts and studies. The thesis will be structured along the six-step design science research methodology suggested in the aforementioned study as follows:

1. Design problem identification and motivation (covered in 1.1: Motivation and Background)
2. Objectives of a solution (covered in 1.2: Design Objectives and Questions)
3. Design and development (covered in 2: Design and Development)
4. Demonstration (covered in 3: Demonstration)
5. Evaluation (covered in 4: Managerial Implications)
6. Communication

The main part of the thesis will focus on the strategic side of project portfolio management. The mathematical foundations for the methods that are utilized in the experiment are covered in the appendices.

2 DESIGN AND DEVELOPMENT

According to the DSR methodology proposed by Peffers et al. (2006), at the end of the design and development section the actual framework will be created. It is important to define the framework's architecture and functionality comprehensively in this section as the they should correspond to the research problems and objectives that were presented in Section 1.2. Therefore, the following sections will cover the theoretical background that are the basis for the framework created at the end of this design and development section.

2.1 PROJECT PORTFOLIO MANAGEMENT

Many empirical studies have examined the activities that project portfolio managers do on a day-to-day basis (Christiansen & Varnes, 2008; Blichfeldt & Eskerod, 2008), while still majority focus on the theoretical level of portfolio management (Martinsuo, 2013). However, it seems that there exists evidence about a gap between project portfolio management (PPM) in practice versus in the academia. Particularly, the frameworks and optimization models of the academia are often perceived to be hard to apply in a real-world setting as the decision-making context may either lack many of the input attributes, have constraints or differ in other ways making the theoretical frameworks inapplicable (Martinsuo, 2013). For example, many former project portfolio management studies are based on linear programming (Kumar, et al., 2007; Rad & Rowzan, 2018) system dynamics modelling (Rad & Rowzan, 2018; Love, et al., 2002) or other programming models (Tkáč & Lyócsa, 2010), which often assume certain stability in the production environment. When managers then attempt to apply these or similar frameworks in practice, it may be impractical or unrealistic to generate accurate results due to the uniqueness and instability of the decision-making context. To counter these short-comings, research has been suggested on more practical applications of project portfolio management, which could then be generalized on varying contexts, instead of assuming a specific stable problem setting (Engwall, 2003; Martinsuo, 2013).

In the PPM in practice literature, studies have suggested that instead of following formally defined guidelines and rules, portfolio managers make decisions based on personal opinions and power play in reality (Kester, et al., 2011). There are indications that project selection decisions and consequently project portfolio management practices are rather political and path-dependent than deliberate and rational (Martinsuo, 2013). Aaltonen (2010) too suggested that managers' intentions underlying portfolio decisions deserve further attention.

Cooper (1993) pointed out that many of the project portfolio selection methods demand too specific input data, they assume a certain stable environment or treat the risk inherent in the environment inappropriately. Still, there is solid evidence claiming that some selected project portfolio management practices such as strategic PPM methods and portfolio maps and matrices are associated with better portfolio performance compared to statistically unmanaged portfolio (Killen, et al., 2008). Indeed, evaluating the suitability of the project from strategic perspective instead of only judging them by one or two performance measures has gained popularity (Meskendahl, 2010; Archer & Ghasemzadeh, 1999; Thompson, 1967; Venkatraman, 1989).

2.1.1 Project Portfolio Management and Strategy

Conceptual research has clearly suggested that strategy has an influence on the success of the project portfolio management practice in organisations (Archer & Ghasemzadeh, 1999; Meskendahl, 2010). Khurana and Rosenthal (1997) highlighted the importance of the planning phase of strategy in portfolio management. Similarly, Hedley (1977) stressed the dangers of defining strategic objectives and the direction vaguely or poorly, as it makes it impossible for the managers to make any successful project portfolio selections that would be aligned with the intended strategy due to the unclarity.

This notion does not only stay at the conceptual level as practitioners perceive project portfolio management as the continuous practice of reviewing and selecting projects in order to translate the intended strategic objectives set by the management into realized strategy of the organisation (PMI, 2008). Archer & Ghasemzadeh (1999) argued that even though it may be hard to utilise some specific framework in the project portfolio selection process, some broad guidelines should be developed. One group of projects might not have accurate estimates of certain quantitative performance measures such as cost, cycle time or profit estimates available while the other group may lack something else. Although, it might be impossible to define a single universal performance measure for ranking the projects in every decision context, some objective evaluation guidelines should be derived in order to measure the suitability, or strategic fit, of a project for the organisation (Archer & Ghasemzadeh, 1999).

However, that is also one of the key challenges in the project portfolio management theory – it seems to be hard to align a portfolio of projects with the major corporate and business-level strategies. This is especially true in project-oriented organisations, as unlike in the product-oriented businesses, it may be hard, if not impossible, to influence what project opportunities there are in the operating environment (Archer & Ghasemzadeh, 1999). To draw an example from the context of this study, a single infrastructure construction company often has a rather limited influence on the decision makers' – usually the government's – project proposals, which compose the majority of the project opportunities. Therefore, it might be hard, if not impossible, to mould the projects to fit the strategic agenda of the management. As a consequence, the organisation is forced to adapt itself into the external environment. If a project-oriented organisation wins a bid and begins to execute the project, it may significantly affect the organisation's on-going projects and priorities, and almost accidentally drift the organisation in an unintended direction (Rad & Rowzan, 2018).

This notion nicely draws out one of the key motivations for this study. The strategic alignment of different projects in the operating environment must be evaluated already in the candidate project selection phase before preparing any bids to minimize the influence of emergent strategies to the organisation's realized strategy (Mintzberg, 1992; Mintzberg & Waters, 1985). As there often are thousands of opportunities in the market at any given time, this process should not expend too much time and resources either. Often the project-oriented companies may place a heavier focus on ensuring the success of execution than selecting the projects that are the most likely to be executed successful in the first place. Many studies argue however, that the project portfolio should be selected in a way that first of all aligns with the strategy, but also considers the resources and capabilities of the organisation to execute the projects successfully (Archer & Ghasemzadeh, 1999; Englund & Graham, 1999).

The issue that arises with optimizing a project portfolio with regards to resources is that projects and resources are often placed on different time dimensions. Projects span across varying timelines from very short sprints to decades lasting mega projects. Many of the frameworks then try to optimize these multidimensional projects based on fixed resources in a snapshot in time. On top of this, these frameworks often assume that the projects would compete for the same resources and that all of the relevant ones would be known and controlled by the company itself although that is often not the case (Artto, et al., 2008; Martinsuo & Lehtonen, 2009). The organisational structure of the company may limit the control over project resources, as frequently is the case in matrix organisations (Perks, 2007), and the interdependencies between projects are assumed to be fixed and will realize as planned (Nobeoka & Cusumano, 1995; Nobeoka & Cusumano, 1997; Prencipe & Tell, 2001).

These issues emerging from the dynamicity of the operational environment hinder the accuracy and applicability of portfolio optimization models in complex settings. In construction business, resources such as equipment and human capital can be hired externally, and work is often outsourced, which transforms the nature of resources into fuzzy adjustable constraints rather than fixed set of rules. Given this notion, optimizing a project portfolio based on fixed resource constraints can yield inaccurate and impractical solutions in such a real-world situation.

2.1.2 Project Portfolio Management and Matrices

Instead of thinking about the issue of project portfolio selection as a pure optimization problem, Archer & Ghasemzadeh (1999) noted that portfolio matrices can be a useful tool to map the project opportunities in a simpler way during the project portfolio selection process. With the Project Portfolio Selection framework (Figure 1), they aimed to ensure that the overarching strategy is always taken into consideration at each step in the optimal portfolio selection process. Matrices were suggested as a tool from the starting pre-processing stage up until the final portfolio adjustment phase demonstrating their utility in various situations. Although, matrices do not offer a silver-bullet solution for every situation they were perceived as a flexible visualisation method to segment the project space.

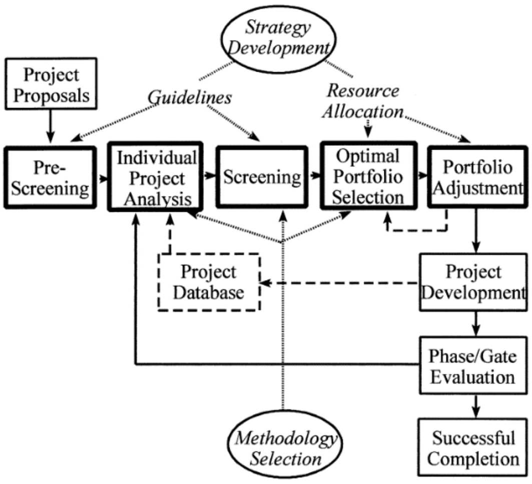


Figure 1: Framework for Project Portfolio Selection (from Archer & Ghasemzadeh, 1999)

Studies have also examined which mapping approaches and dimensions should be included in such a matrix in order to help the firm navigate towards succesfull implementation of intended strategies (Wheelwright & Clark, 1992). For example, strategic fit and portfolio management has been discussed previously by Hedley (1977) from the business portfolio management perspective. Though, Hedley wrote about the widely criticized Growth-share matrix or "BCG-matrix" the core idea behind it is still solid: any business should match their portfolio to match the external environment in order to gain competitive advantage.

BCG-matrix’s downfall lies in its assumptions about the two dimensions it uses, market growth and market share. First, the matrix assumes that the current market share of a company has a direct causality with the organisation's future ability to compete (Armstrong & Brodie, 1994). It seems that there is little empirical evidence to back this claim up for the reason that the current success of a company is not a guarantee of its future succes. This has been proved with the infamous downfalls of several past incumbents including Blockbuster, Kodak and Nokia to name a few. Second dimension, market growth, on the otherhand draws a direct causal relationship from it to the profitability of an organisation. Studies have shown that there is little empirical evidence to support this claim either as profitability does not depend on a single dimension but instead arises from the interplay between different factors in a competitive environment. Porter's famous Five Forces already provide multiple other factors that contribute to the profitability including the bargaining power of customers and suppliers, threat of emerging and invading rivals from other industries and internal organisational success factors to name a few (Porter, 1979; Wensley, 1981; Jacobson & Aaker, 1985).

Therefore, the assumptions related to the uncertainty of future resources and capabilities, market growth and market share among other issues have undermined the performance of previous portfolio management frameworks. Instead of relying on any of these assumptions, this study aims to use a predictive model and plot it against the strategic alignment of a project to forecast the success and fitness of projects. Next, these two measures will be defined and their suitability in the project portfolio management context will be justified.

2.2 Strategy and Strategic Fit

In the academia, a wide range of empirical studies has been made to study strategic fit. The concept has been applied from many different perspectives including the fit of M&A targets to the strategy of the acquirer (Chen, et al., 2018), fit of Information Systems (IS) strategy to the business strategy of the company (Chan, et al., 1997) and fit of Supply Chain Management IS to the competitive strategy of the organisation (McLaren, et al., 2004) to cite a few.

However, scholars do not agree unanimously on the definition of strategic fit. What makes defining it difficult is that the definition seems to be completely dependent on the context of the study. Perhaps its oldest form comes from the contingency theory, which defines strategic fit as the alignment between the structure of the organisation and the environment. The concept can be traced back to the book Organisations in Action by Thompson (1967). Thompson states that an organisation gains competitive advantage and alleviates the environmental uncertainty by having a strategy that fits the external environment. Therefore, it treats strategy as a variable which facilitates the alignment between internal resources and competences of the organisation and the external environment. Important observation in Thompson's definition is that often companies cannot change the environment drastically but can instead mould their strategy to fit the environment.

While Thompson highlights the importance of what can be achieved with proper strategy work, Henderson and Venkatraman (1993) underline the difficulty of this achievement. They claimed that reaching perfect strategy and strategic fitness is a hard, if not an impossible task as the environment is dynamic and in the state of constant evolvement. A strategy that was supposed to be nearly perfect after a planning session can turn out to be irrelevant the next morning due to the inherent change in the operating environment. Common theme for all of these definitions seems to be that strategic fit cannot be perceived as a Boolean fact describing whether one organisational structure, an M&A target or a project would perfectly fit the strategy of the organisation. Instead each possess a certain degree of fitness depending on various internal and external factors. Consequently, the measure for strategic fit should also be a continuous attribute.

American national standard for portfolio management suggests that strategic fit or the suitability of a project should be evaluated in contrast to the strategic goals set by the management team from the PPM perspective (PMI, 2008). Talantsev & Sundgren (2013) expanded upon this statement and defined fit as the "degree to which the project is relevant to and consistent with the strategic goals of the organisation". Consequently, managers can have a direct impact on the values of strategic fit by defining the strategic objectives appropriately according to the aforementioned definition.

Talantsev & Sundgren's approach also differs from the older methods such as Venkatraman's (1989) study about strategic fit as a profile deviation. In the study, the researchers aimed at deriving the ideal values of fitness from the historical profit performance of past projects. This and similar approaches that try to measure strategic fit based on merely historical performance are inherently exposed to common knowledge among financiers that past performance does not guarantee future results (see e.g. SEC Rule 156 (SEC, 2003)). Instead, when comparing the projects in relation to strategic goals, the goals itself include a prediction of what the future may bear for the organisation given that the strategy work behind the goals is conducted in an all-encompassing and rigorous manner (Clegg, et al., 2011). Hence, an evaluation of fitness in relation to strategic goals can provide a more future-oriented assessment by respecting the expert opinion and vision of managers and strategists of the organisation. Next, the exact way to measure the fitness will be defined in order to design a PPM framework which utilises the measure.

2.2.1 Measuring Strategic Fit

As the definition of strategic fit has changed from year to year and author to author, there are also almost as many ways to quantitatively and qualitatively measure fitness. For instance, Meilich (2006) used regression in order to estimate strategic fit, Beynon et al. (2010) used Classification and Ranking Belief Simplex (CaRBS) to model the strategic fit of public organisation and Chen et al. (2018) approximated potential merger and acquisition targets in the Chinese banking industry with Bayesian stochastic frontier model. Some studies have adopted Euclidian distance as a simpler measure for strategic fit as it just measures the distance between two different points in a n-dimensional space (Venkatraman, 1989). For example, McLaren et al. (2004) used it to measure the fit of supply chain management IS projects to the competitive strategy of the organisation.

Practitioners have also adopted various multi-criteria decision analyses combined with some utility function to evaluate the fitness of projects (PMI, 2008). Along the lines in practical settings, strategic fit is often measured in a questionnaire format by asking the personnel of the organisation how well the project aligns with the strategic objectives of the company on a Likert scale (Center for Business Practices, 2005). Talantsev & Sundgren (2013) took this approach a bit further and implemented a fuzzy linguistic logic to assign the final values for the fitness of a project based on a questionnaire, in which the respondents had to evaluate each project in contrast to the strategic goals of the organisation.

Once again two distinct approaches can be separated from each other: the ones that try to evaluate fitness quantitatively based on historical results (Meilich, 2006; Beynon, et al., 2010; Chen, et al., 2018; Venkatraman, 1989) and the ones that along the side with quantitative methods also utilise qualitative analysis in determining whether the projects fit the strategic objectives (PMI, 2008; Center for Business Practices, 2005; Talantsev & Sundgren, 2013; Fiss, 2011; Rahman & Rahman, 2019). A method that compares the projects against strategic objectives will be adopted in this thesis, because of the benefits attributed to the future-orientation as outlined in Section 2.2 and the practicality of the setting in this study.

2.2.2 Steps of Measuring Strategic Fit

Depending on what kind of data is available and how the strategic goals are phrased authors within the strategy discipline have contained slightly different steps and methods within the process of estimating strategic fit. The following chapters will attempt to unify the required steps in the evaluation of strategic fit to derive a robust framework for the strategic fit estimation process.

2.2.2.1 Step 1: List strategic goals

Regardless of the study and the guideline, all the relevant strategic goals have to be listed in a clear format, if they are to be used as the basis of strategic fit estimations. This step is included in one way or another in all of the studies that have the objective of defining strategic fit values through the goals (Fiss, 2011; Talantsev & Sundgren, 2013; PMI, 2008; Rahman & Rahman, 2019). A strategic goal can be defined as “a textual statement about a desired state or condition of the organisation” (Talantsev & Sundgren, 2013, p. 452). Often the goals can be naturally found from the organisation’s reporting material, or the top-management team can provide them (Talantsev & Sundgren, 2013; PMI, 2008).

2.2.2.2 Step 2: List project opportunities

While gathering the strategic goals, listing the project opportunities must likewise be one of the first actions to be done in the process. In this step all the relevant projects should be gathered and listed and, if possible, some can be already filtered out to make the size of the list more manageable. PMI (2008) for example, suggests that size or the urgency of a project opportunity could be used as preliminary filter in this step. While the project opportunities are being listed, their key descriptors, or the features that define them should also be recorded (PMI, 2008). These features could include qualitative variables such as name of the project, project type, description and documentation related to the project as well as quantitative attributes such as ROI, risk measures, size and resource requirements to name a few.

2.2.2.3 Step 3: Identify the required input measures based on strategic goals

After the goals and the projects have been listed, the next step is to evaluate what kind of an answer is required for each of the strategic goals. In practice, the goals can be divided into two different categories based on their specificity. The goals that are more high-level and require a qualitative assessment are often referred to as *"soft goals"* and the ones that are very specific and require a quantitative measure are called *"hard goals"* (PMI, 2008, p. 72; Talantsev & Sundgren, 2013).

For example, Talantsev & Sundgren (2013) only included soft strategic goals, which were very hard to quantify in absolute terms. One goal of the case company in their study about optimal development project portfolio selection was to "become experts in the fields of payroll and labour law". Whether one project opportunity takes the case company closer towards such a high-level, wide and multidimensional strategic goal is arguably impossible to evaluate with one quantitative measure. As there is no one truth whether or not a project opportunity aligns with such a strategic goal, they suggested to assess the fitness based on a group of evaluators' evaluations linguistically.

Hard strategic goals, however, should be measured in absolute terms. According to PMI measures that could be quantified with numerical values include for example the size and duration of the project, or the required resources for completing the project (PMI, 2008, p. 52). An example of a study with hard strategic goals could be e.g. Rahman and Rahman's (2019) paper about the strategic fit of garment unit's resources and capabilities. One strategic goal of the case company was to ensure "the availability of materials at the beginning of an order". The target was important for the factories as it had a direct impact on the efficiency of the manufacturing unit and the delivery times of the products. In this case, the strategic fit was evaluated based on how many times the strategic target was fulfilled whenever an order came in during the past months. As it was possible to quantitatively measure whether a unit could fulfil the goal, a numeric measure provided a much more accurate estimation for the fitness than a linguistic evaluation would have.

As hard and soft strategic goals differ in their requirements for input measures, the values have to be collected from different types of sources and the transformation into numeric strategic fit values will also be conducted differently.

2.2.2.4 Step 4: Transform input measures into strategic fit values

Hard goals and quantitative measures

Quantitative measures can act as a standalone definition of strategic fit, if the strategic goals are defined in a simple way. However, problems will arise in the aggregation of multiple quantitative measures into a single strategic fit value, if the ranges of values vary per measure.

To make the quantitative measures uniform, they should be rescaled with some transformation function.

Utility functions have been suggested as a simple way to transform the quantitative input measures that characterize certain strategic goals into strategic fit values (PMI, 2008). A utility function represents the preference of an individual for something (Encyclopedia, 2019). As strategic goals fundamentally embody the preferred direction the executive team wishes to take the company towards to, utility functions in this context can be thought as the company's preference for a project in contrast to its strategy. Going back to Rahman and Rahman's (2019) example about the strategic goal of ensuring "the availability of materials at the beginning of an order", it could be that if the materials are available less than 70% of the time, the company could risk going bankrupt due to inferior customer service. In such a case the 70% threshold value could receive the worst utility of 0 and from there onwards the utility could increase linearly or as some other function based on the times the materials were readily available. Note that the utility function has to be based on the organisation's own preference for projects and therefore varies case by case. Hints for the preferences can be found from the annual reports and the strategy materials as well as by interviewing the management team members and strategists in the company (PMI, 2008).

Soft goals and qualitative measures

Due to the vagueness, soft goals often demand for a qualitative evaluation to measure the fitness of a project. If the qualitative measures, which the projects are being compared against are not readily available in the key descriptors gathered in step 2, a group of evaluators has to be formed next for the evaluation of projects in relation to soft goals. The role of the evaluators is to assess each project in relation to each soft strategic goal. Often the correct group of evaluators can be found naturally from the organisation from for example the management team. Many different ways to evaluate projects in relation to soft goals has been proposed including a simple Likert-scale questionnaire (Center for Business Practices, 2005), a multi-criteria decision analysis (PMI, 2008) as well as fuzzy linguistic logic (Talantsev & Sundgren, 2013; Fiss, 2011). These different approaches vary in their method of transforming the judgements into numeric strategic fit values, but on a high-level each include the same steps that have to be executed.

After the group of evaluators has been decided, the second step will be to evaluate each project against each individual goal. When using a Likert-scale the evaluators will give numeric estimates about how well each project aligns with the strategic goals (Center for Business Practices, 2005). With multi-criteria decision analysis and fuzzy linguistic logic on the other hand, it is possible to assign qualitative evaluations for how well the projects align with the strategic goals. In general, each strategic goal should be evaluated with several questions to minimize the impact of misinterpretations and randomness, which will increase the validity of the evaluations (PMI, 2008, p. 58). After gathering all the responses, they will have to be aggregated and converted into quantitative measures with for example fuzzy logic like done in Talantsev & Sundgren's (2013) and Fiss's (2011) studies.

Note that gathering evaluations for a large sample of project opportunities can be very laborious and time consuming. Therefore, the method may often be invalid for the candidate project selection phase, which can include a large sample of individual project opportunities. Luckily, as the strategic goals of the case company in this thesis are hard goals, it was unnecessary to manually gather qualitative evaluations. Even though handling qualitative evaluations will not be covered in this thesis, a method for treating them is still outlined in Appendix 7.4 for interested readers.

2.2.2.5 Step 5: Aggregate the evaluations

After all the strategic goals have been compared against the project opportunities, the values should be aggregated into a single measure of strategic fit, which can conveniently be used to compare the projects against each other. Some studies have in the aggregation step assigned different weights for the importance of different strategic goals (PMI, 2008, p. 58; Rahman & Rahman, 2019), whereas some studies have assumed a similar importance for each goal (Talantsev & Sundgren, 2013). In PMI's multi-criteria scoring model (2008, p. 58), the values for different quantitative measures are first multiplied by the weight representing the importance of the strategic goal after which all of the values are simply summed up together. PMI also suggested to re-scale the final measures so that they lie between 0 and 1 for easier interpretability – zero representing the worst and one the perfect fitness. In Talantsev & Sundgren's (2013) study on the other-hand, the measures were already scaled on a 0-to-1 scale during the transformation of linguistic values into fuzzy numbers. In the final aggregation step

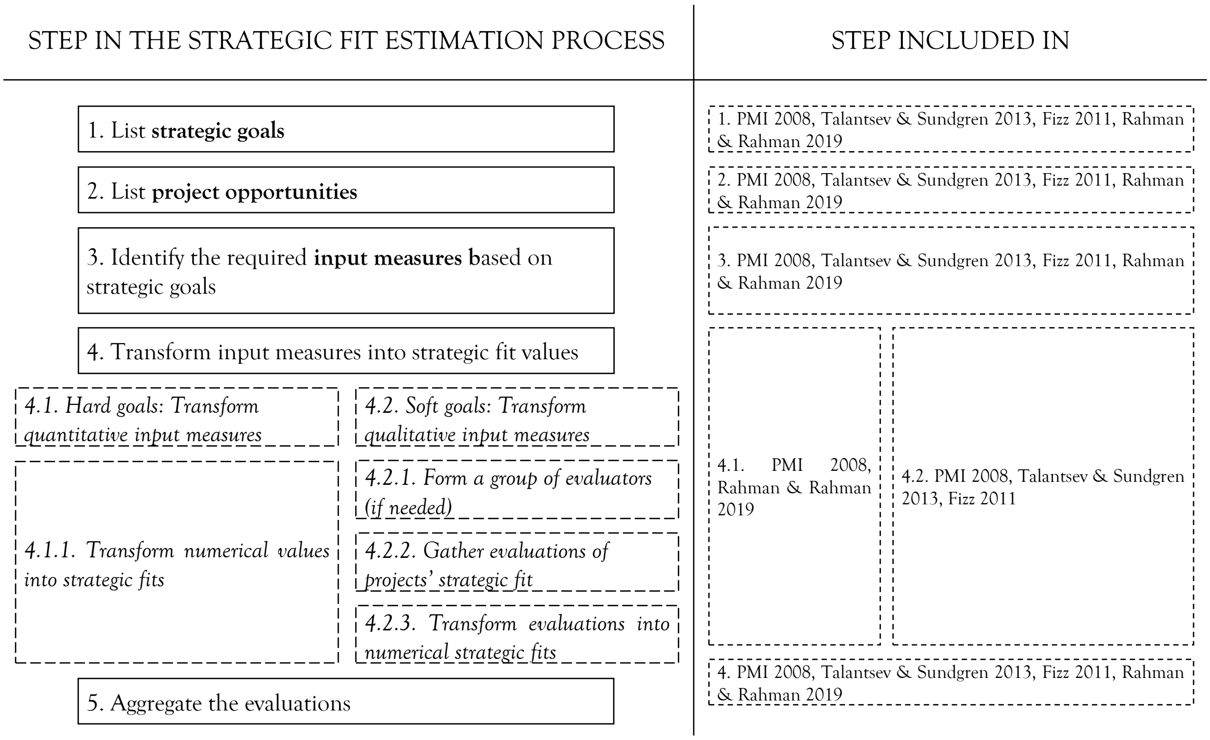

STEP IN THE STRATEGIC FIT ESTIMATION PROCESS
STEP INCLUDED IN
1. List strategic goals
2. List project opportunities
3. Identify the required input measures based on strategic goals
4. Transform input measures into strategic fit values
4.1. Hard goals: Transform quantitative input measures
4.2. Soft goals: Transform qualitative input measures
4.1.1. Transform numerical values into strategic fits
4.2.1. Form a group of evaluators (if needed)
4.2.2. Gather evaluations of projects' strategic fit
4.2.3. Transform evaluations into numerical strategic fits
5. Aggregate the evaluations
1. PMI 2008, Talantsev & Sundgren 2013, Fizz 2011, Rahman & Rahman 2019
2. PMI 2008, Talantsev & Sundgren 2013, Fizz 2011, Rahman & Rahman 2019
3. PMI 2008, Talantsev & Sundgren 2013, Fizz 2011, Rahman & Rahman 2019
4.1. PMI 2008, Rahman & Rahman 2019
4.2. PMI 2008, Talantsev & Sundgren 2013, Fizz 2011
4. PMI 2008, Talantsev & Sundgren 2013, Fizz 2011, Rahman & Rahman 2019

Figure 2: Compilation of methods used in the strategic fit estimation process.

the average of the values was then simply taken without taking into consideration any differences in importance between the strategic goals.

Figure 2 summarizes the strategic fit estimation process and the steps that it contains. A demonstration of how the project opportunities were evaluated against the strategic goals will be provided later in the Section 3: Demonstration of the thesis.

2.3 PROJECT PERFORMANCE

Many studies have examined the ways of measuring project portfolio performance. For project-oriented organisations like construction contractors, controlling the project portfolio for a performance measure that characterizes whether the project portfolio achieves its goals is of paramount importance, and the criteria used to measure success completely depends on the context of the organisation. Frame (2003, pp. 5-31) recognized the following five general criteria for evaluating and prioritizing different projects in a portfolio:

1. Financial criteria; including measures like NPV, payback-period, IRR, terminal value and benefit-cost ratio, ROI etc.
2. Technical criteria; including measures like analysing benefits for carrying out the project, ability to execute the project etc.

3. Production criteria; including measures like construction time, resource and equipment requirements, productivity, cost of quality, cycle time etc.
4. Risk-related criteria; including qualitative measures like complexity and contractual issues etc.
5. Human-resources criteria; including measures like number of personnel with experience of similar projects executed before etc.

Often engineers with proficiency in executing projects tend to focus on the more technical criteria in the previous list; namely technical and production related issues when considering whether to participate in a bidding competition or not (Frame, 2003, p. 5). Many authors have stressed the importance of profit-based criteria in project portfolio selection, but these common financial measures have also been criticised due to their assumptions and limited forecasting capability (Yescombe, 2002; Esty, 2003; Phillips & Phillips, 2006).

For the case company, it was important to derive a likelihood measure for winning a tendering competition. This was an issue as preparing tenders is costly and reserves a lot of resources from the valuable tendering organisation. Up until now, the candidate project selections have been made based on resource constraints and with some rough guidelines about, which projects are aligned with the segment's strategy. However as in many other organisations, no statistical methods have been utilised in the candidate project selection phase. Next, the literature review will cover some common statistical tools that can be used to predict tendering competition outcomes.

2.3.1 Predicting Performance Measures

Scholars and industry practitioners alike have started to develop and apply machine learning methods for predicting performance measures based on secondary data analysis (Rokach & Maimon, 2014). Large corporations gather massive storages full of data about various aspects of their business. Often the need to gather the data is based on some trivial aspect of doing business; you might have to know the mailing address of your client to send them invoices or change the status of a lead to closed won as a sign of the deal that you were able to close today. The primary purpose for the existence of such data is that it simply enables the company to run its daily operations. Hand (1998) defined knowledge discovery from databases (KDD) process as the secondary data analysis of large databases, in which the term "secondary" refers to the fact that the primary purpose of the data was not the data analysis in the first place.

Depending on the research problem at hand, predictive and/or descriptive methods can be utilised to solve a problem with pre-existing secondary data. Predictive data mining methods aim to understand the rules between a certain *target variable* (also called *dependent variable/attribute*) based on a set of *predictor variables* (also called *features* or *independent variables/attributes*). These methods are often referred to as supervised learning techniques as they require records of the target variable to conduct novel predictions. On the contrary, descriptive methods rather aim to understand the way the underlying data operates. Descriptive methods include for example unsupervised learning and visualisation methods that do not necessarily require any records of the target variable for data analysis (Rokach & Maimon, 2014).

One research project can contain elements from both methods. If the ultimate goal of the project would be to predict values of some target variable, but it seems that the prediction accuracy with the initial set of input variables is low, descriptive methods like clustering could be used to engineer new input attributes that can be included in the predictive model. Furthermore, it is common to deploy machine learning techniques for dimension reduction tasks before going into the main task itself, whether prediction or description (Kozachenko & Leonenko, 1987).

In addition to measuring the strategic fitness, the second aim of the study is to predict an outcome of a tendering competition based on project master data. The probability of winning a tendering competition will also be referred to as the *"likelihood of success"*. However, choosing the optimal algorithm for the task is not a trivial undertaking as all the algorithms have their unique characteristics and vary in performance based on the dataset.

2.3.2 Common Algorithms for Predictive Modelling

There is often a trade-off between interpretability and performance in predictive models. Naturally, simple models are easier for a business user to understand, but if they do not perform the task well enough, they are unusable. However, a well performing model that is hard to comprehend can be hard to trust as the user cannot see the reasoning behind the output. Several algorithms for predictive modelling have been developed including e.g. Decision Trees, Support Vector Machines, different Regression Techniques and Neural Networks to name a few (Quinlan, 1993; Hand, 1998; Rokach & Maimon, 2014).

The final result of deploying a predictive algorithm is a relationship structure referred to as the *model*. It explains how the predictor variables are related to the behaviour of the target variable. Therefore, a model is able to assign a label with a certain probability for the target variable with a specific set of predictor values.

Predictive models are divided between *Classification Models* and *Regression Models* depending on the target variable's data type. The difference between the two is that a classification model aims to classify, as its name suggests, a correct discrete value out of a predefined set of discrete classes to the target variable. A regression model, on the contrary, aims to map a continuous value to the target variable that must not necessarily be limited to belonging in a certain finite set. For example, predicting what will the temperature be on the day of company's midsummer party would demand for a regression model whereas predicting whether it will be sunny or rainy on that day would require a classification model.

2.3.3 Choosing the Algorithm: Decision Trees

In this study the aim is to determine whether the tendering competition will be won or lost. As the target variable has two possible outcomes, a classification model suits the purpose of this study. Decision trees are one option among the many machine learning algorithms that can be used for classification problems. They have multiple advantages including their ability to handle missing data and different data types, they can be visualised well and are all non-parametric (i.e. do not assume that the data would be distributed along any particular probability distribution or that the distribution would remain stable). Moreover, their *rule induction* ability makes decision trees very easily interpretable and as such quite intuitive for a business user in a practice-oriented setting, such as the context of this paper. (Quinlan, 1986)

Rule induction refers to the ability to formally extract a rule to represent a specific local pattern in the data or even the whole model. Therefore, a decision tree is simply a sequence of *if* antecedent *then* precedent rules that aim to group a dataset in a such a way that would make the groups as homogenous as possible (Quinlan, 1986). A typical decision tree can also be seen as an expert system as it can at least partially automate and suggest a course of action for its user based on the values of the input features. Even though decision trees can get mathematically quite complicated, their output is often self-explanatory and reasoning easy to follow, especially if the number of leaves and nodes can be kept in reasonable amounts. On the contrary, some other algorithms (e.g. neural networks) are often described as "black boxes" as it can be very hard, if not impossible for a human to follow their decision-making process. Implementation and adoption of such systems in the daily routines of management-level business personnel can be a hard task as their output may be hard to trust. (Rokach & Maimon, 2014)

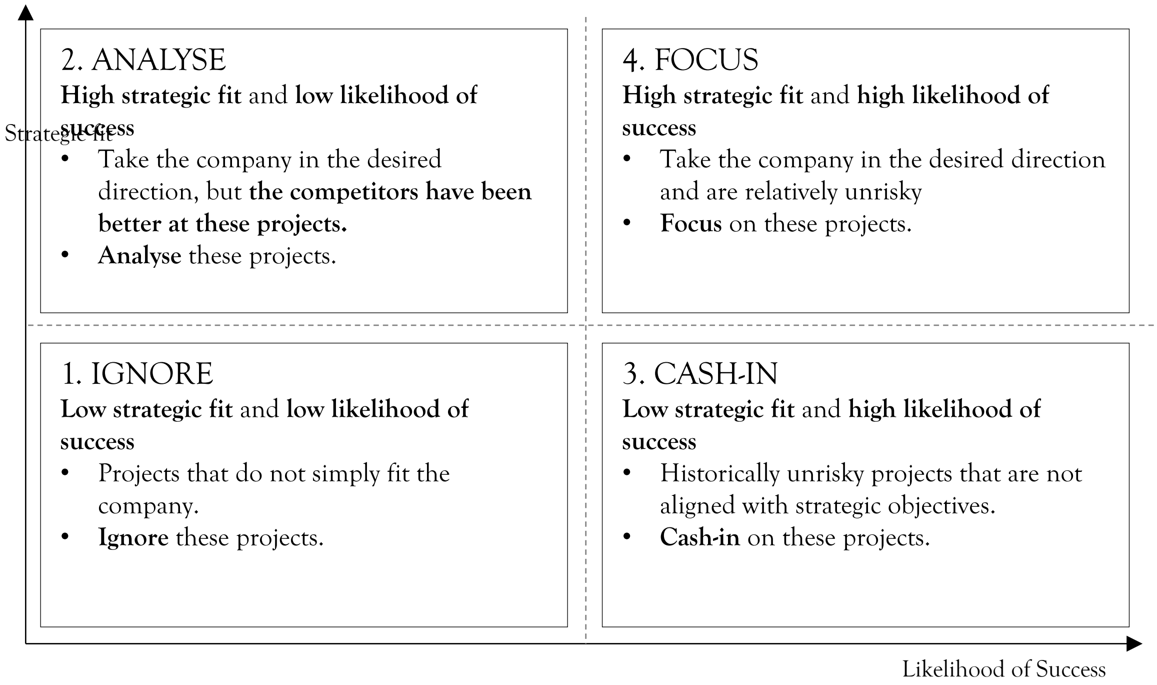
Decision trees have often been used in problems that include choosing the optimal opportunities to be pursued in the markets (Kass, 1980). Although, infrastructure construction companies do not have to actively perform direct marketing activities like many other companies, the tendering competitions are fundamentally very similar to these direct marketing campaigns. Not all opportunities in the markets can be pursued and due to the constrained resources only the best and most probable ones should be picked out. Decision trees can be effective in narrowing the large market down in a logical manner to the ones that are predicted as the most likely to be won.

On top of the above-mentioned benefits, decision trees also performed very well in the preliminary test for finding the most effective algorithm for the problem and therefore, were chosen as the machine learning algorithm to be used in this project. It must be noted though, that the final model will be as good as the predictive accuracy of the generated model. If the accuracy is very low, the output of the model cannot be trusted and therefore, either the accuracy should be increased, or another performance measure should be selected for predictions. Theory regarding decision tree induction, evaluation and selection are covered in depth in the Section 7.2: Performance Prediction with Decision Trees of the thesis.

2.4 DESIGN OF THE FRAMEWORK

As noted in Section 2.1: Project Portfolio Management, matrices have been often recommended and signified as an effective way in assisting in the project portfolio selections. While making the selections first of all, the projects should be selected by considering their alignment with the strategic objectives of the organisation. This will help to reduce the riskiness of the selections and keep the organisation on the desired strategic path as covered in 2.2: Strategy and Strategic Fit. Another way to reduce the riskiness of the selections is to predict the likelihood that the project will be successful in respect to a vital performance indicator. By utilising machine learning techniques, it is possible to alleviate this risk with a model that captures the capabilities of the organisation to execute a project successfully as described in 2.3: Project Performance. Quantifiable measures were derived for both, strategic alignment can be measured with *strategic fit* and project performance with *likelihood of success.*

The research objective was to improve project portfolio management practices by designing a framework, which incorporates both the likelihood of success and strategic objectives to help organisations optimise their project portfolios. By plotting *strategic fit* and *likelihood of success* in a matrix it possible to derive a simple framework for analysing projects with regards to both aspects. This matrix can be further segmented into a 2-by-2 matrix in order to guide the analysis (see Figure 3).



2. ANALYSE
High strategic fit and **low likelihood of success**
- Take the company in the desired direction, but **the competitors have been better at these projects.**
- **Analyse** these projects.

4. FOCUS
High strategic fit and **high likelihood of success**
- Take the company in the desired direction and are relatively unrisky
- **Focus** on these projects.

1. IGNORE
Low strategic fit and **low likelihood of success**
- Projects that do not simply fit the company.
- **Ignore** these projects.

3. CASH-IN
Low strategic fit and **high likelihood of success**
- Historically unrisky projects that are not aligned with strategic objectives.
- **Cash-in** on these projects.

Figure 3: Strategy –Success matrix.

Quartile 1. **Ignore: Low strategic fit and low likelihood of success.**

These projects do not fit the organisation's strategic objectives and have a low predicted success based on previous projects. Therefore, they should be ignored.

Quartile 2. **Analyse: High strategic fit and low likelihood of success.**

These projects are aligned with the strategic objectives of the organisation but have a low predicted success rate. Due to the mismatch, careful analysis should be made about these projects in order to figure out why the performance has been poor, or whether the actual issue lies in the definition of the strategic objectives itself. Therefore, these projects should be analysed.

Quartile 3. **Cash-in: Low strategic fit and high likelihood of success.**
These projects have historically been successful for the organisation but do not take the company towards the desired strategic direction. These projects can be considered as they are predicted to be quite safe, but still do not align with the strategic objectives. They can be used to generate revenue safely, if desired.

Quartile 4. **Focus: High strategic fit and high likelihood of success.**
These projects have historically been successful and are aligned with the strategic objectives. These projects should be the top priority of the organisation.

By analysing the positions that all the projects take in the matrix, it is possible to also judge whether the organisation's planned strategy and its capabilities align with the external environment. If for example majority of the projects would lie in the *Cash-in* quartile, it could be an indication that the organisation's strategic objectives are not actually aligned with what the capabilities of executing projects have been from the historical point of view. On the other-hand, if majority of the projects would lie in the *Analyse* quartile it would hint that the current strategy is defined in a way that fits the external environment, but the organisation has not been successful in such projects in the past. Finally, if many of the projects are in the *Focus* quartile, it would be a positive indication of two perspectives. First of all, the planned strategy would in such a case seem to fit the external environment well as there would be many project opportunities in the market that align with what the organisation wishes to execute. On top of this, the projects that are highly aligned with the planned strategy would also often tend to be successful and therefore quite unrisky for the organisation to execute. In such a situation the organisation would overall have a solid position in the market.

The case company's strategic position in the market will be analysed in a similar fashion in Section 4.1: Analysing Model's Performance Through the Matrix. Additionally, the performance of the tool will be analysed through a simulation study in Sections 4.2.1 and

4.2.2. First however, the matrix will be constructed, and the simulation will be prepared in the Section 3: Demonstration.

3 DEMONSTRATION

To increase the transparency, interpretability, rigidity and validity of the whole study and the matrix, the following sections will demonstrate how the matrix was built and tested in this study. The various design choices will also be explained in order to ease the evaluation of the results as well as to guide how similar models and simulations could be planned and executed in another context.

3.1 CONSTRUCTING THE MATRIX

Constructing the Strategy-Success Matrix can be done with the following three stages.

Stage 1: Prepare the data

Stage 2: Calculate the strategic fit values (See Sections 2.2 and 7.4)

Stage 3: Calculate the likelihood of success predictions (See Sections 2.3 and 7.2)

Next, the way these stages were conducted in this study will be described and the matrix will be assembled in the last section.

3.1.1 STAGE 1: PREPARE THE DATA

First, the data had to be prepared for calculating the strategic fit values and the likelihood of success predictions. The source system of data in this study was NCEC's customer relationship management (CRM) system Salesforce. The platform is used to track all the candidate project opportunities, various customer accounts and contacts as well as tasks and events related to the sales process overall. As a result, an extensive database has been accumulated in the CRM system, which accurately describes what kind of projects have been won and lost in the past.

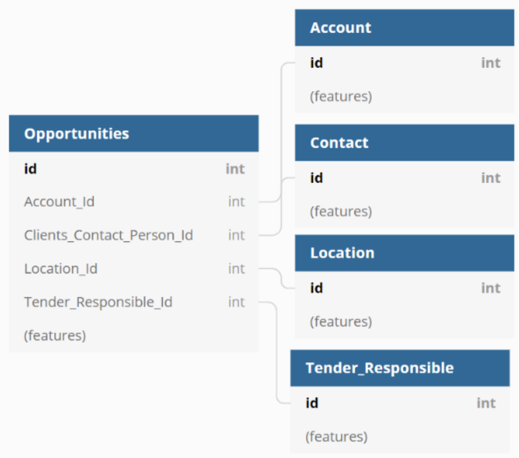


Figure 4: Database relationship diagram for forming the initial dataset. The bolded lines describe the primary keys of the tables. All the merges were formed using left joins.

To form as extensive and useful dataset for measuring the strategic fit and predicting the likelihood of succeeding in a tendering competition multiple tables were first joined together. Namely, *opportunities* , *account*, *contact*, *tender_responsible* and *location* tables were used, which describe the project opportunities, the customer accounts, external and internal contact persons and geographic locations related to the projects respectively. As there were some features in the initial data table that were poorly filled, the columns with a fill-rate of under 30% were dropped from the data set. Next, some feature engineering was conducted in order to transform dates into a more usable format for the modelling phase. Finally, the categorical values in the data set were transformed into one-hot-dummies as that will automatically take care of missing values and is the most suitable format for the classifier to be used later on in the data mining process. One-hot-dummy function essentially unpivots each class of each categorical attribute in its own column and includes a binary true or false statement as the value whether that category was associated to a specific record in the database. An example of a one-hot dummy transformation is demonstrated in Table 1.

Table 1: Example of one-hot dummy transformation.
The function converts all the categorical classes (on the left-hand side) into true and false statements (right-hand side).

Project	*Project Type*		*Project*	*Project Type = "Paving"*	*Project Type = "Road construction"*	*Project Type = "Foundation works"*
Project 1	Paving	→	Project 1	TRUE	FALSE	FALSE
Project 2	Road construction	→	Project 2	FALSE	TRUE	FALSE
Project 3	Paving	→	Project 3	TRUE	FALSE	FALSE
Project 4	Foundation works	→	Project 4	FALSE	FALSE	TRUE
Project 5	Null	→	Project 5	FALSE	FALSE	FALSE
Project 6	Paving	→	Project 6	TRUE	FALSE	FALSE

After these initial transformations were completed the data set was divided based on the *close date* of the projects into training, validation and test sets (also denoted as the *simulation set* or the *simulation period*). The experiment took place on February 2019, and it was decided that the goal was to simulate the project selections that have closed between 1st of July 2018 and 31st of January 2019. All of the projects that closed before 1st of February 2018 were used in the training set. Projects that closed between 1st of February 2018 and 31st of June 2018 were used as the validation set to select the best performing classification model out of the ones trained with the training data for the final likelihood of success predictions. And finally, as already described, the projects that closed between the period of 1st of July 2018 and 31st of January 2019 were used in the actual simulation to validate the performance of the matrix by comparing the selections with the previously covered hit rate and average strategic fit measures. Refer to Table 2 for the threshold values for all the data sets.

Table 2: Division of data into training, validation and test sets.

Data Set	Lower Limit	Upper Limit
Training set	no limit	31.1.2018
Validation set	1.2.2018	31.6.2018
Test set (also referred to as "simulation set")	1.7.2018	1.2.2019

As the one-hot-dummy transformation resulted in an extensive number of 5015 different features, only the 200 best features were selected with mutual information classifier method originally proposed in Kozachenko & Leonenko (1987) paper. The algorithm essentially measures the information that two variables share, in this case the dependent variable and all the independent variables, and filters out the features that have the lowest standalone predictive power. Only the training set was used to determine the best independent variables that were left as the final predictors after feature selection in order to avoid leakage of information from the validation and test sets. Finally, all the other projects than NCEC's infrastructure segment's projects were filtered out from the data set.

The original unfiltered and merged data set without the data set splits and date, one-hot-dummy and feature selection transformations included 7677 rows, in which each row represented a single past or upcoming project opportunity and 94 columns, which represented the various qualitiative and quantiative features of the projects. After the data transformations the train set $\mathbb{X}^{train}$ contained 1651 projects, validation set $\mathbb{X}^{val}$ 501 projects and test set $\mathbb{X}^{test}$ 473 projects. Every project is characterized with a feature vector $\boldsymbol{x}$, which contains 200 best features selected with the mutual information classifier and a label y characterizing the outcome of the tendering competition. A more comprehensive description of the notation is covered in the very beginning of the thesis as well as in appendix 7.1.

3.1.2 STAGE 2: CALCULATE THE STRATEGIC FIT VALUES

The methodology proposed in 2.2: Strategy and Strategic Fit gives the researcher the tools to evaluate projects against quantitative and qualitative strategic goals. The steps detailed in the section will be followed to evaluate the strategic fit values.

Step 1: List strategic goals

The management team of the case company has fortunately defined the strategic objectives quite clearly. As the largest construction company in Finland, and one of the largest in the Nordics, the case company has more overhead expenses than some of the smaller construction firms in the market. Consequence of these additional expenses is that smaller less-risky projects have historically delivered worse operational profit than larger and more complex projects. These larger projects are usually offered with a larger risk reservation, but with the size and capability advantages of the case company these risks can be mitigated. Therefore, the first strategic goal is defined as: *"Aim for larger projects"*. A second, goal can be derived from the KPIs that were used in the scorecards of the divisions. All of the divisions under infrastructure segment should *"minimize the number of projects with total value under 3.0 million euros in their portfolio"*. The two goals and their respective input variables are described in Table 3.

Table 3: Strategic goals of the case company.

Strategic Goal	Goal Type	Input Variable	Data Type
1: Aim for larger projects	Hard goal	Size of the Project	Continuous attribute
2: Minimize the number of projects with total value under 3 million euros in their portfolio	Hard goal	Size of the Project	Continuous attribute

Step 2: List project opportunities

The following step of listing the relevant project opportunities was largely conducted in the previous data preparation stage in Section 3.1.1. In summary, the project opportunities were imported from the case company's CRM system and merged with other tables that included additional information related to the client, people and location of the projects. As a preliminary filter, only the projects in which the segment of the case company was involved in were left in the data table. Regardless of the filtering, the data set still included 2625 projects in total, and 473 projects in the simulation data set. Conducting quantitative evaluations for such a large sample size of projects would be quite impractical for busy managers of the company. Fortunately, the strategic goals that were collected in the previous step are hard in nature and demand an input variable that can readily be found from the data set.

Step 3: Identify the required input measures based on strategic goals

In the following step, the input measures that will be required for measuring the strategic fit are identified and collected. The goals listed in the first step require one continuous input attribute *size of the project*, which can be found from the CRM system of the case company and was already collected in the data preparation stage. No other input attributes were necessary to gather.

Step 4: Transform input measures into strategic fit values

The strategic goals describe the preference of the case company when comparing different project opportunities. Therefore, they will also act as the basis for the utility function used to transform the input measures into the values of strategic fit. The first goal, *"aim for larger projects"*, indicates that larger projects should receive higher strategic fit values uniformly and that the highest project should receive the highest strategic fit value of one (1). The second goal, *"minimize the number of projects with total value under 3.0 million euros in the portfolio"*, gives a clear threshold under which the projects should belong more into the set of strategically unaligned projects and receive values below 0.5. Another possibility would be to directly assign the strategic fit value of zero (0) to each project that is below the threshold value, but because some of the smaller projects were still relevant for some business divisions of NCEC the former rule described the case company's preference in a more realistic manner.

However, the strategic goals do not give a clear indication about how the case company's utility increases and decreases depending on the values of the projects. Therefore, to simplify

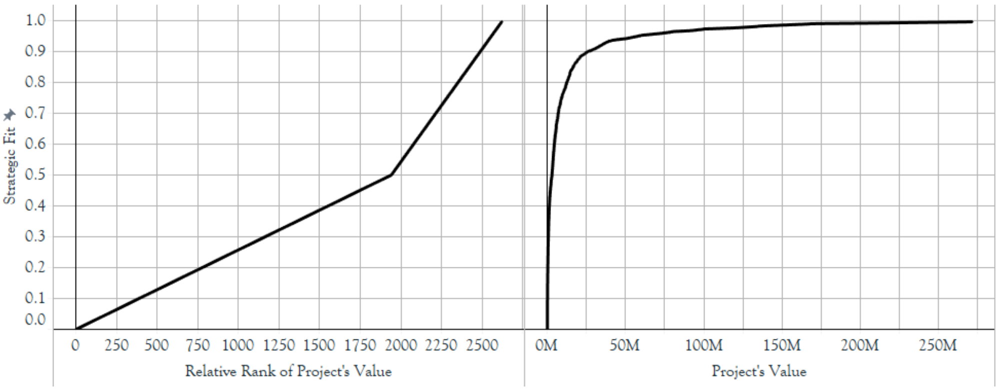


Figure 5: Graphical representation of the utility function to measure Strategic Fit.
Strategic fit values were derived based on the rank of the project's value (on the left). Plotting the strategic fit against the actual project values, however, conveniently visualizes what kind of project values correspond to each strategic fit value (on the right).

the analysis only the rank in terms of the projects' euro-based value matters in determining whether one project is more preferable than the other for the case company at any given time. Rank is also useful as one very large project could completely skew the distribution of the rest of the strategic fit values, if untreated projects' values were directly used in a linear utility function.

Based on the two goals, it is possible to determine the utility function for measuring strategic fit, which uniformly distributes the strategic fit values for projects with a project value of less than 3.0 million euros between [0, 0.5) and for projects above and equal to 3.0 million euros between [0.5, 1]. The left-hand side of Figure 5 visualizes the resulting utility function. To further analyse how the strategic fit estimates are distributed across the projects of different sizes, the curve on the right-hand side in Figure 5 plots the resulting strategic fit values against the absolute euro-based project sizes. It seems that the strategic fit values roughly follow some logarithmic function, which is based on the distribution of the projects' euro-based values.

The final step would include the aggregation of multiple utility functions for different goals into one strategic fit value. As only one input attribute and one utility function were necessary in this case, the last aggregation step did not have to be performed. The derived strategic fit values will be plotted on the y-axis of the Strategy-Success Matrix. In the next stage 3, the x-axis of the matrix will be constructed.

3.1.3 Stage 3: Calculate the Likelihood of Success Predictions

Predicting project performance is a common machine learning problem in the PPM field, which can be solved in many different ways as overviewed in Section 2.3. The objective in this study is to predict the likelihood of winning a tender offer based on the project master data generated through the everyday tendering activities. It is a classic secondary data analysis problem as the data is not gathered for the purpose of predicting tender competition outcomes originally. As the data preparation and splitting was already conducted in Section 3.1.1, the steps in the process of predicting the likelihood of success values are:

Step 1: Induce the decisions tree classifiers with different hyperparameter settings with the training set

Step 2: Select the best classifier based on the accuracies achieved with validation set

Step 3: Predict the likelihood of success values with the best classifier for the projects in the test set

Because the python script should be able to re-train itself reliably without constant superivison by a data scientist, the script was written so that it constantly compared different algorithms and hyperparameter settings against each other and then chose the best one out of them autonomously based on results gained from the validation set. The following sections will cover the steps taken to write the script. Also note that Section 7.2 covers all the methods that were used to produce and evaluate the likelihood of success predictions in this study.

Step 1: Induce the decision tree classifiers with different hyperparameter settings with the training set

In the first step, the best hyperparameter settings for each of the four algorithms were determined with scikit-learn's GridSearchCV function and the best classifier was selected out of the best performing algorithms with the validation set. The algorithms under experimentation were all variations of scikit-learn's DecisionTreeClassifier (DTC) algorithm. The hyperparameter settings were optimized for the precision score as that minimizes the number of false positives, which in this context are the tendered, but lost projects. Precision is also identical to the "hit rate" used by the case company to determine how many of the tendered projects (*predicted positive*) were actually won (*true positive*). Furthermore, precision score is not affected by the unequal distribution of positive and negative labels in the sample, which makes it a convenient measure for this specific case. Note that it is important to contrast the results to the distribution of the labels due to the imbalance when analysing any of the accuracies. See Table 4 for the distribution of labels in the data sets.

Table 4: Distribution of labels in the dataset

Set	*Samples*	*Positive*	*Negative*	*Share of Positive Labels*
train	1651	368	1283	22%
validation	501	146	355	29%
test	473	126	347	27%

The first and only non-ensemble method to be trialled was a simple decision tree classifier without any wrappers around it (described in Section 7.2: Performance Prediction with Decision Trees). The hyperparameters that were experimented with were the splitting criterions available, namely *gini impurity* and *entropy*, and the minimum number of samples in a leaf, which controls the generalizability of the classifier. In this case, gini impurity consistently performed better than entropy as the splitting criterion and the best classifier was found at around 35 minimum samples in a leaf node. Results of hyperparameter optimization for the first simple decision tree classifier are shown in Table 5.

Table 5: Hyperparameter optimization with GridSearchCV for decision tree classifier

rank	*validation precision*	*validation recall*	*validation accuracy*	*criterion*	*min samples in a leaf*
1	**0.5**	**0.364**	**0.75**	**gini**	**35**
2	0.479	0.329	0.743	gini	25
3	0.474	0.208	0.744	gini	40
4	0.465	0.231	0.741	entropy	35
5	0.459	0.295	0.737	gini	30
6	0.425	0.179	0.734	entropy	40
7	0.419	0.254	0.725	entropy	25
8	0.403	0.179	0.728	entropy	30

In order to boost the stability as well as the accuracy of the model, the decision tree classifier was next wrapped in a bootstrap aggregation algorithm (described in Section 7.2.3.1: Bagging). Experimented hyperparameters were *bootstrapping*, which controls whether the samples were drawn out with replacement or not, *maximum amount of samples in a tree* and *warm start*, which determines whether the algorithm utilizes the previously fitted classifier to save time in the tree induction phase. The base classifier to be used inside the bagging wrapper was the previously mentioned decision tree classifier with gini impurity as the splitting criterion as it seemed to perform quite well already without any ensemble methods. The best classifier was induced with bootstrapping and warm start turned on and with a cap of 15% of the samples taken in per tree. In total 100 estimators were induced per model to ensure the generalizability while keeping the time required to induce the trees reasonable. Table 6 describes the results for the bagging classifier.

Table 6: Hyperparameter optimization with GridSearchCV for bagging classifier

rank	*validation precision*	*validation recall*	*validation accuracy*	*bootstrap*	*max samples in a tree*	*warm start*
1	**0.513**	**0.335**	**0.754**	**TRUE**	**0.15**	**TRUE**
2	0.505	0.318	0.751	FALSE	0.1	FALSE
3	0.496	0.341	0.749	TRUE	0.2	TRUE
4	0.492	0.358	0.747	FALSE	0.2	FALSE
5	0.484	0.347	0.744	FALSE	0.15	TRUE
6	0.47	0.364	0.738	TRUE	0.15	FALSE
7	0.46	0.329	0.736	FALSE	0.2	TRUE
8	0.454	0.312	0.734	FALSE	0.15	FALSE
9	0.45	0.364	0.73	TRUE	0.1	TRUE
10	0.444	0.341	0.728	FALSE	0.1	TRUE
11	0.441	0.364	0.725	TRUE	0.2	FALSE
12	0.419	0.254	0.725	TRUE	0.1	FALSE

The third classifier to be experimented with was an adaptive boosting algorithm (described in Section 7.2.3.2: Boosting). Hyperparameters to be experimented with were the *learning rate*, which the adaptive boosting algorithm uses to decrease the contribution of subsequent classifiers and *algorithm*, which determines what kind of boosting algorithm AdaBoost uses. The base classifier again was the same decision tree classifier with gini impurity as the splitting criterion. Challenge with AdaBoost often is that it tends to overfit itself quite easily to the training set, which reduces the generalizability of the induced classifier. To counter that minimum samples in a leaf for the base classifier was set to a moderately high value of 35, which forces the induction to end much before the algorithm converges (as similarly suggested in previous research see e.g. (Zhang & Yu, 2005)). The best results were gained with learning rate set to 0.005 and by using the SAMME.R algorithm, which also tends to converge faster than the other alternative SAMME algorithm. The results of different variations of AdaBoost are shown in Table 7.

Table 7: Hyperparameter optimization with GridSearchCV for adaptive boosting classifier

rank	*validation precision*	*validation recall*	*validation accuracy*	*learning rate*	*algorithm*
1	**0.516**	**0.272**	**0.754**	**0.005**	**SAMME.R**
2	0.5	0.272	0.75	0.01	SAMME.R
3	0.442	0.393	0.724	0.001	SAMME.R
4	0.42	0.514	0.701	0.005	SAMME
5	0.42	0.497	0.702	0.01	SAMME
6	0.387	0.526	0.673	0.001	SAMME

The final algorithm to be trialled with was a random forest classifier. The number of estimators was set to 200 as the algorithm could handle the larger amount of trees faster than the other ensemble methods. Furthermore, a constant seed was set to the classifier so that the results can be repeated confidently multiple times. *Maximum number of features* and *maximum tree depth* limitations were the hyperparameters that were being altered. The first one injects more randomness to the induced trees by limiting the number of available features in the splitting phase whereas the second one controls the generalizability of the classifiers. The best precision was achieved with maximum features in a tree set at 20% of the total number of features and maximum tree depth limited at 20 nodes. Table 8 visualizes the results of the random forest experiments.

Table 8: Hyperparameter optimization with GridSearchCV for random forest classifier

rank	*validation precision*	*validation recall*	*validation accuracy*	*max features in a tree*	*max depth of the tree*
1	**0.518**	**0.254**	**0.754**	**0.2**	**20**
2	0.511	0.26	0.753	0.3	50
3	0.505	0.318	0.751	0.1	50
4	0.5	0.266	0.75	0.3	20
5	0.491	0.306	0.747	0.1	20
6	0.489	0.26	0.747	0.2	50
7	0.463	0.439	0.733	0.3	2
8	0.46	0.462	0.73	0.2	2
9	0.458	0.445	0.73	0.1	2
10	0.457	0.486	0.727	0.05	2
11	0.371	0.422	0.676	0.05	50
12	0.37	0.445	0.672	0.05	20

Step 2: Select the best classifier based on the accuracies achieved with validation set

Table 9 summarizes the results achieved by the best classifiers of each algorithm. The results show that all of the classifiers reached relatively similar precision, recall and accuracy levels after optimizing them with the GridSearchCV function. There seems to be a clear trade-off between the precision and recall of the models, which intuitively makes sense. It is easier to achieve a higher value of precision by limiting the positive predictions to the most certain ones in a sample, which has approximately 1-to-5 ratio of positive labels. On the other hand, by increasing the number of positive predictions it is more likely that a higher number of samples with a positive label belong into the set of positive predictions.

Table 9: Comparison of the best classifiers per algorithm

algorithm	*validation precision*	*validation recall*	*validation accuracy*
Decision tree	0.5	**0.364**	0.75
Bootstrap aggregation	0.513	0.335	**0.754**
Adaptive boosting	0.516	0.272	**0.754**
Random forest	**0.518**	0.254	**0.754**

As all the validation precisions for the best classifiers of each algorithm were practically identical (all within the range of ±0.09) other aspects than validation accuracy should also be taken into account when selecting the final classifier. If more stable results are preferred on the expense of increasing the complexity of the model, either the bootstrap aggregation or random forest algorithms should be chosen over the other two. Both algorithms increase the stability of the model by injecting randomness into the estimators in order to ameliorate the generalizability as covered in Section 7.2.3.1. On the other-hand, if simplicity and the rule induction ability of simple decision trees is preferred, the first algorithm without any ensemble methods should be selected. As the purpose of the Python script was to generate as accurate and generalizable results without data scientist's supervision, the bootstrap aggregation classifier will be used to predict the test set likelihoods for winning the tendering competitions. Note that the test set covers the same projects that will be used in the *simulation*.

Step 3: Predict the likelihood of success values with the best classifier for the test set

The bagging classifier generated quite satisfactory predictions with the test set as well. Altough precision and accuracy scores falled slightly, the recall score raised a bit. Considering that there were approximately 27% positive labels in the test set, a precision score of 0.428 is satisfactory. Table 10 shows the test set accuracies.

Table 10: Results with the test set.

algorithm	*test precision*	*test recall*	*test accuracy*	AUC
Bootstrap aggregation	0.428	0.358	0.696	0.65

While it is certainly not a perfect result, it shows that there are some underlying patterns in the data that give an indication about, which kinds of projects the case company tends to win. It also validates the assumption that a machine learning model can assist the case company in skimming through the projects in the market and as such is a valid tool in the candidate project selection phase. A confusion matrix about the predictions achieved with the test set is visualized in Table 11.

Table 11: Confusion matrix of the test predictions.

	predicted positive	*predicted negative*
actual positive	62	111
actual negative	83	383

Finally, the individual feature importances per independent variable were inspected to flesh out how the probability estimates were derived by the bagging classifier. The feature importance measure characterizes the individual contribution of each feature so that higher values mean a more significant role in the predictions and the sum of all importances equals one (Pedregosa, et al., 2011). It seems that the floating numbers and integers were far superior in their predictive power when compared to the boolean predictors. Factors that describe the complexity and scale of the projects, namely *Project value* and *Construction duration in days* along with dates representing the recency of the projects were the most important predictors. Also, features that described the client relationship like *NPS* and *Number of open opportunities for the* *account* along with factors characterizing which (geographical) divison of NCEC was tendering the project, namely *Division*, *Country* and *Region* features also played a part in the predictions. Table 12 shows the 20 most important features for the final chosen bagging classifier.

Table 12: Twenty most important features for the final predictor.
It seems that floating numbers and integers were far superior in their predictive power than the boolean features.

Name of the Feature	*Type*	*Importance*	*Rank*
Project value	Floating	0.2241	1
Closing date	Integer	0.1777	2
Submission date (date difference to today)	Integer	0.1006	3
Construction date (date difference to today)	Integer	0.0884	4
Construction duration in days	Integer	0.0777	5
Closing date (date difference to today)	Integer	0.0735	6
Construction date (date difference to today)	Integer	0.0529	7
Construction duration in months	Integer	0.0452	8
Number of open opportunities for the account	Integer	0.0403	9
NPS score of the account	Integer	0.0225	10
Region = "Uusimaa"	Boolean	0.0206	11
Division = "NRFE"	Boolean	0.0170	12
Country = "Estonia"	Boolean	0.0155	13
NPS = [9, 10]	Boolean	0.0143	14
Project type = "Other"	Boolean	0.0138	15
Region = "Ida-Virumaa"	Boolean	0.0059	16
Account = "Tallinna Kommunaalamet"	Boolean	0.0036	17
Country = "Latvia"	Boolean	0.0028	18
Project type = "Other infrastructure construction"	Boolean	0.0019	19
Division = "NSWE"	Boolean	0.0012	20

The output of the bagging classifier gives each project opportunity a crisp classification describing whether the project is predicted to be won or lost as well as a probabilistic estimation describing how certain the model is about the prediction. The probabilistic value is especially important as it will be used as the x-axis in the Strategy-Success Matrix. Now as both of the measures for the two axes have been constructed in stages 2 and 3 respectively the Strategy-Success Matrix can be assembled.

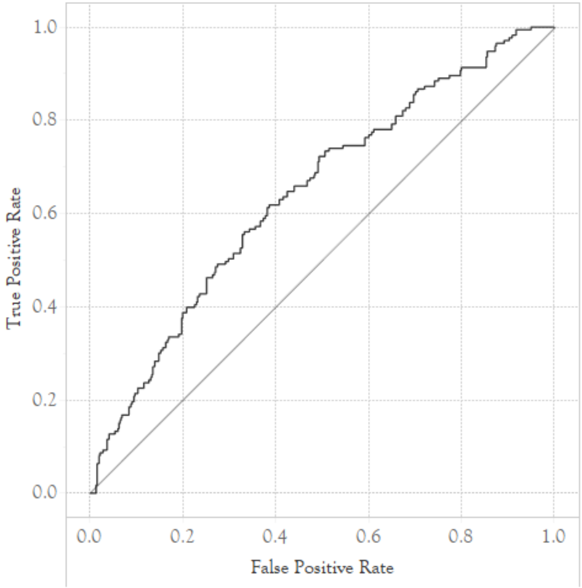


Figure 6: ROC-curve for the bagging algorithm with test data.

3.1.4 Assembling the Strategy-Success Matrix

To bring the matrix into life, the final end-user interface was implemented in Tableau business intelligence software. Note that any visualization tool could be used to create the matrix as the design is quite simple. The following section will cover the suggested components of the dashboard, which includes two linked visualizations, one that summarizes the contents of the four quartiles and one which displays all the projects along the two axes (the left and right-hand sides of Figure 8 respectively).

As noted before, if the likelihood of success predictions are very inaccurate the matrix can easily become unusable. Fortunately, there is no such problem with the strategic fit measure as the evaluations are always subjective and based on factual data, not predictions. To measure the accuracies of the predictions, a performance metric should be chosen based on the target variable that is being used to characterize likelihood of success.

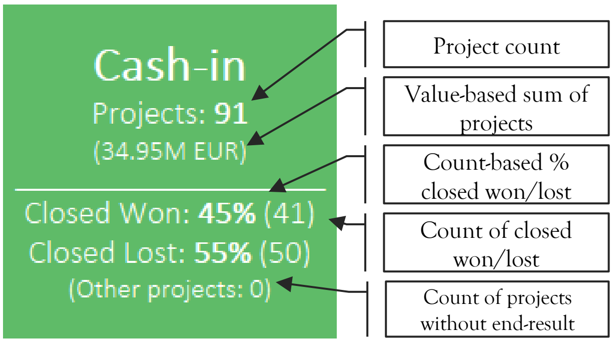
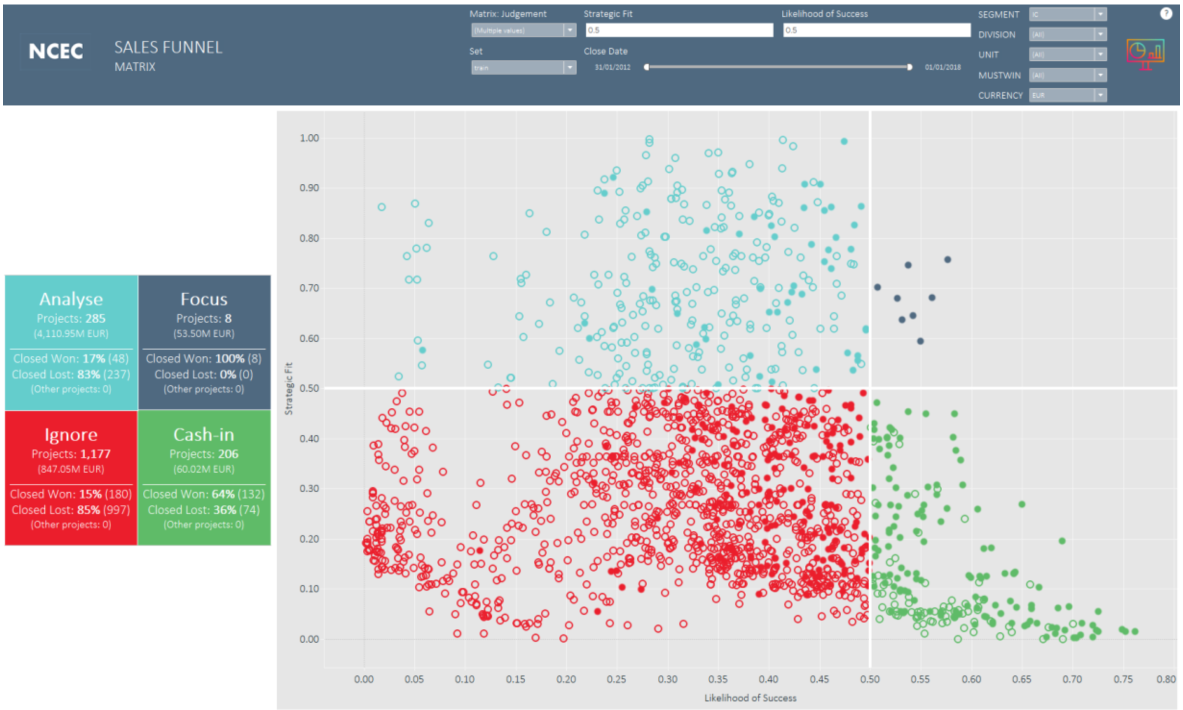


Figure 7: Description of matrix summary visualization

As in this study the decision tree model is classifying projects based on whether it is likely that they are won or lost, the percentage of projects in the *Closed Won* and *Closed Lost* categories was selected as a natural performance measure within the quartiles. It is calculated for each of the four quartiles as demonstrated in Figure 7. In the two high likelihood of success quartiles, *Cash-in* and *Focus*, a higher Closed Won rate indicates more accurate predictions. On the contrary, in the two low likelihood of success quartiles, *Ignore* and *Analysis*, a higher Closed Lost rate indicates a better performance of the predictions. The summary visualization on the left-hand side in Figure 8 is used to evaluate these accuracies. In the example in Figure 8, the predictions for the *Analyse*, *Ignore* and *Focus* quartiles are very accurate as the certainty of the predictions lies between 0.83 to 1.00. The *cash-in* quartile has a slightly lower accuracy, but can still be considered sufficient as it beats a random classifier with a considerable margin. These accuracies are obtained based on the projects that have already closed and therefore the outcome of the tendering competition is already known.

In order to carry out appropriate project selections, the dashboard should be used in the following manner. First, the user should use historical data to determine the approximate accuracies of predictions in each of the four quartiles. If the rates in each of the quartiles are almost equal, the likelihood of success predictions are not accurate and the projects' positions on the likelihood of success axis has little significance. To make the matrix flexible, the user should also be allowed to alternate the threshold points to determine when the framework classifies the projects in each quartile. For example, by increasing the likelihood of success threshold, the accuracy of the predictions in the *analyse* and *focus* quartiles can be improved as the framework requires a higher likelihood of success estimation to classify the projects positively. However, it will also decrease the accuracy in the other two segments as projects with relatively high likelihood of success values will still be classified negatively.

After the thresholds have been set, the user should only filter the open projects that have not yet closed into the visualization. Keeping in mind the accuracies of each of the quartiles, the user can then skim through the most promising projects and by incorporating her expert judgement, carry out the appropriate project selections. The matrix should therefore be used as a descriptive expert system, which assigns recommendations for selecting certain projects.



NCEC SALES FUNNEL MATRIX

Matrix: Judgement (Multiple values)

Set: train

Strategic Fit: 0.5

Likelihood of Success: 0.5

Close Date: 31/01/2012 – 01/01/2018

SEGMENT: IC

DIVISION: (All)

UNIT: (All)

MUSTWIN: (All)

CURRENCY: EUR

Analyse Projects: **285** (4,110.95M EUR) Closed Won: **17%** (48) Closed Lost: **83%** (237) (Other projects: 0)	**Focus** Projects: **8** (53.50M EUR) Closed Won: **100%** (8) Closed Lost: **0%** (0) (Other projects: 0)
Ignore Projects: **1,177** (847.05M EUR) Closed Won: **15%** (180) Closed Lost: **85%** (997) (Other projects: 0)	**Cash-in** Projects: **206** (60.02M EUR) Closed Won: **64%** (132) Closed Lost: **36%** (74) (Other projects: 0)

Figure 8: Strategy - Success Matrix in Tableau.
Projects that have a large solid fill were won and the ones with an empty fill lost respectively.

3.2 Validating the Performance of the Matrix

In addition to using the matrix as a simple visualization tool to skim through the project opportunities as described in Section 3.1: Constructing the Matrix, it is possible to also make autonomous decision with the framework by calculating a score measure to prioritize projects. As such a simulation study shall be conducted, it is also important to evaluate the scoring model's selections appropriately. The following section will first cover how the performance of the matrix can be validated through the scoring model and with what metrics it should be evaluated.

3.2.1 Evaluation of the Matrix

Evaluating the matrix's performance appropriately is one of, if not the most crucial task in conducting rigorous design science research. To select the appropriate evaluation method, this thesis followed Venable et al.'s (2012) DSR evaluation matrix detailed in Figure 9. The study divided the appropriate DSR evaluation methods according to the environment in which, the designed framework should be tested (naturalistic vs. artificial) and time frame when the evaluation should happen (ex-ante vs. ex-post).

	EX-ANTE	*EX-POST*
NATURALISTIC	• Action research • Focus group	• Action research • Case study • Focus group • Participant observation • Ethnography • Phenomenology • Survey
ARTIFICIAL	• Mathematical or logical Proof • Criteria-based evaluation • Lab experiment • Computer simulation	• Mathematical or logical proof • Lab experiment • Role-playing simulation • Computer simulation • Field experiment

Figure 9: DSR Evaluation matrix. Adopted from Venable et al. 2012

In this study, the evaluand is a framework that consists of two distinct measures. The first one, strategic fit, models the strategic alignment between projects and the company's strategic goals. The second one, likelihood of success, models the predicted outcomes of projects with regards to the chosen performance measure. As it consists of two separate mathematical models, the framework in this study is rather a product than a process. It is not purely technical in a sense that a human will in most cases be the final decision maker determining whether to enter the tendering competition or not. However, its performance is not related to its user at all. Rather from the performance point of view, its ability to make candidate project selections can be tested as a standalone product without a human user as its able to do the selections autonomously.

First it is important to establish that the matrix is the one causing the observed improvement, if there even will be any. Therefore, as clinical testing environment as possible should be preferred in this first stage due to the novel nature of the framework. After the design has been proved to be valid the matrix could be evaluated as an instantiation in a real world setting in order to flesh out the more subtle socio-technical issues related to using the design. However, the first priority is to ensure its statistical performance and only after then its acceptance in an organisational setting. In the scope of this thesis therefore is the first step covering the statistical validation of the matrix whereas the second stage concerning the user acceptance validation will be left-out for future studies. Thus, based on the previous analysis an artificial - ex-ante setting will be the most appropriate testing environment for this design science research project. The methods for testing a framework in such an environment can be seen from the lower-left hand-side in Figure 9.

3.2.2 Selection of the Appropriate Evaluation Method

Out of the possible evaluation methods, criteria-based evaluation was determined as the most appropriate for this research. As a benchmark it is possible to use the historical results that the case company's management team was able to deliver without having the matrix. This data can be found in the form of historical records of decisions to participate in a tendering competition. It is then possible to simulate a certain time period as if the selections were purely made according to the suggestions of Strategy-Success Matrix. Finally, the actual selections of the case company and the selections of the framework can then be compared against each other.

There should be two distinct measures, one for both of the axes, that determine the performance difference between the model's and the company's selections. A viable way to measure the performance would be to use a performance metric already used by the case company or select another simple measure. Strategic fit is a novel concept within the case company and consequently they do not have an established way to measure it. Therefore, average strategic fit of the portfolio will be used as a simple way to measure the strategic alignment of the whole portfolio. For measuring the performance of candidate project selections though, the case company is already using a measure called hit rate which captures its likelihood of succeeding in a tendering competition. These two measures will be further elaborated next.

3.2.2.1 *Evaluation Metric 1: Strategic Fit*

To determine how close the project portfolio is to the ideal portfolio from the strategic fit perspective, the average strategic fit of projects selected by the scoring model will be compared against the portfolio selections done by the management team. If the model's portfolio has a higher average strategic fit than the management team's portfolio it can be concluded to follow the strategic objectives more accurately than the management team and therefore, fulfil its purpose. Also, by calculating the average strategic fit for the management team's portfolio it is possible to roughly evaluate how well the organisation has been following the planned strategy of the executive team in its candidate project selections.

3.2.2.2 *Evaluation Metric 2: Hit Rate*

Hit rate in this study corresponds to the percentage of closed won projects from all the tenders, which were submitted. This is a common measure for both, the business users of the case company as it is used to measure the performance of the business divisions, but also for machine learning specialists to measure the accuracy of the model. In the machine learning context, it is often called *precision* and it corresponds to the true positive rate of the model

$$Precision = \frac{Count\ of\ True\ Positives}{Count\ of\ True\ Positives + Count\ of\ False\ Positives}. \qquad (3.1)$$

An optimal solution would be that the model would avoid picking any tenders that were actually lost and therefore gain a hit rate (precision) of 1.00. As it is highly likely that the model will select some projects that were not actually tendered at all, those will be excluded from the hit rate calculations as the outcome of the tendering competition is simply unknown had the company taken part in the tendering competition. These projects are still interesting though and should not be excluded from the simulation data as they can give a valuable direction of where the potential opportunities would have been for the case company. The model can be considered successful, if its portfolio has a higher hit rate than the actual tendering portfolio chosen by the management team of the case company.

3.3 Constructing the Simulation

On top of the three stages covered in Section 3.1 there is one additional stage to construct the simulation study, which can be used to validate the performance of the matrix.

Stage 4: Calculate the scores and make the selections

Even though stage 4 is not necessary to construct the matrix, it provides a useful way to prioritize the projects in a consistent way. Therefore, calculating the scores for the individual projects can be very helpful for any user that wishes to use the matrix.

3.3.1 Stage 4: Calculate the Scores and Make the Project Selections

PMI (2008, p. 58) suggested a scoring model for prioritizing different project opportunities. Their scoring model consists of a summation of individual evaluation criteria multiplied by the relative weight of each criterion. Along these lines in this thesis the prioritization of the project opportunities in the simulation will be conducted with a similar scoring function

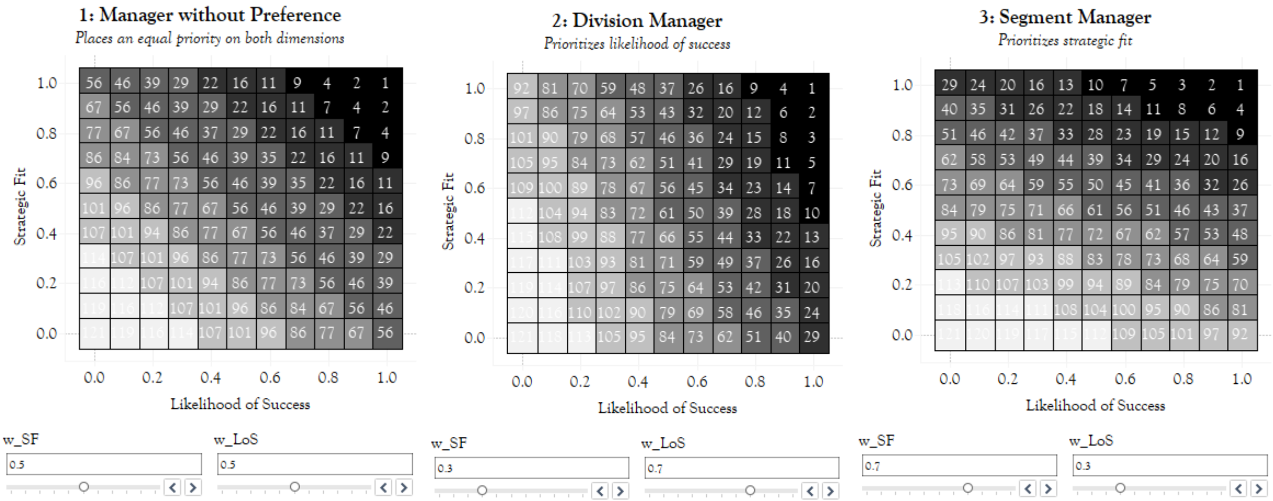
$$Score(\boldsymbol{x}) = w_{SF} \times SF(\boldsymbol{x}) + w_{LoS} \times LoS(\boldsymbol{x}) \tag{3.2}$$

in which,

1. $SF(\boldsymbol{x})$: Strategic fit of a project with a feature vector $\boldsymbol{x}$
2. $LoS(\boldsymbol{x})$: Likelihood of success of a project with a feature vector $\boldsymbol{x}$
3. w_{SF}: The weight of strategic fit in the simulation
4. w_{LoS}: The weight of likelihood of success in the simulation.

Weights are multiplied with the strategic fit and likelihood of success measures in order to give the managers freedom in deciding, which factors to emphasize more. The weights can be used to modify the project selection order as demonstrated in Figure 10. Note that the problem at hand is not an optimization problem, per se, as the goal is not to maximize for the sum of scores, but instead just simulate the selections that a naïve user might make by selecting the projects with the highest scores. The weights are used to characterize the different utilities and priorities that different users may have. For example, a top-level manager might be more concerned about selecting projects that align with the high-level strategic objectives ($w_{SF} > w_{LoS}$) whereas a line-manager prefers to secure future orderbook and thus, select the projects that are the safest and most likely to be won ($w_{LoS} > w_{SF}$).

Note that there is a direct relationship between increasing the weight of one measure to increasing its respective evaluation metric as well in the simulation. In practice, if for example a higher likelihood of success weight is used in the simulation the scoring model will select project opportunities with higher probabilities to win and, therefore is likely to end up with a higher hit rate while disregarding the average strategic fit of the portfolio. Different weights and their impact to the evaluation metrics will be explored in the Section 4: Managerial Implications.



1: Manager without Preference

Places an equal priority on both dimensions

56	46	39	29	22	16	11	9	4	2	1
67	56	46	39	29	22	16	11	7	4	2
77	67	56	46	37	29	22	16	11	7	4
86	84	73	56	46	39	35	22	16	11	9
96	86	77	73	56	46	39	35	22	16	11
101	96	86	77	67	56	46	39	29	22	16
107	101	94	86	77	67	56	46	37	29	22
114	107	101	96	86	77	73	56	46	39	29
116	112	107	101	94	86	77	73	56	46	39
119	116	112	107	101	96	86	84	67	56	46
121	119	116	114	107	101	96	86	77	67	56

w_SF 0.5

w_LoS 0.5

2: Division Manager

Prioritizes likelihood of success

92	81	70	59	48	37	26	16	9	4	1
97	86	75	64	53	43	32	20	12	6	2
101	90	79	68	57	46	36	24	15	8	3
105	95	84	73	62	51	41	29	19	11	5
109	100	89	78	67	56	45	34	23	14	7
112	104	94	83	72	61	50	39	28	18	10
115	108	99	88	77	66	55	44	33	22	13
117	111	103	93	81	71	59	49	37	26	16
119	114	107	97	86	75	64	53	42	31	20
120	116	110	102	90	79	69	58	46	35	24
121	118	113	105	95	84	73	62	51	40	29

w_SF 0.3

w_LoS 0.7

3: Segment Manager

Prioritizes strategic fit

29	24	20	16	13	10	7	5	3	2	1
40	35	31	26	22	18	14	11	8	6	4
51	46	42	37	33	28	23	19	15	12	9
62	58	53	49	44	39	34	29	24	20	16
73	69	64	59	55	50	45	41	36	32	26
84	79	75	71	66	61	56	51	46	43	37
95	90	86	81	77	72	67	62	57	53	48
105	102	97	93	88	83	78	73	68	64	59
113	110	107	103	99	94	89	84	79	75	70
118	116	114	111	108	104	100	95	90	86	81
121	120	119	117	115	112	109	105	101	97	92

w_SF 0.7

w_LoS 0.3

Figure 10: The effect of weights on project selections.
Numbers represent the order in which the model would select the projects in the Strategy – Success Matrix.

It makes sense to set some restrictions with regards to how many projects the scoring model is allowed to select. The infrastructure construction segment of the case company is divided into six business divisions that have their own resources for preparing and calculating tender offers. It can roughly be estimated that the sum of tender offers measured in monetary terms corresponds to the resources each division have at their disposal to prepare tenders. Therefore, to limit the number of project selections made by the model, the sum of project value per business division of projects selected by the model should always be less or equal than the sum of project value per business division of projects tendered during the time period. Hence, the model is forced to select projects from different business areas in a similar manner as was tendered during the specific time period. This also makes the selections comparable and realistic from the tendering organisation's resource usage point of view. With the constraint in place, if a performance of one business division would be superior to that of an another the model cannot simply just suggest not selecting any projects for that division. The following steps summarize how the Strategy – Success Matrix can be constructed and how the project selections can be simulated.

Step 1: Prepare the data and calculate the strategic fit and likelihood of success values for all the project opportunities according to stages 1,2 and 3

Step 2: Set the weights w_{SF} and w_{LoS} according to the user's preferences

Step 3: Calculate the scores according to equation (3.2) and rank the projects in a descending order per division

Step 4: Select projects starting from the highest ranked project while the sum of project value per division is equal or less than the actual sum of projects tendered per division.

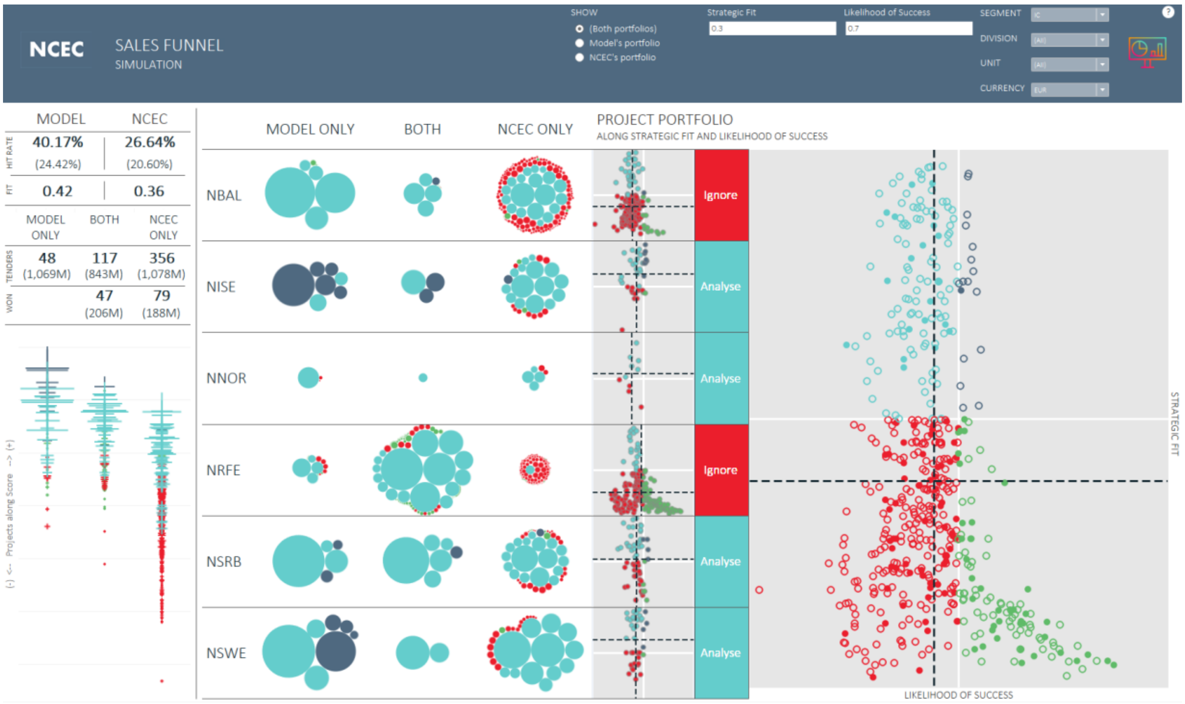
Now that the selections for the simulations can be made, it will help the analysis if the selections can be visualized in a convenient way. The following section will elaborate on how the simulation was analysed in this study.

3.3.2 ASSEMBLING THE SIMULATION

Along-side the *Matrix* dashboard covered in Section 3.1.4, *Simulation* dashboard was created in Tableau as well to validate the performance of the matrix. This was done by comparing the model's suggested selections to the real project portfolio selections made by NCEC during the test period covering the 6 months starting from 1st of July 2018 and ending to 31st of January 2019. The following chapter will briefly cover the different visualization elements in the dashboard as the latter part of the following chapter 4: Managerial Implications will then dig deeper into the results of the simulation to analyse its implications for the case company.

The simulation is based on the scoring system, which uses the score measure to first rank and then select the best projects out from the market. By using the output from the Python implementation, Tableau is able to calculate the scores based on the likelihood of success and strategic fit values by multiplying them with the respective weights that can be dynamically set within the platform. By altering the weights of strategic fit and likelihood of success measures, it is possible to simulate how different users might make decisions with the matrix.

Figure 11 shows the *Simulation* dashboard in Tableau. The measures on the left-hand side can be used to summarize and compare the hit rates and strategic fit values of model's and NCEC's project portfolios respectively. The matrices on the right-hand side visualize the positions of the project selections on the Strategy-Success Matrix. The dots in the matrices represent the various project selections similarly as in the first *Matrix* dashboard covered in Section 3.1.4: Assembling the Strategy-Success Matrix. By averaging the project selections, the dashed lines indicate the position of the whole portfolio on the same matrix. In the dashboard this aggregation is done on the division-level (the smaller matrices in the middle of the dashboard) and on the segment-level (the larger matrix on the right-hand side of the dashboard). Based on the intersection of the dashed lines in the matrices, the portfolios per division can be labelled as *ignore*, *analyse*, *cash-in* or *focus* as seen on the right-hand side next to the smaller division-level matrices in Figure 11. The bubble chart in the middle visualizes what were the differences and similarities of the candidate project selections made by NCEC's management and the model. The ones that neither selected are filtered out of the visualization. Only projects belonging in the test set are shown in the visualisation.



SHOW
○ (Both portfolios)
● Model's portfolio
● NCEC's portfolio

Strategic Fit: 0.3
Likelihood of Success: 0.7
SEGMENT: IC
DIVISION: (All)
UNIT: (All)
CURRENCY: EUR

	MODEL	NCEC
HIT RATE	40.17% (24.42%)	26.64% (20.60%)
FIT	0.42	0.36

	MODEL ONLY	BOTH	NCEC ONLY
TENDERS	48 (1,069M)	117 (843M)	356 (1,078M)
WON		47 (206M)	79 (188M)

Figure 11: Simulation dashboard in Tableau.

The performance of the model's selections are evaluated on the left-hand side of the dashboard. As demonstrated in Figure 12, the selections made by the model and NCEC are evaluated based on the count-based $\left(\frac{\#\ of\ projects\ won}{\#\ of\ projects\ selected}\right)$ and value-based hit rates $\left(\frac{EUR\ sum\ of\ projects\ won}{EUR\ sum\ of\ projects\ selected}\right)$. The average strategic fit of the selections are also compared against each other.

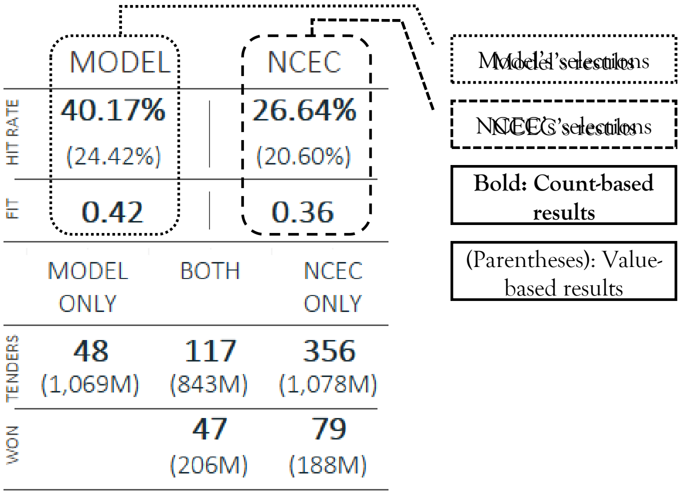


Figure 12: Results of the simulation.
The selections are evaluated mainly based on count-based hit rate and strategic fit. The value-based values are also shown in the parentheses. In this example, the model performed slightly better than the case company did when measured with count and value-based hit rates. The model also selected much more accurately aligned projects to its portfolio than the management indicated by the fitness value.

As there are no information about the end-result regarding the projects that were only selected by the model, the hit rates have to be calculated on the basis of mutually selected projects (refer to Figure 13). Therefore, NCEC's hit rate functions as a baseline, which would be achieved by a random classifier due to a chance. If the model's hit rate surpasses NCEC's hit rate, it is an indication that there is an underlying pattern that can be utilized to predict the probability to win a tendering competition to a degree. Because the model is forced to select a set of projects with the same total sum in value as NCEC selected, the only way the model can achieve a better hit rate than NCEC is to select the projects that were actually won by NCEC, avoid lost projects and with the remaining resources select new projects from the project opportunity space that were not selected by NCEC's management. As demonstrated in Section 3.1.3, the classifier surpassed the accuracy of a random estimator (AUC > 0.5), and thus similar results will be expected in the simulation as well.

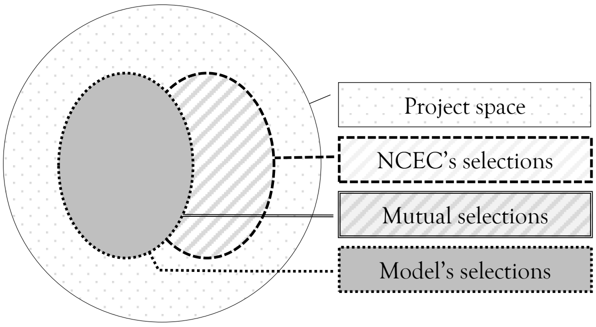


Figure 13: The project space.
All the project opportunities either belong into the set where neither NCEC nor the model selected the project, either one did, or both selected it. NCEC's hit rate is calculated from the whole set what was selected by NCEC. In contrast, model's hit rate can only be calculated from the set where both NCEC and the model selected the project, because the outcome of the tendering competition is unknown in the ones where NCEC did not submit the tender.

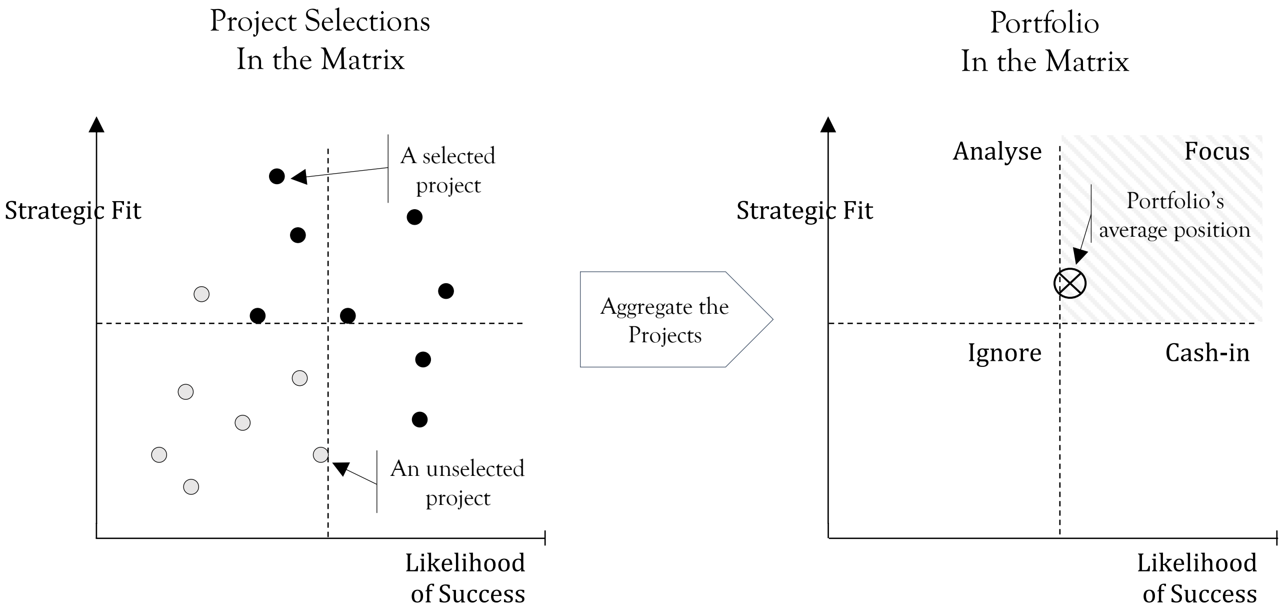


Figure 14: Candidate project selections and the portfolio's position.
By taking the averages of strategic fit and likelihood of success measures over the entire set of selected projects, it is possible to evaluate the hypothetical average position of the portfolio in the matrix. This can be aggregated both, on a segment level (the large matrix on the left-hand side in Figure 11) or on a division-level (the smaller matrices in Figure 11).

4 MANAGERIAL IMPLICATIONS

The next chapters will evaluate the managerial implications through inspecting the matrix as a standalone tool and through a simulation study. All the figures will be presented from the test data set, which covers the time period of the simulation from the 1st of July 2018 to the 31st of January 2019. Before going into detail with the managerial implications though, let's establish the baseline hit rates that can be used to compare the performance and accuracy of the model.

During the test period the case company had a 26.64% hit rate, which should be used as the baseline result. This means that 26.64% of all the projects in the sample are positive and 73.36% are negative, and that a random classifier would get a 26.64% hit rate by making the candidate project portfolio selections randomly. The set of projects that were categorized as high strategic fit opportunities and thus had a strategic fit value of 0.5 or higher had 19.42% hit rate and the set of low strategic fit projects had 28.65% hit rate. While this already indicates an interesting discrepancy between NCEC's strategic goals and the success in tendering competitions, the 19.42% hit rate of the high strategic fit opportunities can be thought as the baseline result for a random classifier for the two high strategic fit categories, *Focus* and *Analysis*, whereas the 28.65% hit rate acts as the baseline result for low strategic fit categories, *Cash-in* and *Ignore*.

Table 13: Baseline hit rates.

Samples	*Baseline for*	*Hit Rate*	*Number of Projects*
All tendered projects in the test set	Whole portfolio	26.64%	473
Tendered projects in the test set with high strategic fit	Focus and Analysis	19.42%	103
Tendered projects in the test set with low strategic fit	Cash-in and Ignore	28.65%	370

4.1 ANALYSING MODEL'S PERFORMANCE THROUGH THE MATRIX

First the model's performance will be evaluated as a standalone installation without forcing it to make any project selections by analysing how the projects are plotted in the matrix. This is possible by evaluating the model's ability to correctly classify the projects that were tendered by NCEC into the high and low likelihood of success categories. As the strategic fit is in this study only defined through the project's value as covered in Section 3.1.2, higher strategic fit values can be simply thought as larger project. Given the former observation, it is quite clear that there seems to be a proportionally inverse relationship between the size of the project and the likelihood of succeeding in a tendering competition when looking at the right-hand side of the Figure 15, which illustrates the position of each tendered project within the simulation period in the matrix. This implies that the larger more demanding projects that the case company is aiming to win are harder to win for the case company. In the following sections, the aim is to cover the reasons behind this finding and other remarks that have managerial implications for NCEC.

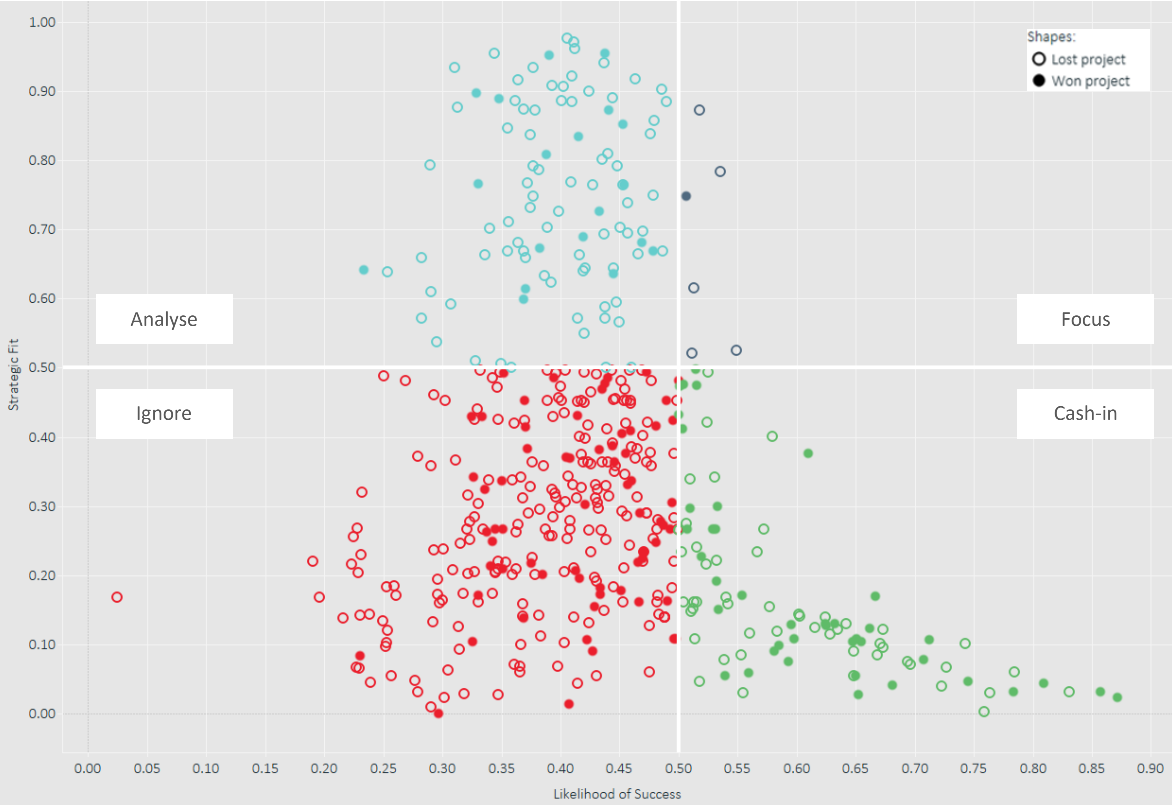
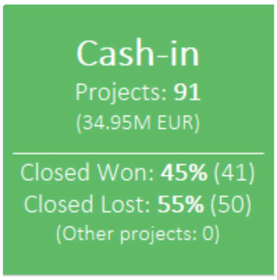


Figure 15: NCEC's tendered projects from the test period in the Strategy-Success matrix. Projects with lower strategic fit values seemed to have the highest likelihoods to succeed.

4.1.1 Cash-in Projects

The model was quite efficient in identifying potential cash-in projects out of all the low strategic fit projects in the test set. With a hit rate of 45.05% (vs. 28.65% for low strategic fit opportunities) the model quite clearly beat the baseline level. This indicates that there is an underlying pattern on what kind of projects NCEC has won out of the smaller projects with a value estimate of under 3 million euros.



Cash-in
Projects: **91**
(34.95M EUR)

Closed Won: **45%** (41)
Closed Lost: **55%** (50)
(Other projects: 0)

Figure 16: Cash-in projects. Higher hit rate than 28.65% beats random classifier.

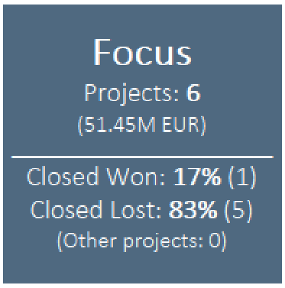
Taking only into consideration the projects that NCEC tendered during the test period's time frame 1.7.2018 - 31.1.2019, a vast majority of the cash-in projects belong into the Rock and Foundation Engineering (NRFE) division (~82% of all the projects categorized as potential "cash-in" as seen in Table 14). These projects have a high predicted likelihood of succeeding but are too small to be considered as strategically aligned projects. Looking at the projects one by one, it seems that many of these 74 projects are small foundation projects including pile-driving, earthworks and stabilisation works.

Going back to NCEC's strategic goals (Table 3), according to the case company's management team, the smaller projects seemed to have delivered, on average, lower operational profit than larger projects and therefore, the case company wished to avoid them. A very high hit-rate of 45.05%, but low historical operational profit among the "cash-in" projects might indicate that NCEC and especially its NRFE division tends to offer these certain foundation projects with too optimistic cost estimation, with too low risk reservations or simply with too low operational profit margins. As the historically low operational margins are the reason for avoiding smaller projects, higher risk reservations should be made for these under 3 million-euro NRFE division's projects at the expense of lower hit rate, but better operational margins in the future.

Table 14: Cross-tabulation of categories of the matrix and business divisions. Values in parentheses indicate the euro-based sum.

	NBAL	NNOR	NRFE	NSWE	NISE	NSRB
Focus	1 (3M)				3 (36M)	2 (12M)
Analyse	23 (245M)	5 (28M)	17 (384M)	14 (385M)	16 (191M)	22 (359M)
Cash-in	9 (4M)		74 (22M)	1 (2M)	2 (3M)	5 (4M)
Ignore	118 (100M)	2 (4M)	108 (59M)	16 (26M)	11 (19M)	24 (26M)

4.1.2 Focus Projects



Focus
Projects: **6**
(51.45M EUR)

Closed Won: **17%** (1)
Closed Lost: **83%** (5)
(Other projects: 0)

Figure 17: Focus projects. Higher hit rate than 19.42% would beat random classifier.

The model was unfortunately only able to pinpoint 6 projects that it categorized as *Focus* projects out of all the opportunities that were tendered. Out of these 6 projects only one was eventually won. This means that the model was essentially unable to find meaningful patterns in the data within the projects that had a strategic fit value of over 0.5. Even within all the projects that belong into the focus category all the likelihood of success predictions were within the range of 0.51 and 0.55. This undoubtedly indicates that the model did not have a high confidence in allocating any of the larger projects as an opportunity with a high certainty of succeeding in a tendering competition.

The previous finding can have several possible reasons. First of all, the simplest conclusion is that either from the historical point of view, these larger projects have been won by construction contractors quite randomly, or that the input variables did not have attributes within them that had high predictive power specifically to predict these larger construction projects. Another conclusion could be that as the tendering and the execution phases are to a degree conducted by different people with different competences, it seems that the teams that execute the large projects are more capable in dealing with them than the tendering organisation trying to win the large projects. Final reason for the low hit rate of the high strategic fit projects could be that the tendering organisation is simply allocating larger risk reservations and thus higher operational margins in their cost estimations, which is lowering the hit rates in the tendering phase. Note that the final reason would be a positive and a justifiable reason for having a lower hit rate. Regardless of the reason the tendering organisation should carefully analyse the explanations for such a low hit rate given that these large and high strategic fit projects also demand for much more resources in the tendering phase.

When looking at the type of projects that were tendered and belong into the focus project group according to the model in Table 16, the model categorised couple metro stations, wind power parks as well as railroad construction projects as focus projects. These certainly are projects that interest the case company, but unfortunately were lost this time around. Next, it makes sense to analyse the rest of the projects with high strategic fit values and figure out why the likelihood of succeeding in them was predicted to be rather low

4.1.3 Analyse Projects

Within the *Analyse* category similar trend continues as in the focus category. Out of the 97 projects in which the outcome of the tendering competition is known, 19.59% were won and 80.41% were lost, which indicates that the model was not able to find meaningful patterns to predict the tendering competition outcomes. Therefore, the same analysis applies as for the projects in the focus category. It is either quite random which contractor will win the larger projects within NCEC's market, or the input variables lacked features with predictive power to judge larger projects. A second iteration of the model would most likely benefit from a predictor algorithm that would separately induce decision trees for high strategic fit and low strategic fit projects in order to force the model to categorize these high strategic fit opportunities independently.

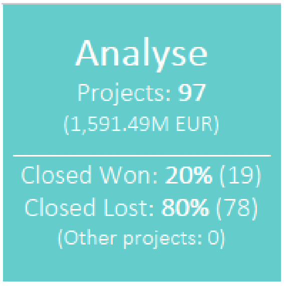


Figure 18: Analyse projects. Lower hit rate than 19.42% would beat random classifier.

4.1.4 Ignore Projects

As the Ignore projects are such that the model does not recommend being tendered, a lower hit rate indicates a better performance in this category. Similarly as with Cash-in projects, the model was able to beat the baseline classifier's 28.65% hit rate in the Ignore category with its slightly lower 23.30% hit rate. However, an interesting point here is that by setting the likelihood of success threshold lower and forcing the model to be very critical when classifying a project as negatively, the model's performance improves significantly. With around 0.3 threshold of allocating a negative judgement for the tendering competition outcome the model can identify a group of 50 projects that were tendered by NCEC but have a mere 4% probability of winning the tender. This group of projects is an interesting one as the model can visibly filter the worst projects out of the market. Next, let's dive deeper into what these projects contain and what might be the possible reasons for NCEC in trying to tender them even though they are clearly not aligned with the strategic goals and very likely to be lost.

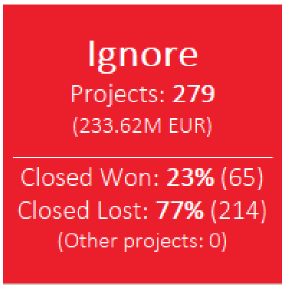


Figure 19: Ignore projects. Lower hit rate than 28.65% beats random classifier.

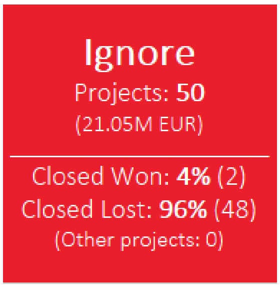


Figure 20: Ignore projects with likelihood of success threshold of 0.32.

This group of 50 projects with a 96% certainty of losing the tendering competition includes basically two types of projects: small projects by the NRFE division including especially pile driving, and small road construction projects around the Baltic region (NBAL division). After dividing these projects into the business units that were responsible for preparing the tenders, the root cause appears to be quite evident as seen in Table 15. Out of the 48 lost projects 45 of them belonged in two distinct business units: Foundation and Special Engineering (NRFE-FS) and NCEC's Estonian business unit (NBAL-EE). Looking at the reasons for losing the tendering competition, NRFE-FS unit has reported that 28 out of the 29 lost projects were actually lost due to the high price of the tender. As the model was able to pinpoint these low potential projects out of the whole population very accurately, NRFE-FS unit's managers should further investigate whether NCEC is at some kind of a disadvantage when tendering these and similar types of projects. It could be that for example, the fixed costs of the large corporation raise the total cost estimation so high that these certain kind of foundation projects are not suitable for NCEC. Similar analysis should also be made in the Estonian business unit as well, which could not be conducted here as they did not provide reasons for losing the tenders.

Table 15: Cross-tabulation of cause of loss and business unit.
NRFE-FS's 28 of the 29 lowest predicted likelihood of succeeding, were lost by due to high price.

Division	***NBAL***			***NNOR***	***NRFE***
Unit	*NBAL-EE*	*NBAL-LV*	*NBAL-LT*	*NNOR-TS*	*NRFE-FS*
High Price	0 (0M)	0 (0M)	0 (0M)	0 (0M)	28 (5M)
Other / Not stated	16 (10M)	1 (1M)	1 (1M)	1 (3M)	1 (2M)

4.1.5 Strategy-Success Matrix's Performance

In conclusion, the model seems to perform on a satisfactory level with regards to smaller, low strategic fit, projects. The reason for this can be manifold. First of all, there were a lot more samples in the training data for the decision tree algorithm to extract patterns from these smaller tendering competition outcomes as they are simply more common in the market. Secondly, it is possible that large construction projects are very unique in the sense that the outcome might be quite random from the statistical point of view. Finally, with regards to the larger projects, using merely basic project master data from the CRM system might not provide the necessary level of detail for the classification model to conduct accurate likelihood of success predictions.

Table 16: Cross-tabulation of project types and categories of the matrix. Numbers in parentheses indicate the euro-based sum.

Project Type	*Analyse*	*Cash-in*	*Focus*	*Ignore*
Biogas plant	2 (26.85M)	0 (0M)	0 (0M)	0 (0M)
Bridge structures	5 (46.87M)	0 (0M)	0 (0M)	10 (9.33M)
Earthworks	2 (11.38M)	2 (0.74M)	0 (0M)	11 (13.06M)
Environmental works	0 (0M)	0 (0M)	0 (0M)	3 (0.69M)
Foundation works	1 (6.7M)	1 (2.93M)	0 (0M)	3 (3.64M)
Industrial and production premises	2 (13.37M)	0 (0M)	0 (0M)	1 (2M)
Other	26 (512.9M)	15 (2.71M)	1 (3.3M)	111 (96.96M)
Other infrastructure construction	1 (15.1M)	2 (0.45M)	0 (0M)	8 (9.45M)
Other infrastructure works	1 (5.83M)	34 (11.37M)	0 (0M)	37 (15.64M)
Parking Facility	2 (14.29M)	1 (1.66M)	0 (0M)	3 (2.87M)
Pile driving	5 (78.19M)	26 (2.91M)	0 (0M)	32 (10.35M)
Rail network	2 (11.4M)	1 (2.84M)	1 (9M)	0 (0M)
Rock quarrying works	11 (301.93M)	1 (0.65M)	0 (0M)	4 (5.03M)
Sewage treatment plant	1 (33.69M)	1 (1.5M)	0 (0M)	1 (1.49M)
Span building	1 (5.82M)	0 (0M)	0 (0M)	0 (0M)
Special piling	0 (0M)	1 (0.5M)	0 (0M)	1 (2.04M)
Street and road building	15 (292.52M)	4 (4.22M)	0 (0M)	38 (33.86M)
Underground facilities	5 (65.92M)	0 (0M)	2 (22.85M)	3 (6.2M)
Waste treatment plant	1 (30M)	0 (0M)	0 (0M)	0 (0M)
Water engineering	5 (60.19M)	2 (2.47M)	0 (0M)	8 (11.81M)
Water supply	3 (17.54M)	0 (0M)	0 (0M)	3 (5.75M)
Water treatment plant	1 (10M)	0 (0M)	0 (0M)	0 (0M)
Wind Power	5 (31M)	0 (0M)	2 (16.3M)	2 (3.45M)
Grand Total	97 (1591.49M)	91 (34.95M)	6 (51.45M)	279 (233.62M)

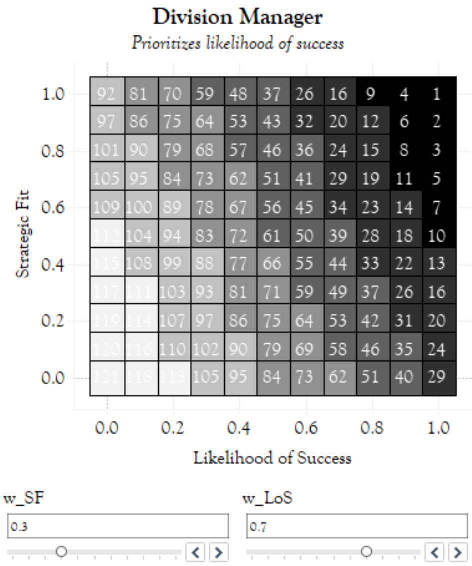
However, this conclusion might only be natural and even preferred from the business point-of-view. Larger high strategic fit projects always demand for careful qualitative and quantitative analysis, and they will regardless receive a lot of attention from the management team. Smaller low strategic fit projects on the other hand might receive less attention in the candidate project selection phase as their business impact for the whole segment is smaller. With the developed model, NCEC can easily and quite trustfully explore the market opportunities that could be used to Cash-in and use the model's suggestions in combination with the managers' expert judgement to make more accurate candidate project selections. A very high accuracy of 96% when filtering the worst projects out from the market can also significantly assist the managers to avoid the most disadvantageous project opportunities. Finally, as the model found relatively trustworthy patterns from the data to pinpoint the worst projects, the managers should use the information to identify the core reasons for the recurring losses in these specific tendering competitions. Decision tree model's rule induction ability can be very helpful in this analysis as the managers can directly look at the rules behind the predictions.

4.2 Testing Model's Performance Through the Simulation

After analysing how the projects during the test period were plotted in the matrix, next the thesis will demonstrate the model's performance through a simulation study. It is possible for the model to autonomously make selections based on the scoring equation and constraints described in 3.3: Constructing the Simulation. This simulation will give an idea of how the model would perform in a real-life situation and it will also demonstrate its accuracy based on realistic constraints. It will also be very interesting to see how similar the project selections were for the model and NCEC as well as how accurately NCEC is following the strategic goals it has set for itself.

4.2.1 Division-manager: Priority on Likelihood of Success

In the first example the model is simulating the candidate project selection in a fashion that a division manager could perform them. A division manager has the responsibility to maintain a high orderbook in order to keep her organisation up-and-running and employed. Therefore, she might prefer to tender projects, which are likely to be won as her priority is to maintain the business. Figure 21 visualises the order of candidate project selections that a division manager might make.



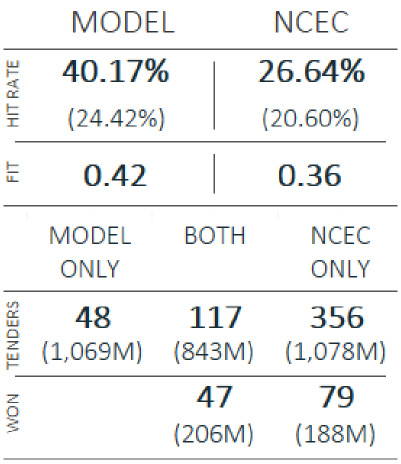
92	81	70	59	48	37	26	16	9	4	1
97	86	75	64	53	43	32	20	12	6	2
101	90	79	68	57	46	36	24	15	8	3
105	95	84	73	62	51	41	29	19	11	5
109	100	89	78	67	56	45	34	23	14	7
112	104	94	83	72	61	50	39	28	18	10
115	108	99	88	77	66	55	44	33	22	13
117	111	103	93	81	71	59	49	37	26	16
119	114	107	97	86	75	64	53	42	31	20
120	116	110	102	90	79	69	58	46	35	24
121	118	113	105	95	84	73	62	51	40	29

w_SF: 0.3

w_LoS: 0.7

Figure 21: Division manager's order of candidate project selections.

In the simulation, the weight of strategic fit was set at 0.3 and weight of likelihood of success at 0.7 like in Figure 21. These weights are appropriate, because they don't completely neglect the less-weighted measure of the matrix, but still have enough impact to highlight the larger weighted measure. Figure 22 illustrates the segment-level results from the simulation whereas Table 17 demonstrates the division-level results.



	MODEL	NCEC
HIT RATE	40.17% (24.42%)	26.64% (20.60%)
FIT	0.42	0.36

	MODEL ONLY	BOTH	NCEC ONLY
TENDERS	48 (1,069M)	117 (843M)	356 (1,078M)
WON		47 (206M)	79 (188M)

Figure 22: Segment-level simulation results: NCEC vs. model (Division manager). $w_{SF} = 0.3$ *&* $w_{LoS} = 0.7$.

The model beat NCEC's actual project selections made during the test period on all of the ratios on the segment-level. Portfolio's count-based hit rate was improved from the initial 26.64% to 40.17% and project value-based hit rate raised from 20.60% to 24.42%. The results can be explained with model's ability to distinguish the projects with low-strategic fit and low likelihood to be won from the market as demonstrated in the previous chapter. This was true especially for the NRFE and NBAL divisions, which contained a very large number of small projects that the tendering organisation was not able to win, but the model was able to avoid. When looking at the value and count-based hit rates as well as average strategic fit measures of NCEC's and model's selection in Table 17, it becomes evident that the model clearly outperformed NCEC's candidate project selections with regards to these two divisions as well.

Model's selections outperformed NCEC's selections measured with the portfolio's average strategic fit measure regardless of the low 0.3 strategic fit weight. The average strategic fit increased from the initial 0.36 to 0.42. When further dividing the results on the division-level, there are certain divisions, namely NBAL and NRFE, that have a very large number of projects that fall below the 3-million-euro threshold value and are therefore, categorized as strategically not-aligned when compared against the strategic goals. All other divisions reach the 0.5 strategic fit threshold level and can be considered to follow the segment-level strategic goals on average. This holds true for both, NCEC's original selections as well as model's selections.

The previous notion raises a question of whether NBAL and NRFE divisions are actually aligned with the direction NCEC's group-level strategists want to take the company towards to. If the small projects are supporting some larger and more strategically aligned undertakings by NCEC's other divisions, keeping NBAL and NRFE divisions under the segment is reasonable. However, if there are no synergies between the projects of the two divisions and other NCEC's business divisions or segments, these two organisations would most-likely make better operational profit without the large overhead costs that are casted from the group as they explicitly hinder the profit performance of small projects. In any case, the managers of NBAL and NRFE divisions should reflect what kind of other more strategically aligned opportunities there can be found from the market. If there are no such opportunities it could also be worthwhile to analyse whether defining the strategic goals differently for these two divisions would make sense, given that the divisions are making positive profit and thus, want to be maintained. Also, the managers could at any rate ignore the low strategic fit projects that have a low likelihood of success as the model was quite accurate in filtering the very worst out from the market.

Table 17: Cross-tabulation of simulation results between division manager's simulation and NCEC. The better result from each performance measure is underlined.

		NCEC			MODEL		
	Similarity (value-based)	Strategic Fit	Hit Rate (Count)	Hit Rate (Value)	Strategic Fit	Hit Rate (Count)	Hit Rate (Value)
NRFE	88 %	0.24	33.17 %	21.81 %	0.29	42.55 %	21.99 %
NISE	25 %	0.6	21.88 %	7.78 %	0.84	0.00 %	0.00 %
NSRB	50 %	0.5	16.98 %	6.02 %	0.76	28.57 %	4.56 %
NSWE	21 %	0.6	19.35 %	29.43 %	0.86	50.00 %	74.54 %
NNOR	13 %	0.58	14.29 %	4.48 %	0.5	0.00 %	0.00 %
NBAL	20 %	0.36	24.50 %	35.75 %	0.51	40.00 %	58.39 %
SEGMENT	44 %	0.36	26.64 %	20.60 %	0.42	40.17 %	24.42 %

Looking at the similarities between the project selections of the model and NCEC, out of all the 473 projects that were tendered during the test period 117, or 24.74% were tendered by both. In value-based terms the model and NCEC used 43.88% of their 1 921 million-euro tendering resources similarly. Value-based similarities were especially high with regards to NRFE division, in which the model and NCEC hand-picked the very same high strategic fit opportunities from the market. Continuing on the previous point about NRFE division's strategic un-alignment, focusing on these, and similar, mutually picked and strategically well-aligned projects could provide NRFE division better performance going further.

All-in-all, the model is useful for some specific divisions of NCEC's infrastructure segment. NRFE, NSRB, NSWE and NBAL divisions could specifically benefit from using the model as its suggestions beat the baseline figures on all performance measures as seen in Table 17. For the remaining NISE and NNOR divisions however, the model did not provide any useful results as NCEC and the model did not make any mutual selections that were won and therefore, the hit rates were zero percent for the model's selections.

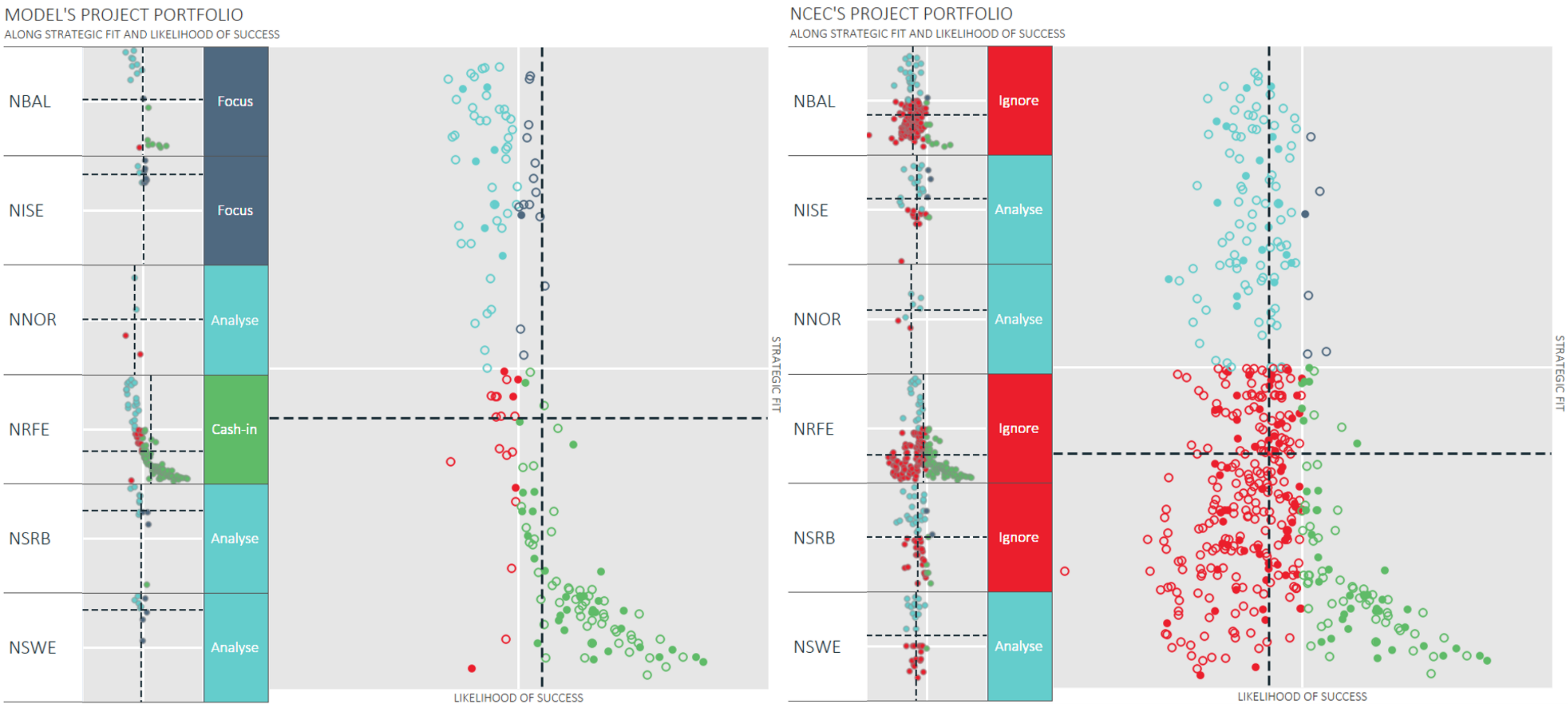
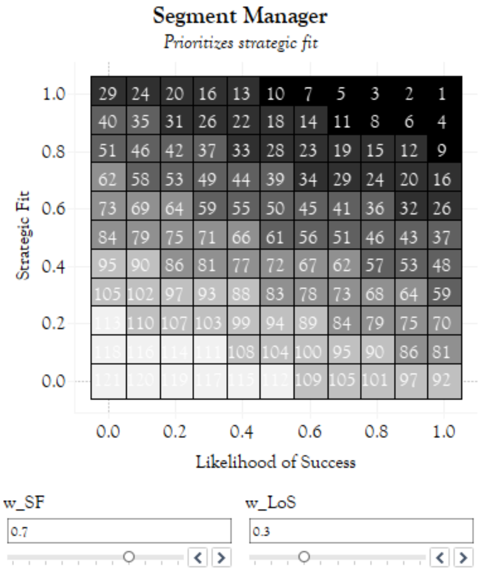


Figure 23: Comparison of model's (Division manager) and NCEC's selections.
Instead of selecting low likelihood of success projects, model used the resource to tender more projects with high strategic fit.

4.2.2 Segment-manager: Priority on Strategic Fit

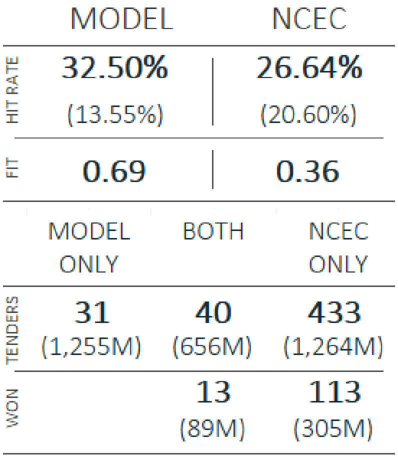
In the second example, the model will be acting in a way a segment manager might behave. Segment manager works closely with the group's management team and is responsible for making sure that the corporate-level strategic goals are cascaded downwards, and that the planned strategy will also realize. Therefore, her first priority might be to favour projects that are aligned with the strategic goals rather than striving for easy successes regardless of the consequences. Figure 24 demonstrate how a segment manager might make her candidate project selections.



29	24	20	16	13	10	7	5	3	2	1
40	35	31	26	22	18	14	11	8	6	4
51	46	42	37	33	28	23	19	15	12	9
62	58	53	49	44	39	34	29	24	20	16
73	69	64	59	55	50	45	41	36	32	26
84	79	75	71	66	61	56	51	46	43	37
95	90	86	81	77	72	67	62	57	53	48
105	102	97	93	88	83	78	73	68	64	59
113	110	107	103	99	94	89	84	79	75	70
118	116	114	111	108	104	100	95	90	86	81
121	120	119	117	115	112	109	105	101	97	92

Figure 24: Segment manager's order of candidate project selections.

The same weights were used as in the Figure 24. Strategic fit had a multiplier 0.7 whereas the likelihood of success measure had the weight of 0.3 to the total score. Figure 25 represents the segment-level results for the simulation and Table 18 demonstrates the results divided per division.



	MODEL		NCEC
HIT RATE	32.50% (13.55%)		26.64% (20.60%)
FIT	0.69		0.36
	MODEL ONLY	BOTH	NCEC ONLY
TENDERS	31 (1,255M)	40 (656M)	433 (1,264M)
WON		13 (89M)	113 (305M)

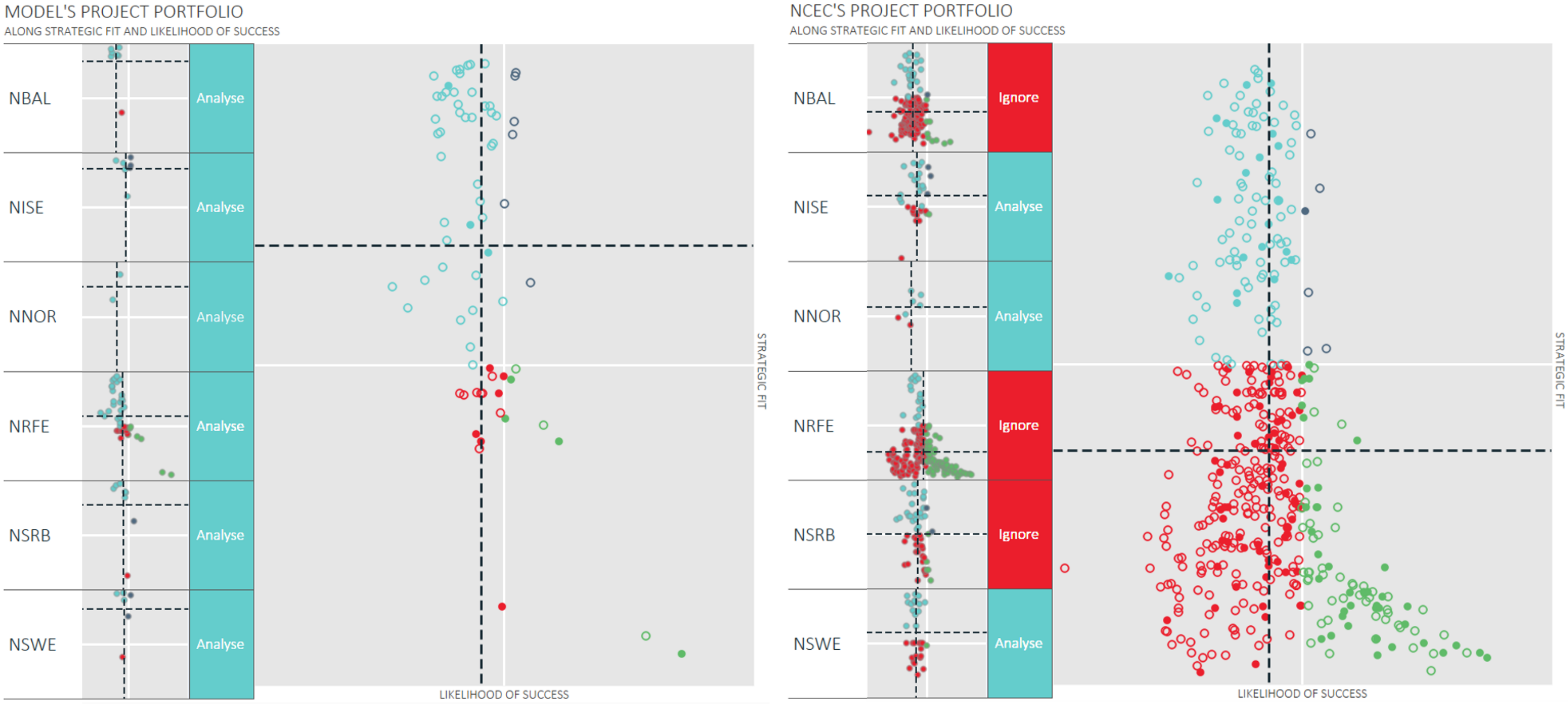
Figure 25: Segment-level simulation results: NCEC vs. model (Segment manager) $w_{SF} = 0.7$ & $w_{LoS} = 0.3$.

When looking at Table 18 it becomes quite evident that most of the performance measures from this simulation attempt are unreliable, because the model and NCEC largely selected different projects. Even though, on the segment level the count-based hit rate rose from 26.64% to 32.50%, it is derived from only 40 commonly selected projects, which only cover less than one-tenth of the projects tendered by NCEC. The reason for the low value-based hit rate simply is that as NCEC's baseline hit rate was low for the high strategic fit projects and the decision tree algorithm was inaccurate in predicting the tender competition outcomes for them, it is natural that the hit rate decreases. A conclusion from this trial is that the whole model fails at its candidate project selections when high strategic fit projects are prioritized, because the decision tree predictor behind the model was in accurate in predicting the high strategic fit projects.

Table 18: Cross-tabulation of simulation results between segment manager's simulation and NCEC. Because the model and NCEC did not share many of the project selections, NISE, NSWE, NNOR and NBAL results are unreliable.

		NCEC			MODEL		
	Similarity (value-based)	**Strategic Fit**	**Hit Rate (Count)**	**Hit Rate (Value)**	**Strategic Fit**	**Hit Rate (Count)**	**Hit Rate (Value)**
NRFE	12 %	0.24	33.17 %	21.81 %	0.59	34.38 %	21.37 %
NISE	25 %	0.6	21.88 %	7.78 %	0.86	0.00 %	0.00 %
NSRB	41 %	0.5	16.98 %	6.02 %	0.79	25.00 %	0.06 %
NSWE	0.2 %	0.6	19.35 %	29.43 %	0.83	100.00 %	100.00 %
NNOR	0 %	0.58	14.29 %	4.48 %	0.78	0.00 %	0.00 %
NBAL	7 %	0.36	24.50 %	35.75 %	0.85	0.00 %	0.00 %
SEGMENT	34 %	**0.36**	**26.64 %**	**20.60 %**	**0.69**	**32.50 %**	**13.55 %**

An interesting implication of this trial is that as the similarity between the segment-manager simulation and NCEC's actual selections is quite low, it indicates that either the strategic goals defined earlier do not accurately capture the strategic objectives of NCEC, or NCEC is not following the strategic objectives defined by the group and the segment-level managers. Whichever the case, this conclusion first of all, highlights the difficulty of formulating strategic goals that truly exhaustively capture the strategic objectives of a company, and secondly, emphasizes the difficulty of following and executing the defined strategic objectives. Most likely both of the factors play a part in this result.



MODEL'S PROJECT PORTFOLIO
ALONG STRATEGIC FIT AND LIKELIHOOD OF SUCCESS

NCEC'S PROJECT PORTFOLIO
ALONG STRATEGIC FIT AND LIKELIHOOD OF SUCCESS

Figure 26: Comparison of model's (Segment manager) and NCEC's selections.
This time around model radically avoided all the small low strategic fit projects and only selected large high strategic fit projects. As a result the value-based hit rate dropped, but count-based hit rate and average strategic fit rose.

5 CONCLUSIONS

This thesis is motivated by the difficulty of following the planned strategy in a project-based business. In addition to this difficulty, these businesses rarely utilise statistical methods to alleviate the risk of selecting projects that are unlikely to be successful. In response to these obstacles, a matrix was constructed based on existing frameworks about strategic fit estimation and project performance prediction. The following section will bring the thesis to a closure by summarizing and discussing the whole study and answering the research questions that were introduced in the beginning.

5.1 RESEARCH SUMMARY

Case company had just lost a large railroad project, which had a cost calculation phase occupying a team of 10-20 top-level civil engineers for a half-year period. Participating in a tendering competition is manual labour, which has to be covered by the contractor itself. Losses in tendering competitions can be perceived to directly increase the overhead costs of the organisation, which then lower the profit margins of projects in the execution phase as they have to cover the overhead costs for the whole organisation.

The first motivation for the study was that some candidate project selections, especially with regards to the smaller projects are often done based on gutfeel instead of using any statistical tools to guide the tendering portfolio selection. Therefore, the case company wished to develop a framework, which would alleviate the risk of selecting candidate projects to be tendered that they are inherently losing. The x-axis of the matrix, likelihood of success, was constructed to answer to this requirement. The first step in defining the likelihood of success is to identify the relevant performance measure that encompasses whether the project selection was successful. For the case company it was the likelihood of winning a tender, because preparing tenders is costly and these costs increase the fixed costs and impact the profit performance of the organisation. The second step was then to identify the most effective algorithms to predict the likelihood of success values, and for this study decision trees were selected due to their good performance in the preliminary study along with other benefits.

Secondly, for project-based organisations it is often difficult to follow the planned strategy of the executive team, because the project opportunities that are available in the market are always rather limited. As already highlighted in Archer & Ghasemzadeh (1999), the project portfolio selections that are then conducted from this limited set of opportunities are the driving force to bring the planned strategy into reality for a project-based company. Without proper guidelines and governance from the strategy point of view in the selection phase, the organisation can easily be filled with emergent strategies, which might take the company in an unintended direction (Mintzberg, 1992; Mintzberg & Waters, 1985). If done rigorously, strategists should incorporate their expert knowledge of the company's competitive advantage within the strategic objectives and goals about what is the most beneficial path for the company, whether measured by profitability, customer retention rate or any other KPI. By utilising these objectives in the project portfolio selection, the managers can make well-informed selections to deliver the best possible performance for the company in the future.

The y-axis of the matrix, strategic fit, was therefore selected to capture the aforementioned strategic aspect early on in the candidate project selection phase. Strategic fit was defined as the alignment between the characteristics of the project and the strategic goals of the company. By comparing the attributes of the projects, whether in qualitative or quantitative terms against the strategic goals, it is possible to derive a single value of strategic fit, which managers can utilize to conduct well-informed candidate project selections. A five-step strategic fit measuring process was then compiled based on earlier research about the estimation process.

And on that note, the first and second research questions, "*1) How to reliably estimate the strategic fit of a candidate project with the company's strategic objectives?*" and "*2) How to reliably estimate the probability of winning the tendering competition for a given candidate project?*" were also answered during the Sections 2.2: Strategy and Strategic Fit and 2.3: Project Performance as was just summarized.

By combining these two measures into a matrix format, the Strategy-Success Matrix was formed. It is divided into four quartiles that confine the different types of projects there are in the market. The first quartile, *Ignore*, comprises of all the low strategic fit and likelihood of success projects. These projects do not either align with the strategic objective nor have been successful for the company in the past and thus should be ignored. The second quartile, *Analyse*, contains all the projects with high strategic fit, but low predicted likelihood of success. These are the project opportunities that are aligned with the strategy but have not been successful and thus, should be further analysed. Third quartile, *Cash-in*, cover all the projects that are not aligned with the defined strategic goals, but are predicted to be successful. These projects are safe picks, but do not take the company into the desired direction and thus, can be used to cash-in. Final quartile, *Focus*, constitutes of the first priority projects that are well-aligned with the strategic objectives as well as rather likely to be successful. These are the top priority projects that should be selected first.

Next, the matrix was used in practice to contrast how the project selections made by NCEC compared against the scoring model's selections during the simulation period starting from the 1st of July 2018 and ending on 31st of January 2019. Likelihood of success and strategic fit values were first calculated in Python and based on them a score was calculated for each project to prioritize them. The final results were then visualized in Tableau in an interactive format.

By only inspecting the accuracies within the four quartiles and how the projects were plotted in the matrix during the test period the following main findings were made. First, the matrix was not equally accurate for all of the four quartiles. The likelihood of success predictions were only accurate for low strategic fit projects. It also seemed that there was an inverse relationship between strategic fit and the predicted likelihood of success values. This might indicate a discrepancy between the planned strategy and the competences of the company.

Second finding was that case company's NRFE division contained a large number of small pile-driving, earthworks and stabilisation projects in the Cash-in quartile that the model was accurate in predicting positively correctly. The implications from this are two-fold: if the operational margins of these projects are low, the division is probably offering these projects with too low margins, or if the operational margins are on a decent level, NCEC's NRFE division has a clear competitive advantage when it comes to these certain types of projects.

Third finding was that the model was very accurate in identifying the 50 most unlikely to be won projects within the ignore segment with an accuracy of 96%. The projects included small foundation projects in Finland and certain road construction projects in Estonia that were regardless tendered and offered even though the tendering competitions have resulted in continuous losses. Looking at the reasons for losing these projects, many of them were due to the high price of the tender. It could be that NCEC's high overhead costs put the company at a disadvantage when competing against smaller more agile contractors for these smaller projects, and thus NCEC keeps repeatedly losing them. As these projects are not even aligned with the strategic objectives, it seems quite clear that these and similar projects should be avoided in the future.

Finally, a small simulation study was conducted to compare how the selections of the scoring model would contrast against NCEC's selections during the test period. First, division-manager's selections were simulated by emphasising the likelihood of succeeding in a tendering competition. These settings correspond to her willingness to win projects as she is responsible of maintaining a high orderbook and keeping her employees employed. On average, the count-based hit rate rose from 26.64% to 40.17% whereas the value-based hit rate rose from 20.60% to 24.42%. This improvement can be explained with the matrix's accuracy to filter out the best and worst low strategic fit projects from the market. The scoring model and NCEC shared 44% of the selection when measured by the project's value, and the strategic fit of model's portfolio rose to 0.42 and was 0.06 units better than NCEC's initial 0.36 average strategic fit.

Inspecting the selections on a division-level clearly highlighted, which divisions could benefit from the model the most. NRFE, NSRB, NSWE and NBAL divisions' results were clearly improved by the model's selections and these divisions could reliably utilize model's suggestions in the future. Especially, NRFE and NBAL divisions would benefit from the model as they both contained a lot of smaller projects that are still important for the divisions and which, the model was able to accurately predict well. In these divisions the initial count-based hit-rate was improved from 33.17% to 42.55% for NRFE and from 24.50% to 40.00% for NBAL.

Lastly, an attempt was made at trying to simulate how a segment-manager would make the candidate project selections by emphasizing the weight of strategic fit. However, as the model then simply picked projects in the order of project value, no meaningful results were produced from that experiment.

And subsequently, the third research question, "3) *How can these (Strategic Fit and Likelihood of Success) estimates be used to guide the process of selecting which candidate projects to pursue?*", was answered in the Section 4: Managerial Implications. In essence, the matrix can be a useful tool for any project-oriented business as it forces the practitioners to critically assess the strategic alignment versus the actual likelihood of succeeding in the project.

5.2 LIMITATIONS OF THE STUDY

There are couple main limitations that were identified regarding the study. First of all, the demonstration only covers the perspective of one company. The matrix should be validated in various contexts in order to generalize its performance. Other contexts might also have different kind of data available, which might result in worse or sometimes better performance of the matrix.

Second, the performance of the scoring model is directly related to the performance of the matrix's dimensions. If either the likelihood of success predictions are inaccurate or the strategic goals and as such the strategic fit values do not reflect the real intended strategy of the organisation, the matrix and the scoring model will yield useless results. The former was partially true in this case study as the predictions for the high strategic fit opportunities were not optimal. Therefore, careful planning should always be taken, when the matrix is being planned to be used.

Thirdly, the strategic fit measure did not involve qualitative metrics in this study due to the way the strategic goals were defined. Validation of this method will be left for further studies although, previous studies suggests various methods for converting them into numeric strategic fit values as was covered in Sections 2.2 and 7.4.

Finally, as the data source that was used to produce both, the likelihood of success as well as strategic fit values was based on secondary operational data, which was generated by the sales engineers who manually insert the values into the CRM system, it is prone for human errors and inaccuracies. These inaccuracies will especially be detrimental for the likelihood of success predictions even though, measures were taken to counter outliers. In this thesis, mainly the pre-processing of data, the usage of ensemble methods and limitations to the sizes of the leaves and the trees were used to increase the generalizability of the final predictor. Please note that this is a very common issue in practice-oriented studies such as this one and had to be accepted from the beginning.

5.3 SUGGESTIONS FOR FURTHER RESEARCH

There are some clear research avenues that this study will open for other researchers and practitioners. First of all, a follow-up study could focus on the socio-economic aspect of adopting this or a similar matrix in practice. With such a study it could be possible to reveal subtle traits that may hinder or assist the adoption of this kind of an expert system in practice.

For example, it would be very interesting to study, if there are differences in the adoptability of different machine learning algorithms used to produce the likelihood of success predictions. If e.g. the practice-oriented users would prefer to clearly see the reasons why a certain likelihood of success was assigned for the specific project, then a simple decision tree could be better for the final predictions as long as the accuracy is sufficient. However, if the users do not care about the transparency of the algorithm, the most accurate and generalizable algorithm should always be prioritized.

A second suggestion would be to duplicate this study in different industries and contexts to see and validate how it performs on a general level. It is important to highlight that this kind of a matrix might not be limited to project-oriented settings as with small modifications it should be suitable for other contexts as well. An experiment in a normal customer-oriented business could be interesting as the matrix could, for example, be used to identify the most important customer segments or even individual prospects from the market based on their characteristics and features.

Finally, a study comparing, which kind of likelihood of success measures would perform the best in different contexts would be interesting. In this study, only likelihood of winning a tender competition was being predicted, but just as well profit performance, lifetime value or any other important KPI for the specific industry could be estimated. The selection of performance measure might also affect the adoptability and accuracy of the matrix.

6 REFERENCES

Aaltonen, P., 2010. *Co-selection in R&D Project Portfolio Management: Theory and Evidence.* Espoo: Helsinki University of Technology, Department of Industrial Engineering and Management, Doctoral Dissertation Series.

Akintoye, A. & MacLeod, M., 1997. Risk analysis and management in construction. *International Journal of Project Management,* 15(1), pp. 31-38.

Archer, N. & Ghasemzadeh, F., 1999. An integrated framework for project portfolio selection. *International Journal of Project Management,* 17(4), pp. 207-216.

Armstrong, S. J. & Brodie, R. J., 1994. Effects of portfolio planning methods on decision making: Experimental results. *Journal of Marketing Management,* Volume 7, pp. 105-129.

Artto, K., Martinsuo, M., Dietrich, P. & Kujala, J., 2008. Project strategy - strategy types and their contents in innovation projects. *Journal of Managing Projects in Business,* 1(1), pp. 49-70.

Bergeron, F., Raymond, L. & Rivard, S., 2001. Fit in strategic information technology management research: An empirical comparison of perspectives. *Omega: The International Journal of Management Science,* Volume 29, pp. 125-142.

Beynon, M. J., Andrews, R. & Boyne, G. A., 2010. Evidence-based modelling of strategic fit: An introduction to RCaRBS. *European Journal of Operational Research,* 207(2), pp. 886-896.

Bilal, M. et al., 2016. Big Data in the construction industry: A review of present status, opportunities, and future trends. *Advanced Engineering Informatics,* 30(3), pp. 500-521.

Blichfeldt, B. S. & Eskerod, P., 2008. Project portfolio management – there's more to it than what management enacts. *International Journal of Project Management,* Volume 26, pp. 357-365.

Breiman, L., 1996. Bagging predictors. *Machine Learning,* 24(2), pp. 123-140.

Breiman, L., Friedman, J. H., Olshen, R. A. & Stone, C. J., 1984. *Classification and regression trees.* Belmont: Wadsworth International Group.

Center for Business Practices, 2005. *Measures of Project Management Performance and Value.* [Online]
Available at: https://www.pmsolutions.com/audio/PM_Performance_and_Value_List_of_Measures.pdf
[Accessed 21 January 2019].

Chang, A. & Leu, S.-S., 2006. Data mining model for identifying project profitability variables. *International Journal for Project Management,* 24(3), p. 199.206.

Chang, P.-T., Hung, L.-T., Pai, P.-F. & Lin, K.-P., 2013. Improving project-profit prediction using a two-stage forecasting system. *Computers & Industrial Engineering,* 66(4), pp. 800-807.

Chang, S. L., Wang, R. C. & Wang, S. Y., 2007. Applying a direct multi-granularity linguistic and strategy-oriented aggregation approach on the assessment of supply performance. *European Journal of Operational Research,* 177(2), pp. 1013-1025.

Chan, Y. E., Huff, S. L. & Copeland, D. G., 1997. Assessing realized information systems strategy. *Journal of Strategic Information Systems,* 6(4), pp. 273-298.

Chen, D.-N. & Liang, T.-P., 2011. Knowledge evolution strategies and organizational performance: A strategic fit analysis. *Electronic Commerce Research and Applications,* 10(1), pp. 75-84.

Chen, H. L., Chen, C.-I., Liu, C.-H. & Wei, N.-C., 2013. Estimating a project's profitability: A longitudinal approach. *International Journal of Project Management,* 31(3), pp. 400-410.

Chen, Z., Wanke, P. & Tsionas, M. G., 2018. Assessing the strategic fit of potential M&As in Chinese banking: A novel Bayesian stochastic frontier approach. *Economic Modelling,* Volume 73, pp. 254-263.

Christiansen, J. K. & Varnes, C. J., 2008. From models to practice: Decision making at portfolio meetings. *International Journal of Quality & Reliability Management,* 25(1), pp. 87-101.

Clark, P. & Boswell, R., 1991. Rule induction with CN2: Some recent improvements. *Machine Learning - Proceedings of the Fifth European Conference,* pp. 151-163.

Clegg, S., Carter, C., Kornberger, M. & Schweitzer, J., 2011. Introduction. In: *Strategy: Theory and Practice.* London: SAGE Publications Ltd, pp. 2-17.

Cooper, R., 1981. An Empirically Derived New Product Project Selection Model. *IEEE Transactions on Engineering Managment,* 28(3), pp. 54-61.

Cooper, R. G., 1993. *Winning At New Products.* 2nd ed. MA: Addison-Wesley.

Costantino, F., Di Gravio, G. & Nonino, F., 2015. Project selection in project portfolio management: An artificial neural network model based on critical success factors. *International Journal of Project Management,* 33(8), pp. 1744-1754.

Drummond, C. & Holte, R., 2003. C4.5, class imbalance and cost sensitivity: Why under-sampling beats over-sampling. *Workshop on Learning from Imbalanced Datasets,* Volume 2.

Encyclopedia, 2019. *Utility Function: International Encyclopedia of the Social Sciences.* [Online] Available at: https://www.encyclopedia.com/social-sciences/applied-and-social-sciencesmagazines/utility-function [Accessed 7 December 2019].

Englund, R. & Graham, R., 1999. From experience: Linking projects to strategy. *Journal of Product Innovation Management,* 16(1), pp. 52-64.

Engwall, M., 2003. No project is an island: Linking projects to history and context. *Research Policy,* Volume 32, pp. 5789-5808.

Esty, B. C., 2003. *Modern project finance: a casebook.* New York: John Wiley & Sons.

Finlex, 2016. *Laki julkisista hankinnoista ja käyttöoikeussopimuksista.* [Online] Available at: http://www.finlex.fi/fi/laki/ajantasa/2016/20161397 [Accessed 12 December 2019].

Fiss, P. C., 2011. Building better causal theories: A fuzzy set approach to typologies in organization research. *Academy of Management Journal,* 54(2), pp. 393-420.

Frame, J. D., 2003. *Project finance: tools & techniques.* 1st ed. s.l.:UMT Press.

Freund, Y. & Schapire, R. E., 1996. *Experiments with a new boosting algorithm.* Bari, Machine Learning: Proceedings of the Thirteenth International Conference.

Hand, D., 1998. Data mining-reaching beyond statistics. *Research in Official Statistics,* 1(2), pp. 5-17.

Han, S., Kim, D. & Kim, H., 2007. Predicting Profit Performance for Selecting Candidate International Construction Projects. *Journal of Construction Engineering and Management,* 133(6), pp. 425-436.

Hedley, B., 1977. Strategy and the "Business Portfolio". *Long Range Planning,* Volume 10, pp. 9-15.

Henderson, J. C. & Venkatraman, N., 1993. Strategic alignment: Leveraging information technology for transforming organizations. *IBM Systems Journal,* 38(2 & 3), pp. 472-484.

Hevner, A., March, S., Park, J. & Ram, S., 2004. Design science in information systems research. *Management Information Systems Quarterly,* 28(1), pp. 75-105.

Jacobson, R. & Aaker, D., 1985. Is market share all that it’s cracked up to be?. *Journal of Marketing,* Volume 49, pp. 11-21.

Japkowicz, N. & Stephen, S., 2002. The class imbalance problem: A systematic study. *Intelligent Data Analysis,* pp. 429-449.

Kass, G., 1980. An exploratory technique for investing large quantities of categorical data. *Applied Statistics,* 29(2), pp. 199-127.

Kayri, M. & Kayri, İ., 2015. The comparison of gini and twoing algorithms in terms of predictive ability and misclassification cost in data mining: An empirical study. *International Journal of Computer Trends and Technology,* 27(1), pp. 21-30.

Kester, L., Griffin, A., Hultink, E. J. & Lauche, K., 2011. Exploring portfolio decision-making processes. *Journal of Product Innovation Management,* 28(5), pp. 641-661.

Khurana, A. & Rosenthall, S. R., 1997. Integrating the fuzzy front end of new product development. *Sloan Management Review,* Volume 38, pp. 103-120.

Killen, C. P., Hunt, R. A. & Kleinschmidt, E. J., 2008. Learning investments and organizational capabilities. Case studies on the development of project portfolio management capabilities. *International Journal of Managing Projects in Business,* 1(3), pp. 334-351.

Kim, D., Han, S. H., Kim, H. & Park, H., 2009. Structuring the prediction model of project performance for international construction projects: A comparative analysis. *Expert Systems with Applications,* 36(2 part 1), pp. 1961-1971.

Kim, G.-H., An, S.-H. & Kang, K.-I., 2004. Comparison of construction cost estimating models based on regression analysis, neural networks, and case-based reasoning. *Building and Environment,* 39(10), pp. 1235-1242.

Kozachenko, L. F. & Leonenko, N. N., 1987. Sample estimate of the entropy of a random vector. *Problemy Peredachi Informatsii,* 23(2), pp. 9-16.

Kumar, D. U., Nowicki, D., Ramírez-Márquez, J. E. & Verma, D., 2008. On the optimal selection of process alternatives in a Six Sigma implementation. *International Journal of Production Economics,* 111(2), pp. 456-467.

Kumar, D. U., Saranga, H., Ramírez-Márquez, J. E. & Nowicki, D., 2007. Six sigma project selection using data envelopment analysis. *The TQM Magazine,* 19(5), pp. 419-441.

Lahdenperä, P., 2009. The competitive single target-cost approach. *VTT Tiedotteita - Research Notes 2472,* pp. 1-79.

Larose, D., 2005. *Discovering knowledge in data: An introduction to data mining.* 1st ed. New Jersey: John Wiley & Sons, Inc.

Larsen, J. & Goutte, C., 1999. On optimal data split for generalization estimation and model selection. *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing IX,* pp. 225-234.

Leśniak, A. & Plebankiewicz, E., 2015. Modeling the decision-making process concerning participation in construction bidding. *Journal of Management in Engineering,* 31(2).

Li, T., Ng, T. & Skitmore, M., 2016. Modeling multi-stakeholder multi-objective decisions during public participation in major infrastructure and construction projects: A decision rule approach. *Journal of Construction Engineering and Management,* 31(2).

Loch, C., 2000. Tailoring product development to strategy: Case of a European technology manufacturer. *European Management Journal,* 18(3), pp. 246-258.

Love, P. et al., 2002. Using systems dynamics to better understand change and rework in construction project management systems. *International Journal of Project Management,* Volume 20, pp. 425-436.

Martinsuo, M., 2013. Project portfolio management in practice and in context. *International Journal of Project Management,* 31(6), pp. 794-803.

Martinsuo, M. & Lehtonen, P., 2009. Project autonomy in complex service development networks. *International Journal of Managing Projects in Business,* 2(2), pp. 261-281.

McLaren, T., Head, M. & Yuan, Y., 2004. Strategic fit of supply chain management information systems: A measurement model. *International Conference on Information Systems 2004 Proceedings,* pp. 597-606.

Meilich, O., 2006. Bivariate models of fit in contingency theory: Critique and a polynomial regression alternative. *Organizational Research Methods,* Volume 9, pp. 161-193.

Meskendahl, S., 2010. The influence of business strategy on project portfolio management and its success – a conceptual framework. *International Journal of Project Management,* 28(8), pp. 807-817.

Mintzberg, H., 1992. The Rise and Fall of Strategic Planning. *Long Range Planning,* 25(4), pp. 99-104.

Mintzberg, H. & Waters, J., 1985. Of strategies, deliberate and emergent. *Strategic Management Journal,* 6(3), pp. 257-272.

Moselhi, O., Fazio, P. & Hegazy, T., 1993. DBID: Analogy-based DSS for bidding in construction. *Journal of Construction Engineering and Management,* 119(3), pp. 466-479.

Moselhi, O., Hegazy, T. & Fazio, P., 1991. Neural networks as tools in construction. *Journal of Construction Engineering and Management,* 117(4), pp. 606-625.

Nobeoka, K. & Cusumano, M. A., 1995. Multiproject strategy, design transfer and project performance: A survey of automobile development projects in the US and Japan. *IEEE Transactions on Engineering Management,* 42(4), pp. 397-409.

Nobeoka, K. & Cusumano, M. A., 1997. Multiproject strategy and sales growth: The benefits of rapid design transfer in new product development. *Strategic Management Journal,* 18(3), pp. 169-186.

Pajares, J. & López, A., 2014. New methodological approaches to project portfolio management: The role of interactions within projects and portfolios. *Procedia: Social and Behavioral Sciences,* Volume 119, pp. 645-652.

Pedregosa, F. et al., 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research,* Volume 12, pp. 2825-2830.

Peffers, K. et al., 2006. The design science research process: A model for producing and presenting information systems research. *DESRIST International Conference on Design Science Research in Information Systems and Technology,* Volume 1, pp. 83-106.

Perks, H., 2007. Inter-functional integration and industrial new product portfolio decision making: Exploring and articulating the linkages. *Creativity and Innovation Management,* 16(2), pp. 152-164.

Phillips, J. J. & Phillips, P. P., 2006. Return on Investment Measures Success. *Industrial Management,* 48(2).

PMI, 2008. *A Guide to the Project Management Book of Knowledge.* 4th ed. Pennsylvania: Project Management Institute.

Porter, M. E., 1979. How competitive forces shape strategy?. *Harvard Business Review,* Volume 57, pp. 137-145.

Prencipe, A. & Tell, F., 2001. Inter-Project Learning: Processes and Outcomes of Knowledge Codification in Project-Based Firms. *Research Policy,* Volume 30, pp. 1373-1394.

Provost, F. & Fawcett, T., 1997. Adaptive fraud detection. *Journal of Data Mining and Knowledge Discovery,* 1(3), pp. 291-316.

Provost, F., Fawcett, T. & Kohavi, R., 1998. The case against accuracy estimation for comparing induction algorithms. *In Proceedings of the Fifteenth International Conference on Machine Learning,* pp. 445-453.

Purnus, A. & Bodea, C.-N., 2014. Project prioritization and portfolio performance measurement in project-oriented organizations. *Procedia: Social and Behavioral Sciences,* Volume 119, pp. 339-348.

Quinlan, J. R., 1986. Induction of decision trees. *Machine Learning,* 1(1), pp. 81-106.

Quinlan, J. R., 1988. An empirical comparison of genetic and decision tree classifiers. *Proceedings of the Fifth International Conference on Machine Learning,* pp. 135-141.

Quinlan, J. R., 1993. *C4.5: Programs for machine learning.* San Francisco: Morgan Kaufmann Publishers Inc.

Quinlan, J. R., 1996. Bagging, Boosting, and C4.5. *Proceedings of the Thirteenth National Conference on Artificial Intelligence,* pp. 725–730.

Rad, F. & Rowzan, S., 2018. Designing a hybrid system dynamic model for analyzing the impact of strategic alignment on project portfolio selection. *Simulation Modelling Practice and Theory,* 89(1), pp. 175-194.

Rahman, H. & Rahman, A., 2019. Strategic fit: model development and fitness analysis of a manufacturing unit. *Production & Manufacturing Research,* 7(1), pp. 44-66.

Rokach, L. & Maimon, O., 2014. *Data mining with decision trees: Theory and applications.* 2nd ed. Singapore: World Scientific Publishing Co. Pte. Ltd..

Sahin, Y., Bulkan, S. & Duman, E., 2013. A cost-sensitive decision tree approach for fraud detection. *Expert Systems with Applications,* 40(15), pp. 5916-5923.

SEC, 2003. *Securities and Exchange Commission - Final Rule: Amendments to Investment Company Advertising Rules.* [Online] Available at: https://www.sec.gov/rules/final/33-8294.htm [Accessed 6 12 2019].

Sheng, V. & Ling, C., 2006. Thresholding for making classifiers cost-sensitive. *Proceedings of the 21st National Conference on Artificial Intelligence,* Volume 1, pp. 476-481.

Smithson, M. & Verkuilen, J., 2006. *Fuzzy Set Theory: Applications in the social sciences.* London: Sage.

Sung, T. K., Chang, N. & Lee, G., 1999. Dynamic of modeling in data mining: Interpretive approach to bankruptcy prediction. *Journal of Management Information Systems,* 16(1), pp. 63-85.

Szymański, P., 2017. Risk management in construction projects. *Procedia Engineering,* Volume 208, pp. 174-182.

Talantsev, A. & Sundgren, D., 2013. Evaluating strategic fit of projects: a fuzzy linguistic approach. *Group Decision and Negotiation,* pp. 449-461.

Thompson, J. D., 1967. *Organizations in Action.* 1st ed. New York: McGraw-Hill.

Ting, K. M., 1998. Inducing cost-sensitive trees via instance weighting. *In Proceedings of the Second European Symposium on Principles of Data Mining and Knowledge Discovery,* pp. 23-26.

Tkáč, M. & Lyócsa, S., 2010. On the evaluation of Six Sigma projects. *Quality and Reliability Engineering International,* 26(1), pp. 115-124.

Turney, P. D., 1995. Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. *Journal of Artificial Intelligence Research,* Volume 2, pp. 369-409.

Vapnik, V. N., 2000. *The nature of statistical learning theory.* 2nd ed. New York: Springer.

Venable, J., Pries-Heje, J. & Baskerville, R., 2012. A comprehensive framework for evaluation in design science research. *Design Science Research in Information Systems,* Volume 7286, pp. 423-438.

Venkatraman, N., 1989. The concept of fit in strategy research: Toward verbal and statistical correspondence. *Academy of Management Review,* 14(3), pp. 423-444.

Wensley, R., 1981. Strategic marketing: Betas, boxes, or basics?. *Journal of Marketing,* Volume 45, pp. 173-181.

Wheelwright, S. C. & Clark, K. B., 1992. Creating project plans to focus product development. *Harvard Business Review,* 70(2), pp. 70-82.

Wolpert, D. H., 1995. On the Bayesian "Occam Factors" argument for Occam's Razor. *Computational Learning and Natural Learning Systems,* Volume 3.

Yescombe, E. R., 2002. *Principles of project finance.* San Diego: Academic Press.

Zhang, T. & Yu, B., 2005. Boosting With Early Stopping: Convergence and Consistency. *The Annals of Statistics,* 33(4), pp. 1538-1579.

7 APPENDICES

7.1 THE ALGEBRA FOR NOTATIONS

It is useful to first briefly cover some common notations that will be used throughout the paper to help the reader along the way.

The target attribute of the model will be marked with y and it has a finite set of possible values, often referred to as its classes or labels. All the possible values that the label y (also called target variable) can take are called its label space, or domain and denoted with $\mathcal{Y}$. The domain of target variable y with n number of classes can be written as $C = dom(y) = \{c_1, c_2, \dots, c_n\}$. In this thesis the label y has two possible outcomes $dom(y) = \{True, False\}$ and thus is a binary target variable. Likewise, every feature x (also called predictor or attribute) in the feature matrix with d number of features (or dimensions) each have their respective domains $dom(x_i) = \{x_{i,1}, x_{i,2}, \dots, x_{i,|dom(x_i)|}\}$, where $|dom(x_i)|$ stands for the *cardinality*, or the distinct count of possible values of the feature. The feature vectors $\boldsymbol{x}$ form the feature matrix $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_d)^T$.

By imagining every possible combination that the values of the predictor attributes are able to form, i.e. taking the *Cartesian product*, it is possible to define the instance, or feature space $\mathcal{X} = dom(x_1) \times dom(x_2) \times \dots \times dom(x_d)$. The universal instance space (or the labelled instance space) is then calculated as the Cartesian product of all the predictor attribute domains and the target attribute domain $\mathcal{U} = \mathcal{X} \times \mathcal{Y}$. In the context of this paper the instance space $\mathcal{X}$ defines every possible combination the project master data can take excluding the end result of the tendering competition. The universal instance space $\mathcal{U}$ defines every possible combination the values of the project master data can take including the target attribute y as well.

$\mathbb{X}^{train}$represents a training set consisting of a set of training instances $\{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$ that were drawn from the labelled instance space $\mathcal{U}$. One instance $(\boldsymbol{x}^{(i)}, y^{(i)})$ in this thesis represents one project where $\boldsymbol{x}^{(i)} \in \mathcal{X}$ and $y^{(i)} \in \mathcal{Y}$.

Finally, a hypothesis $\mathcal{h}$ (also called predictor or classifier) represents an assumed relationship between the feature space $\mathcal{X}$ and the label space $\mathcal{Y}$. The hypothesis space $\mathcal{H}$ then covers all the valid hypotheses that can be used to map a feature space to the label space denoted as $h : \mathcal{X} \to \mathcal{Y}$. Finding the optimal hypothesis is the goal of the classification problem in this study.

7.2 Performance Prediction with Decision Trees

An algorithm that can produce decision trees based on a set of data is called a *decision tree inducer* or *learner*. The distinction between the classification and regression models is particularly important for decision tree algorithms as not all of them can handle target attributes that are continuous. A decision tree that has continuous values as its output is called a *regression tree* whereas a one that has a discrete output is called a *classification tree*. As this thesis aims to predict whether a project will be won or lost based on its attributes it demands for a classification tree.

A classification tree is formed by a sequence of decision nodes that are connected by branches starting from the root node and extending to multiple terminal leaf nodes to form a complete tree. In the beginning all the records are grouped together and then according to the *splitting criteria* the optimal input attribute is chosen to split the records into two or more child nodes. The process is then repeated for each child node until one of the *stopping criteria* is met (Larose, 2005). Whether accuracy or generalizability is emphasised, the tree can be modified with different *ensemble* methods and by *pruning.* However, depending on the algorithm, not all methods are available for use. As stopping criteria, ensembles and pruning affect each other and aim to influence the same balance between accuracy and generalizability, they must be determined in unison.

This chapter will cover the essential components for inducing accurate and generalisable decision trees. For this thesis the most relevant elements are the ones that work as the backbones for *Scikit-learn's DecisionTreeClassifier (DTC)* algorithm as its different hyperparameter settings will be compared against each other. Specifically, the different splitting criteria, stopping criteria and ensemble methods were experimented with in Section 3.1.3 and therefore this section will focus on the theory behind these settings. Moreover, different ways to evaluate the performance of classifiers will be explained in detail as that is essential when selecting the optimal predictor.

7.2.1 Splitting Criteria

The splitting criteria dictate which attribute(s) will be used to split the data, where to set the threshold value and how many child nodes should be created. Large majority of the various splitting functions are univariate in the literature. This means that they only use one attribute to perform the split instead of using some function based on multiple attributes (though, multivariate splitting functions also exist). It is important for the data scientist to understand how the splits are actually made with different algorithms as different splitting criteria have different weaknesses and strengths. The relevant ones for this thesis will be covered next.

7.2.1.1 Gini Impurity, Twoing Criterion and GiniGain

Gini Impurity or *Twoing Criterion* are the basis for the CART algorithm (Classification and Regression Trees) and are also included as hyperparameter settings for Scikit-learn's DTC algorithm. Twoing Criterion is often used instead of Gini in case the domain of the target variable is wide. This is due to the latter's tendency to shift towards uneven splits with such target variables, which can reduce the algorithm's generalizability. However, if the target variable is binary, as is the case in this thesis, the results from Twoing Criterion and Gini Impurity are equal. (Breiman, et al., 1984)

Gini impurity measures how often a randomly chosen record from a dataset would be wrongly labelled. It can simply be calculated as:

$$\text{Gini}(\mathbb{X}) = 1 - \sum_{c \in C} p(c)^2 \tag{7.1}$$

in which, C denotes the set of all the possible classes of the target variable y and $p(c)$ describes the proportion of class c in the set $\mathbb{X}$.

The change in impurity after implementing a split can be calculated by subtracting the impurity in the parent node by the sum of weighted average impurities in the child nodes. Formally, it can be defined as:

$$GiniGain(x, \mathbb{X}) = Gini(\mathbb{X}) - \sum_{t \in T} \left(\frac{|t|}{|\mathbb{X}|} \cdot Gini(t) \right) \tag{7.2}$$

In which T includes all the child nodes that were created after splitting the data with the feature x such that $\mathbb{X} = \bigcup_{t \in T} t$. Then $\frac{|t|}{|\mathbb{X}|}$ is the percentage of data in a child node t and $Gini(t)$ is the Gini impurity measure in that node. A split that maximizes the Gini gain should be selected as the threshold when using Gini as the splitting criterion.

7.2.1.2 Entropy, Information Gain and Information Gain Ratio

Like the Gini measures, *Information Gain* is also an impurity-based criterion that utilises entropy to measure the effectiveness of the split. The second option that Scikit-learn's DTC algorithm offers for the splitting criteria is Entropy, which also produces binary splits by setting a threshold value that maximizes the *Information Gain* (Pedregosa, et al., 2011). The ratio is derived from the *Information Entropy* which can be formally defined as:

$$Entropy(\mathbb{X}) = 1 - \sum_{c \in C} p(c) \cdot \log_2 p(c) \tag{7.3}$$

in which, $p(c)$ describes the probability of a correct prediction in the node like in the Gini Impurity formula.

Information gain follows the same logic as the Gini gain. It describes the change in impurity after the split is done and is calculated by subtracting the entropy in the parent node by the sum of weighted average entropies in the generated child nodes. Formally it is defined as:

$$InformationGain(x, \mathbb{X}) = Entropy(\mathbb{X}) - \sum_{t \in T} \left(\frac{|t|}{|\mathbb{X}|} \cdot Entropy(t) \right) \tag{7.4}$$

In which $\frac{|t|}{|\mathbb{X}|}$ is the percentage of data in one child node and $Entropy(t)$ is the entropy measure in that specific node. As one can see, the only difference between the Gini impurity and Entropy formulas is the additional $\log_2 p(c)$ multiplier. Universally one is not better than the other, so both were experimented in this thesis, although difference in performance between the two is usually quite minimal.

7.2.2 Stopping Criteria

Algorithms often let their user to control the stopping criteria that dictates the stopping of decision tree's induction phase. The following rules are commonly used to stop the tree induction as defined by Rokach & Maimon (2014).

(1) Entropy of zero is reached in a leaf and thus the leaf is homogenous i.e. all instances in a leaf belong to the same class of target attribute y

(2) The threshold value for maximum tree depth is reached

(3) The threshold value for minimum number of instances in a parent node is reached in a terminal node

(4) The threshold value for minimum number of instances in a child node would be breached in at least one generated child node, if the parent node were split

(5) The threshold value of the splitting criterium cannot be reached

There is a trade-off between the accuracy and generalisability of the tree, when it comes to defining the stopping rules. By setting the threshold values very strictly to ensure generalizability of the decision tree, often the model turns out to be underfitted. Conversely, loose stopping criteria allows the tree to grow wide and deep resulting in an overfitted model (Rokach & Maimon, 2014).

Breiman et al. (1984) took on to solve this controversy by developing a method called *pruning*. They suggested that first employing loose stopping criteria to grow an overfitted model based on the training set and then removing sub-branches of the tree that do not contribute to the generalization accuracy of the model would result in a simpler and more accurate tree. Pruning increases the generalizability especially when the initial data set is noisy (Breiman, et al., 1984).

In the performance prediction phase described in Section 3.1.3 the rule number 4 was experimented through the hyperparameter settings in an attempt to minimize the generalization error. The stopping criteria are also essential when a classifier is being wrapped in a boosting wrapper as it tends to converge very fast and easily results in an overfitted classifier. Unfortunately, pruning was not supported in the DCT function version, which was used for the purposes of the thesis.

7.2.3 TREE ENSEMBLES

The final method to be covered in this decision tree induction section are *ensembles* that are useful for improving both the accuracy and the generalizability of the tree. Ensembles have become more and more valid for the machine learning community during the past couple decades as the computing power of computers has increased exponentially and thus, computing time is less of a concern.

Bagging as described by Breiman (1996) and *boosting* by Freund and Schapire (1996) are the two most common methods to form tree ensembles out of individual decision trees. Both create a series of weak classifiers that together form one strong classifier. They implement a *voting* method to decide on the final prediction, though slightly differently as bagging handles the votes of each classifier with the same weight, whereas boosting weights the votes based on the accuracy of each classifier. Furthermore, bagging is said to be an independent method for ensembles as the classifiers can be ran in parallel and are not affected by each other. Boosting, on the other hand, utilises the predictions of the previous classifier in the next iteration and as such is described as a dependent method for generating ensembles (Quinlan, 1996).

Both methods are very convenient in machine learning projects as they can be employed to almost any algorithm. The trade-off of implementing these techniques is that they reduce the interpretability of the final output due to the increased complexity of the final classifier. However, these methods are always worthy of experiment as the accuracy and the stability of the model can substantially increase by employing either of them.

The premise is that due to the greedy nature of the decision trees, they are vulnerable to randomness in data such as errors and outliers. A small change in the values of the predictor attributes can change the composition of the whole tree and thus, decrease or increase its performance depending on chance. This randomness can be exploited though; by combining individual trees that often have a small bias, but large variance with their predictions, the variance can be effectively eliminated, because an average figure derived from multiple classifiers evens it out. This notion is very important for this study as the source data was generated by users their selves and thus is highly likely to include inaccuracies. The two ensemble algorithms were experimented with in Section 3.1.3 and will be introduced in this final chapter of decision tree induction.

7.2.3.1 Bagging

Bagging i.e. bootstrap aggregating is a method to increase the accuracy and stability of the model. Bagging requires that the original classifier is instable in nature. It injects randomness to the model by sampling the training sets for the independent classifiers with replacement. If the randomness caused by the sampled data sets can cause significant changes to the induced classifiers, the overall accuracy and stability of the model can be improved. However, Brieman (1996) also noted that if the initial classifier is very poor, bagging can also worsen the results.

Formally, bagging is implemented so that for each classification tree $T = \{T_1, T_2, \dots, T_k\}$, a training set equal to the original size of the training instances N is uniformly sampled by using the replacement method. Thus, the expected outcome for sampling the training sets from a uniform distribution is that some instances won't be included in the training sets at all and some will contain duplicates of the same instance. After the sampling stage, the decision tree inducers are trained with their designated training sets to form k number of classifiers $\hbar_T$. In order to classify a label y of a previously unseen instance $\boldsymbol{x}$, the final classifier $\hbar^*$ will aggregate the results gained by feeding the instance into each of the classifiers $\hbar_T$, and then letting them vote for which class the instance should be classified. Each classifier $\hbar_T$ will vote with an equal weight and the class with the most votes wins. Formally the bagging function is written as

$$\hbar^*(\boldsymbol{x}) = sign\left(\sum_{T=T_1}^{T_k} \hbar_T(\boldsymbol{x})\right). \tag{7.5}$$

7.2.3.2 Boosting

Like bagging, boosting is also used to increase the accuracy of the classification model, though some studies indicate that boosting can result in a higher variation in performance than bagging (Quinlan, 1996). Freund and Schapire (1996) introduced the *Adaptive Boosting* (*AdaBoost*) algorithm that has become the industry standard for machine learning. It is based on producing a sequence of classifiers, which provide different weights to the predictions of instances to reflect the accuracy of each prediction. For misclassified instances the weight is larger than for correct predictions, which forces the next classifiers to focus on the wrong predictions.

For a binary target variable y, let k be the number of iterations in inducing decision trees $T = \{T_1, T_2, \dots, T_k\}$, $\hbar_T$ the induced classifier and $\omega_T^{(i)}$ be the weight assigned to the i-th instance $\boldsymbol{x}^{(i)}$. For the first tree T_1, there are N instances in the training set and the weights for every instance are assumed to be equal $\omega_{T_1}^{(i)} = \frac{1}{N}$. At each subsequent iteration, the weights are treated as if they formed a proper distribution, i.e. $\sum_{i=1}^{N} \omega^{(i)}$ equals one and as, if they described the probability of instance $\boldsymbol{x}^{(i)}$'s occurrence. The misclassification rate ε_T is also calculated with respect to the weights as a sum of the weights that have been misclassified

$$\varepsilon_T = \sum_{\substack{i=1 \\ \hbar_T(\boldsymbol{x}^{(i)}) \neq y^{(i)}}}^{N} \frac{1}{\omega_T^{(i)}}. \tag{7.6}$$

After the misclassification error is calculated, the weights of each correctly classified instance $\omega_T^{\hbar_T(\boldsymbol{x}^{(i)}) = y^{(i)}}$ is recalibrated by multiplying the previous weight with parameter $\propto_T = \frac{1}{2} \ln\left(\frac{1-\varepsilon_T}{\varepsilon_T}\right)$ to get the next weight $\omega_{T+1}^{(i)}$. As $\propto_T$ is always less than one, after each calibration the weight of correctly classified instance is decreased to force the upcoming classifiers to focus on the misclassified instances. Algorithm continues the iterations until the maximum number of iterations T_k is reached, a misclassification rate for a classifier becomes greater than 0.5 or reaches 0.

To classify an unseen instance $\boldsymbol{x}^{(i)}$, the instance is fed to the ensemble of trees induced with the training set and the trees will vote for its class. Each tree will have a voting power equal to the $\propto_T$ determined earlier. Formally, the boosting function can be written out as

$$\hbar^*(\boldsymbol{x}^{(i)}) = sign\left(\sum_{T=T_1}^{T_k} \propto_T \cdot \hbar_T(\boldsymbol{x}^{(i)})\right). \tag{7.7}$$

Even though it is not constrained by a rule, boosting requires that the predictive power of the classifier is better, even just slightly, than a random classifier. If the misclassification rate would be exactly 0.5, the parameter $\propto_T$ would equal to zero and as such the weight of the predictions made by the classifier would all be zero as well.

7.3 Evaluation of Classification Trees

The goal in the decision tree induction is to induce such a model that can with as high accuracy as possible predict the value of the target attribute correctly based on a set of predictor attributes. Depending on the balance of the data set, multiple different evaluation metrics can be useful in determining whether the model is performing on a satisfactory level. These metrics will be covered in this chapter.

7.3.1 Defining the Generalization Error

The measure to describe the model's capability to perform a task is called *generalization error* or *misclassification error*. Generalization error in supervised learning essentially describes the model's ability to correctly assign values for a target variable for previously unseen data. It can be minimized by avoiding overfitting in the model - that is to make the model as general as possible. As defined by Rokach & Maimon (2014), for nominal target variables and classifier $\mathscr{h}$ it can be written out as:

$$\varepsilon(\mathscr{h}) = \sum_{(\boldsymbol{x},y)\in\mathcal{U}} P(\boldsymbol{x},y)\cdot\mathcal{L}(\mathscr{h}(\boldsymbol{x}),y), \tag{7.8}$$

in which, the $\mathcal{L}(\mathscr{h}(\boldsymbol{x}),y)$ is the 0-1 loss function defined as:

$$\mathcal{L}(\mathscr{h}(\boldsymbol{x}),y) := \begin{cases} 0\ if\ \mathscr{h}(\boldsymbol{x}) = y \\ 1\ if\ \mathscr{h}(\boldsymbol{x}) \neq y \end{cases}. \tag{7.9}$$

In the previous notation, $\mathscr{h}$ represents a classifier and $\mathscr{h}(\boldsymbol{x})$ a prediction that was achieved by feeding an input feature vector $\boldsymbol{x}$ drawn from the universal instance space $\mathcal{U}$ to the classifier $\mathscr{h}$. Therefore,

1. $\varepsilon(\mathscr{h})$ = *generalization or misclassification error of classifier* $\mathscr{h}$
2. $P(\boldsymbol{x},y)$ = *a joint probability distribution for* $\boldsymbol{x}$ *and* y
3. $\mathcal{L}(\mathscr{h}(\boldsymbol{x}),y)$ = *0-1 loss function that results in 1 if prediction is incorrect and 0 if it is correct.*

Thus, generalization error for a classification task is simply the summation of all the probabilities of the predictions that were missclassified by the decision tree i.e. its general probability to misclassify a target attribute y based on feature vector values $\boldsymbol{x}$. However, due to the fact that the distribution $P(\boldsymbol{x},y)$ is often unknown (unless the data set was synthetically generated), it can be impossible to calculate the generalization error precisely. Fortunately, there are several ways to estimate it.

7.3.2 Estimating the Generalization Error

For a classification tree, its *classification accuracy* is defined as one minus the generalization error. As previously noted, the generalization error is often impossible to calculate precisely, if the distribution $P(\boldsymbol{x}, y)$ is unknown. A good approximation of the generalization error can be empirically derived though, which is why *training error* is can be used instead. Training error tells the percentage of records that the classification tree was able to classify correctly from the training set. It is defined as

$$\hat{\varepsilon}\left(\mathscr{h}, \mathbb{X}^{train}\right) = \frac{1}{|\mathbb{X}^{train}|} \sum_{(\boldsymbol{x},y)\in\mathbb{X}^{train}} \mathcal{L}(\mathscr{h}(\boldsymbol{x}), y), \tag{7.10}$$

where $|\mathbb{X}^{train}|$ is the number of records in the training set $\mathbb{X}^{train}$ and $\mathcal{L}(\mathscr{h}(\boldsymbol{x}), y)$ is the 0-1 loss function defined earlier.

However, the training error is not without flaws either. It will typically provide an over-optimistic figure of the generalization error especially, if the algorithm is prone to *overfitting.* Fortunately, generalization error can be estimated either theoretically or empirically, which should be used instead to get a more unbiased measure for the accuracy.

Theoretical estimation utilises the fact that there is often a trade-off between the training error and the confidence assigned to the training error to predict the generalization error. The capacity of the inducer, i.e. its ability to produce different inducers, plays a major role in determining the accuracy of the decision tree. Often the number of nodes in a decision tree correlates negatively with the training error as the model shapes itself more closely to the training set when the number of nodes increases and begins to overfit. Large number of nodes relative to the size of the training set, might indicate that the decision tree is only memorizing the patterns of the training set and hence wouldn't be accurate on novel data. Theoretical frameworks include e.g. VC-Dimension (Vapnik, 2000) and Bayesian (Wolpert, 1995), which are basically formed by first calculating the training error and then adjusting it with some penalty function to simulate the capacity of the inducer.

Another, more practically oriented approach is to empirically estimate the generalization error. Here a completely labelled dataset is split to a training and test set. First, the training set is used to induce a suitable classification tree and then the misclassification rate is measured from the test set. The acquired measure is calculated exactly as the training error defined earlier but instead of using the training set $\mathbb{X}^{train}$ the test set $\mathbb{X}^{test}$ is used instead. A large difference between the accuracy of the predictions with the training dataset and the test set is an indicator of an overfitting issue. The misclassification rate of the test set also represents a more accurate value for the generalization error than that of the training set.

7.3.3 Increasing the Confidence of the Generalization Error

If the overall amount of data is small, the confidence in the generalization error estimated with just one small test set may be low. In such cases, a common way to increase the confidence of the accuracy measure is to re-sample the data into different groups in different ways and perform multiple tests. This thesis adopts the k-fold cross-validation method for resampling, and it is specifically used within the GridSearchCV function during the hyperparameter optimization phase covered in Section 3.1.3.

In k-fold cross-validation the data is first randomly split into k number of mutually exclusive subsets that are approximately equal in size. Then the inducer is trained with k - 1 folds and tested with the remaining one subset. This process is repeated so that each of the k-folds are used for testing, and finally an average of the results can be taken to get a single estimation for the generalization error. To make the results between the tests more stable, especially for unbalanced datasets, a stratified k-fold cross-validation is often used. It modifies the original method so that it ensures a similar distribution of the target classes between the k-folds and the original dataset.

7.3.4 Crisp and Probabilistic Classifiers

A classifier that can explicitly assign a certain class to an unseen instance is called a *crisp classifier* and one that is able to produce probability measures is called a *probabilistic classifier*. In this thesis both are relevant. A crisp classifier without any context about the certainty of the prediction can be hard to rely on. However, a crisp classifier is useful to divide the data into different groups as was done when the Strategy-Success matrix was formed and used in Section 4. Probabilistic classifier on the other hand was used to calculate the relative scores for the simulation in Section 3.3.1 in order to simulate the selections of the tool.

For classification trees the probability is simply calculated as the frequency of the predicted class among all the predictions in one leaf. E.g., if one leaf contained 10 instances of "win" projects and 0 instances of "lost" projects, the prediction for an unseen instance that was classified into that leaf would be “win” with a probability of 1. However, it is generally agreed, according to the Cromwell's rule, that only events that are logically true or false should have probabilities of 1 or 0 (e.g. 1+1=2 or 1+1=3) (Rokach & Maimon, 2014). Therefore, a prediction with predicted probability of 1 as in our previous example will typically be an over-estimation. As decision trees are greedy and unstable classifiers, it is not rare to see leaves with zero entropy. This issue has to be tackled by utilising the stopping criteria, or by inducing randomness to the classifier in the induction phase with e.g. the bagging method covered in 7.2.3.1.

7.3.5 Other Accuracy Measures

Let's say, a company would win only 1% of the tendering competitions on average. Here a classifier that classifies every instance of the test dataset as "lost" would gain 0.99 probability to classify the result correctly. If the target class has very imbalanced distribution as in our previous example, the generalization error is not a sufficient measure for evaluating the performance of the model. In such cases *sensitivity* (or *recall*), *specificity* and *precision* are appropriate measures.

Sensitivity describes how well the model recognizes positive samples and is defined as:

$$Sensitivity = \frac{count\ of\ true\_positive}{count\ of\ positive}. \tag{7.11}$$

Specificity describes how well the model classifies negative samples and is defined as:

$$Specificity = \frac{count\ of\ true\_negative}{count\ of\ negative}. \tag{7.12}$$

On top of these, Precision is often used to measure what percentage of the measures that are classified as "positive" are actually "positive". Formally it is defined as:

$$Precision = \frac{count\ of\ true\_positive}{count\ of\ positive + count\ of\ false\_positive}. \tag{7.13}$$

The precision score is the same evaluation metric as *"hit rate"*, which was the primary performance measure used by the case company as well as the criterion used to select the best classifiers and evaluate the performance of the simulation with regards to likelihood of success.

A detailed reasoning for selecting precision (or hit rate) as the primary evaluation metric to measure the predictions is covered in the main part of the thesis in Section 3.2.2.2.

It is often useful to draw the true positive (TP), false positive (FP), true negative (TN) and false negative (FN) counts as a matrix to visualize and calculate the different rates of accuracy. It can be easily done with a Confusion Matrix that plots all the counts as a handy table. It shows in the main-diagonal line (A and D) the instances that have been correctly predicted as either positive or negative and on the off-diagonal line (C and B) the instances that were wrongly predicted as either positive or negative.

Table 19: Cross-tabulation of classification results

	Predicted negative	*Predicted positive*
Actual negative	A	B
Actual positive	C	D

Out of the above-mentioned table, the following rates can be calculated:

- Accuracy $= \frac{a+d}{a+b+c+d}$
- Misclassification rate $= \frac{b+c}{a+b+c+d}$
- Precision $= \frac{d}{b+d}$
- True positive rate (Sensitivity / Recall) $= \frac{d}{c+d}$
- False positive rate $= \frac{b}{a+b}$
- True negative rate (Specificity) = $\frac{a}{a/b}$
- False negative rate $= \frac{c}{c+d}$.

7.3.6 Evaluation of Classifiers

The previously covered measures may be enough to evaluate and rank the classifiers in the correct order, but they do not take into account the variation in performance and confidence with different probability threshold levels. *Receiving Operating Characteristics (ROC)* curve and the *Area Under the Curve* (AUC) can be used to estimate these more subtle aspects of classifiers by altering the confidence of the predictions from 1 to 0.

7.3.6.1 Receiving Operating Characteristics (ROC)

ROC curve demonstrates the dynamics between the true positive (TP) and false positive (FP) rates of the classifier as the probability measure generated by a probabilistic classifier is varied from higher threshold values to the lower ones. This is convenient as the location of a point in the curve describes its performance with a given threshold. The optimal point in a ROC curve would be $(FP, TP) = (0,1)$, which would indicate that the model is able to classify every positive instance correctly before doing any mistakes by classifying negative ones as positive.

Often the classifiers are not consistent in the sense that the steepness of the curve would increase and decrease in a stable manner. Instead there might be certain optimal spots along the ROC curve in which the TP to FP rate is maximized. Some classifiers might be better with more predictions, while some may have very high TP rate with high probability threshold values but lose predictive power quickly as the probability threshold is lowered. Therefore, the optimal classifier for each individual task might actually depend on the desired threshold level instead of the overall generalization error.

7.3.6.2 *Area Under the Curve* (AUC)

Area under the curve is often computed to rank different models based on their ROC curves. AUC equals to the occupied area below the ROC curve. It can be interpreted to mean the probability that a uniformly drawn positive instance has a higher probability value than a uniformly drawn negative instance. In practice, $AUC = 0.5$ would indicate that the probability that a positive prediction would be true is completely random, and $AUC = 1.0$ that the classifier is always right on positive predictions.

All of the previously covered measures will be useful in determining the strength of the model in different situations. Because there is no ultimate measure that could be used to compare the strength of the models in every possible case, each of these measures have their use cases. In the empirical part of the thesis, the suitable measure is referred to when appropriate depending on the situation.

7.3.7 Considerations in the Usage of Decision Trees

Decision trees are not without flaws, though. Their behaviour is described to follow a "Divide and Conquer" method, that refers to the decision trees' approach to solve problems by dividing them into smaller subproblems, then solving them and finally combining the solutions of the subproblems to form a complete model. This makes them rather greedy as they solve the problems by finding a local optimum at each stage. These characteristics make decision trees perform worse the more relevant attributes there are in the dataset, and even more so, if there are multiple complex interactions between the attributes. This is derived from the fact that a large majority of decision tree algorithms, including Scikit-Learn's DTC, are univariate i.e. they perform splits based on only one attribute instead of a function of multiple attributes. Therefore, if the relationship between the target attribute and the input attributes is complex and based on some function of the input attributes, it would be hard for a univariate decision tree to perform well. It would most likely result in a replication problem, in which the subtrees would be duplicates of the previous splits as in Figure 2.

Decision trees can also be very sensitive to the composition of the training set, irrelevant attributes and noise in the data due to the previous characteristics (Quinlan, 1993). If for example, there would be an accidental relationship between a target attribute and an irrelevant input attribute close to the root node, the whole tree below that split would be affected. This issue must be addressed by prepping the data well and ensuring there are as little misinformation in the data as possible. Often missing values are not an issue, but actual errors and outliers may affect the outcome greatly. Also due to their greedy nature some decision tree algorithms tend to be prone to overfitting. This should be addressed by defining the stopping criteria carefully and experimenting with pruning and ensemble methods to improve the generalisability.

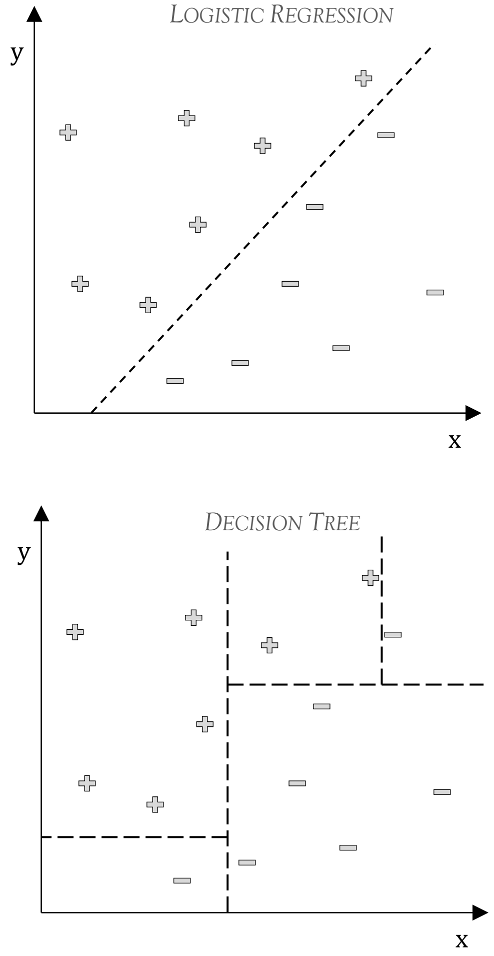


Figure 27: Logistic regression vs. decision tree classifier.
The relationship between the plus and minus signs seems to be split by a hyperplane y = x + c, and therefore e.g. a logistic regression would suite this data nicely. If a decision tree would be deployed it would result in a much more complex model as a univariate decision tree algorithm can only perform splits that are perpendicular to the axes (e.g. x>2 or y<1).

7.4 EVALUATION OF STRATEGIC FIT WITH FUZZY LOGIC

In this section, a method will be described that can be used to incorporate qualitative linguistic evaluations to the strategic fit measure on top of the quantitative method used in this thesis. It must be noted however, that gathering linguistic evaluations from the evaluators is to some extent manual work and can be implausible in case the sample size is large due to the fact that every sample has to be evaluated separately by all the evaluators. The following method is based on Talantsev & Sundgren's (2013) paper with a minor modification. The difference comes from the fact that their framework did not define how to incorporate continuous variables in the fuzzy number aggregation step, though clear guidelines have already been specified on how that should be done (Smithson & Verkuilen, 2006). Table 20 outlines the steps in the strategic fit evaluation process.

Table 20: Method outline (adopted from Talantsev & Sundgren (2013))

Phase	Step
Preparation	Step 1. Form a group of evaluators (if evaluations are needed)
	Step 2. List top-level strategic goals
	Step 3. List identified projects
Evaluation	Step 4. Evaluate projects' strategic fit
Processing	Step 5. Transform continuous linguistic values into fuzzy numbers
	Step 6. Aggregation: Step 6.1. Aggregate individual evaluations for each goal-project pair Step 6.2 Aggregate goals' values on a project level
	Step 7. Defuzzification

The first three steps in their method largely follows the same steps as what was covered in the main thesis. In the s**tep 1,** the process begins with gathering a group of evaluators, if their opinions are needed in the evaluation. Often, this is found naturally from the organisation (e.g. their management team). **Step 2** consists of listing the top-level strategic goals against the projects under evaluation. In Talantsev & Sundgren's method only soft high-level strategic goals were used. The final **step 3** of the preparation phase all the projects that will be evaluated are identified and listed.

In the **step 4,** evaluation of the projects in contrast to the strategic goals will happen. An example of a soft goal could be e.g. "prioritize complex projects over simple ones to utilise our whole organisation" as the "complexity" of a project is a rather fuzzy term and hard to quantify accurately with a single measure. Talantsev & Sundgren suggested a linguistic scale to measure the strategic fitness of different responses from Chang et al. (2007), which is presented in Table 21. Note that the linguistic values can basically be determined by the researcher based on what is appropriate for the context of the evaluation.

Table 21: Linguistic values plotted along the strategic fits that they correspond (from Change et al. (2007))

Linguistic value	**Distinctive points**			
	a	**b**	**c**	**d**
No Fit	0	0	0	0.1
Very Low	0.1	0.15	0.25	0.3
Low	0.25	0.3	0.4	0.45
Medium	0.4	0.45	0.55	0.6
High	0.55	0.6	0.7	0.75
Very High	0.7	0.75	0.85	0.9
Perfect	0.9	1	1	1

In **step 5,** after all the required measures and evaluations have been gathered, they will be translated into fuzzy numbers using appropriate membership functions. There are no right or wrong answers in determining the membership functions as it really depends on the case and context of the question, as well as the distribution of the values (Smithson & Verkuilen, 2006). For the linguistic values Talantsev & Sundgren (2013) suggested the following membership function μ,

$$\mu(x,a,b,c,d) = \begin{cases} 0, x < a \ or \ x > d \\ \dfrac{x-a}{b-a}, a \leq x \leq b \\ \dfrac{d-x}{d-c}, c \leq x \leq d \end{cases} \tag{7.14}$$

Essentially equation (7.14) translates the distinctive points in Table 21 into a trapezoidal membership function with linearly increasing and decreasing degrees of memberships. The parameters a, b, c and d correspond to the threshold points of the membership function and are listed in the Table 21. The distinctive points are not necessarily set in stone, and they can be adjusted if the decision context so requires. The equation (7.14) with the distinctive points in Table 21 translate into the following graphical representation of the function.

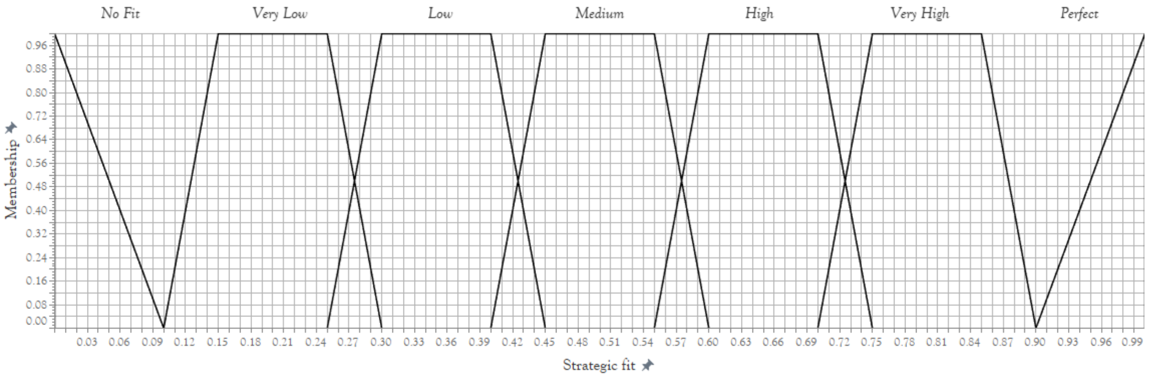


Figure 28: The membership function for the linguistic values.

In the following **step 6.1.,** after each linguistic evaluation has been transformed into a fuzzy set compromised of the four distinct points a, b, c, d, each project-goal pair will be aggregated per distinctive point. This can be done with a simple mean calculation. The result will be the average values for points (a, b, c, d) across the respondents for each project-goal pair.

In **step 6.2.,** with the average results for each project-goal pair, the goals will be further aggregated in order to derive one fuzzy value for each of the distinctive points per project. The output is a fuzzy number and can be calculated using the same simple mean method as in the previous step.

Step 7. Finally, the aggregated fuzzy numbers will be transformed into a crisp number using a defuzzification method. Mathematicians have proposed a plethora of defuzzification methods out of which the appropriate one should be decided on a case by case basis. A plausible general method could be to use the Mean of Maxima (MoM), which essentially takes the mean value from the aggregated distinctive points of the projects. Thus, the MoM method selects the most typical value as the final crisp output. For an example of employing linguistic fuzzy sets refer to the Talantsev and Sundgren (2013) study about the subject.