MSc thesis in Construction Management & Engineering

Virtual Assistant for maintenance budget estimation

G. Masah

VIRTUAL ASSISTANT FOR MAINTENANCE BUDGET ESTIMATION: USING MACHINE LEARNING TO IMPROVE THE OBJECTIVITY OF MAINTENANCE BUDGET ESTIMATES OF CIVIL ENGINEERING STRUCTURES

A thesis submitted to the Delft University of Technology in partial fulfillment of the requirements for the degree of

Master of Science in Construction Management and Engineering

by

G. Masah

December 2020, revised version

G. Masah: ***Virtual Assistant for maintenance budget estimation:*** *Using machine learning to improve the objectivity of maintenance budget estimates of civil engineering structures* (2020, revised version)

The work in this thesis was made in the:

Department of Construction Management & Engineering
Faculty of Civil Engineering & Geosciences
Delft University of Technology

Supervisors:	Prof.dr.ir. A.R.M. Wolfert	TU Delft
	Dr.ir. R. Binnekamp	TU Delft
	Dr.ir. O. Kammouh	TU Delft
Co-readers:	Ing. R. Mirck & Ir. B. Hoogzaad	BAM infra

ABSTRACT

With an increase of data documentation and standardization in the construction field in The Netherlands, by norms such as the NEN, there is a possibility to introduce data-driven approaches to certain areas within the construction industry. One of these is the area of budget estimation which is currently fully dependent on a cost estimating professional. Due to the need for estimations that are effective and time-efficient, especially in the primary phase of a project, the potential of introducing a data-driven approach is explored through this thesis. The main objective of this research is the development of a data-driven model, in the form of a Virtual Assistant (VA), to increase the objectivity of the estimation of maintenance budgets of civil engineering structures. From a literature study it is apparent that a fitting data-driven approach for the development of this model is the machine learning technique Decision Tree Classification (DTC). The VA model is developed using historical data, in the form of past input and past output, to train the model and therefore make predictions. Data that is used as past output for this model is a budget range which is documented as a budget class and data that is used as past input is data that is ensured to be objective and gives a description of each bridge. In this case the past input data are the characteristics of the bridge which refer mostly to the dimension of the bridge, the NEN2767, which captures the decomposition and condition of the bridge and to a lesser extent the duration of the maintenance. Through exploring past cases the machine learns rules and predicts the outcome for a new case and therefore predicts the budget range. This shows in which range the budget guess of the estimator should fall. Generally in order to develop a VA model and make it applicable for industry use it is important that an organization that uses such model aligns their data storage with the DTC methodology. This means the introduction of standardizing data in classes and the introduction of standard procedures to document the data. Only when these elements are present within the organization, the data that is used for past input and output can be regarded as objective and the VA can fulfill its function which is to verify the budget estimators guess in an objective manner.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

List of Algorithms

ACRONYMS

NEN Nederlandse Norm . . . 2
VA Virtual Assistant . . . ix
KBS Knowledge-Based Systems . . . 6
CI Computational Intelligence . . . 6
HS Hybrid Systems . . . 6
ML Machine Learning . . . 6
R2F Run-to-failure maintenance . . . 7
PvM Preventive Maintenance . . . 7
PdM Predictive Maintenance . . . 7
DTC Decision Tree Classification . . . 8
ABT Analytics Base Table . . . 15
LOD1 Level of Detail 1 . . . 34
LOD2 Level of Detail 2 . . . 34

1 INTRODUCTION

As mentioned in the abstract, this research focuses on the development of a data-driven model to increase the objectivity of the estimation of maintenance budgets of civil engineering structures. More about the background of this research follows in this chapter.

1.1 BUDGET ESTIMATION IN THE CONSTRUCTION INDUSTRY: BACKGROUND

Budget estimation is an essential component in the construction industry since it has a direct effect on the contractors' economic performance. Overestimation or underestimation may cause problems in business performance, i.e., overestimation may result in a negative public image of the contractor while underestimation will result in financial losses (Haroun [2015]).

Besides this, during the early project stage budget estimation needs to be performed within a limited time period using limited information in an uncertain environment. Estimating methods at this stage needs to be quick, inexpensive, and reasonably accurate (Kim et al. [2012]).

Therefore there is a need for estimations that are both effective and time-efficient. In every construction company the budget estimation is the job of the cost engineer/cost estimator. Construction and cost engineering professionals have long recognized the need for improvements in cost control (Humphreys [1991]). This research aims to help the budget estimation process by introducing a data-driven approach which intends to help make the estimates of cost estimators more objective and thus help combat overestimation or underestimation.

1.2 PROBLEM STATEMENT

At the moment budget estimates are based on the opinion of the cost estimator without a verification process. This results in estimations being not accurate enough and a possibility for deviations from the predicted budgets in the future. There is no objectivity in the current way of estimating project budgets, so this research will focus on introducing a systematic and data-driven approach to budget estimation which helps make estimates more objective.

1.3 DEVELOPMENT GAP

Current Practice: The accuracy of a budget estimate relies on the level to which the estimator defines a project; the experience and skill of the estimator; the level of accuracy of the used tools and references used to make an estimation.

Gap: Ambiguity in the definition of the project level; high dependence on expert-opinion; ambiguity in the tools and references used by estimator.

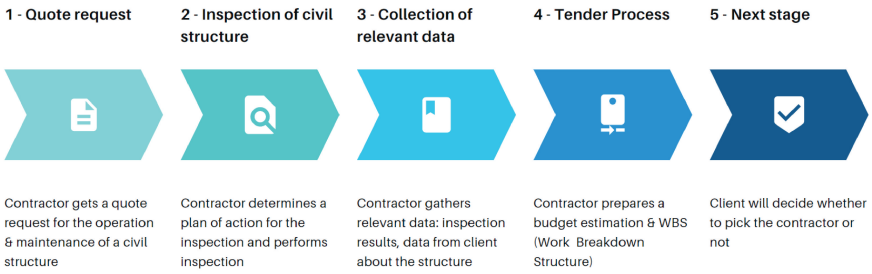


Figure 1.1: Process Quote to Tender

1.4 SCOPE

Budget estimation in construction encompasses a large field of different types of applications. This thesis focuses on the *maintenance* budget estimations because of its large potential of introducing a data-driven approach to, especially in the context of the Netherlands. The general method for i.a. maintenance budget estimation by contractors, as defined by BAM, is depicted in Fig. 1.1 and is consists of the following steps:

1. **Quote request:** The contractor receives a quote request from the client accompanied by documents regarding the decomposition of the elements of the civil structure according to the NEN2767 (Dutch normative inspection). Also previous reports of budget and inspection may be included.
2. **Inspection of civil structure:** The contractor performs a *nulmeting* (initial inspection) to understand the condition of the civil structure and determine further steps. In this initial inspection a condition-number is given to each element of the structure to grade the state.
3. **Collection of relevant data:** All of the previous documents and condition data will be collected to paint a picture of the current state of the civil structure.
4. **Tender Process:** Hereafter the tender process starts, where the ***budget is estimated*** for future maintenance and operation of the civil structure. Also the work that needs to be performed on the structure is defined.
5. **Next stage:** Finally the client decides whether to proceed with this contractor or another one based on the tender documents.

As previously mentioned there is a large potential to introduce a data-driven approach to budget estimation of maintenance budgets, especially in the context of The Netherlands. The main reason for this is that there already exists a standardization of inspection processes, in the form of the Dutch Normative or Nederlandse Norm (NEN) (more in Chapter 4.4.2). This NEN-norm decomposes a civil structure to its building parts in a standardized manner and therefore it makes it easier to compare different civil structures to each other. Due to this standard way of documenting inspection and cost data, there is a potential to automate this process and introduce a data-driven approach.

1.5 RESEARCH OBJECTIVE

Fig. 1.2 shows the old method of budget estimation and Fig. 1.3 shows the new method of budget estimation. The old method uses only the expert guess to come to a budget estimation. This expert guess is based on condition data retrieved

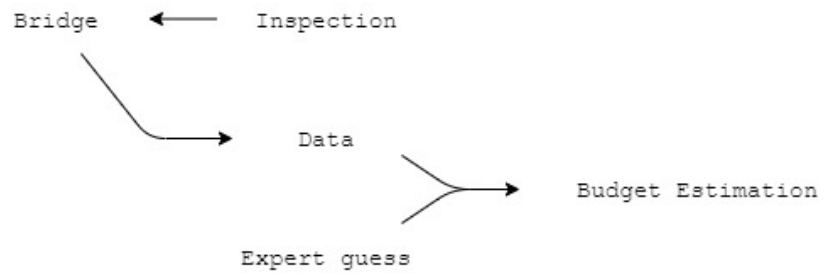


Figure 1.2: Old method of estimating budgets

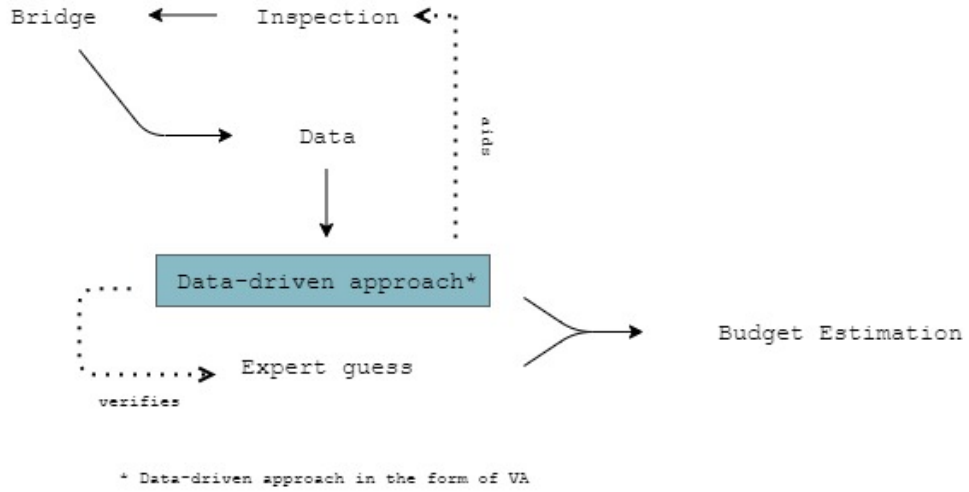


Figure 1.3: New method of estimating budgets

from the nulmeting (initial inspection) and it is based on reference projects from previous structures that are similar as well as the general knowledge of the expert. With the new method this human-based estimation is supported by using a data-driven approach which verifies the human guess by predicting a budget range. In this way the data-driven approach serves as a Virtual Assistant (VA) for the expert by verifying their guess.

Therefore the main objective of this research is:

The development of a model, in the form of a Virtual Assistant, that verifies the expert guess by predicting the budget range of maintenance.

1.6 RESEARCH QUESTIONS

From the main objective the main research question follows:

How can we improve the objectivity of a preliminary budget-estimate, with regards to the maintenance of civil engineering structures?

A series of sub-questions are formulated to direct the study and to provide an answer to the main question:

1. What kind of data-driven approach should be applied?
2. How is the data-driven model trained and what is the output?
3. How does this research compare to the current approach of estimating?
4. What are the main elements of an estimation model for maintenance budgets?

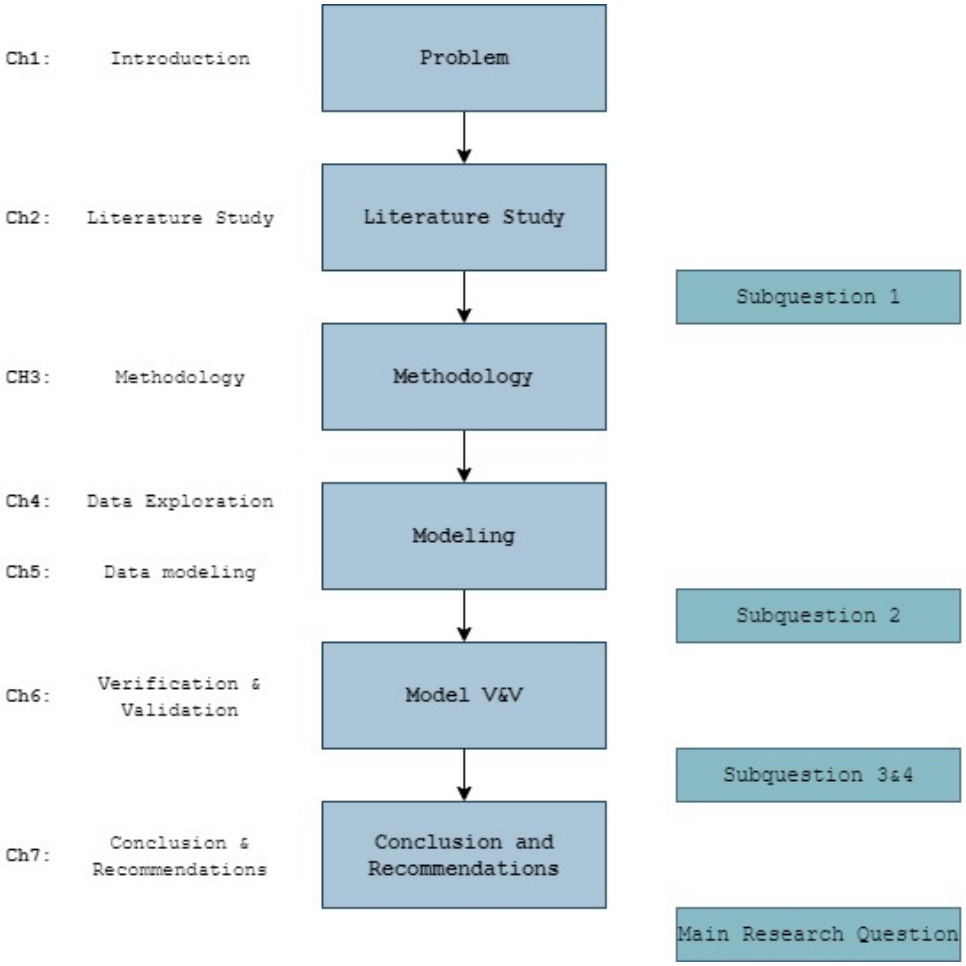


Figure 1.4: Structure thesis

1.7 THESIS STRUCTURE

This thesis introduces the problem to be solved in Chapter 1 and the methodology to do so in Chapter 3. The methodology is derived form a literature study which is conducted in Chapter 2 to find the best fit model for the solution. The development of the model is documented in Chapters 4 & 5 and it is evaluated in Chapter 6. Finally the thesis is concluded in Chapter 7. The overall thesis structure is depicted in Fig. 1.4.

2 LITERATURE STUDY & MODEL THEORY

This chapter introduces budget estimation for civil structures using a range of different method and in special the focus of this thesis, which is data-driven approaches. From these approaches the most fitting one for the thesis is chosen. Chapter 2.1 and 2.2 show the literature study and chapter 2.3 and 2.4 elaborate on the model theory used for this thesis. Therefore this chapter also answers the first sub-question: *What kind of data-driven approach should be applied?*

2.1 BUDGET ESTIMATION IN THE CIVIL ENGINEERING DOMAIN

Budget estimation is the most important preliminary process in any construction project since it ensures the successful completion of a construction project (Elfaki et al. [2014]). At the moment the budget estimation is a knowledge-intensive engineering task, relying heavily on the expertise of the cost estimating professional (Staub-French et al. [2003]). However, the last few decades with the introduction of digitization and use of computers we are able to perform the same numerical and symbolic manipulations a person can, but faster and more reliable (Hopgood [2012]). Raftery [1987] categorized the budget estimation into three generations: 1) budget estimation based on unit price (developed in the 1950's), 2) budget estimation based on statistical methods (developed in the 1970's) and 3) budget estimation based on intelligent methods (developed in the 1980's). Therefore in today's generation the focus should be budget estimation using intelligent systems.

Numerous studies have been reported in literature concerning the maintenance costs and maintenance budgeting (Srivastava et al. [2020]). Several studies have been done on using predictive methods for maintenance work but there is little information to be found on using intelligent systems for maintenance budgeting. The following studies give insight in budget estimation in the civil engineering field in current practice.

Evdorides et al. [2002] propose a framework for the programming of the maintenance of roads and bridges. The output of this framework is a set of maintenance projects where the total cost for the roads are specified. Using this framework a more objective way of defining which maintenance tasks need to performed can be carried out, which leads for prevention of unnecessary costs. However, this is an analytical framework for the prevention of extra costs. This means it is not focused on estimating maintenance budgets but is more focused on what works need to be carried out to have an efficient programming of infrastructure. The framework is generic and there is no proposed method on which algorithm to use to achieve the objective of preventing cost overruns.

Scarf [2007] also proposes a framework for maintenance management. Again it is very generic and it does not specify how it can be used for budgeting.

Wang et al. [2008] propose a more fitting framework. They propose a model which predicts the amount of restorations cost to be made based on previous data and a k-nearest neighbor approach. They made use of historical data to make a prediction. However the estimation is regarding restoration budgets and not maintenance budgets, which means that it has a deterministic character. Maintenance budgets are less predictable and a model exactly like this does not fit.

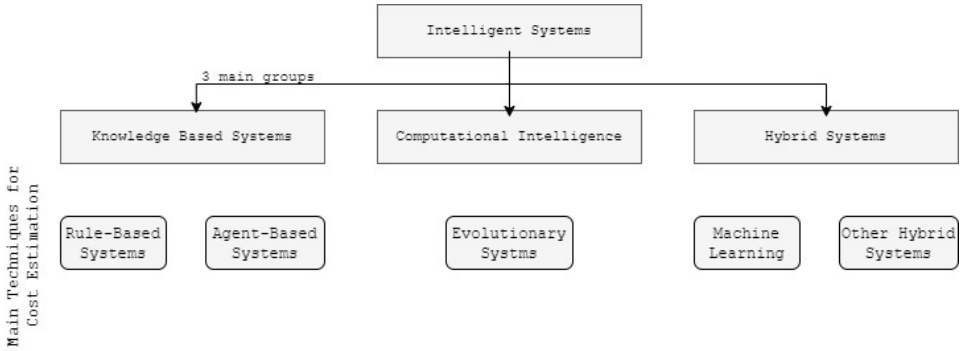


Figure 2.1: Main techniques for cost estimation sorted by Intelligent System group it falls under.

Although current literature on maintenance budget estimation in the civil field does propose the use of intelligent systems to make an estimation, it does not specify on which technique is the best. Current methodologies for budget estimation include regression analysis, artificial neural networks, fuzzy logic, and case-based reasoning (Kim et al. [2012]). The following sections give a background on these different intelligent systems in order to select the technique to be used in this thesis.

2.2 INTELLIGENT SYSTEMS & BUDGET ESTIMATION

Intelligent systems can be roughly divided into 3 main groups:

1. **Knowledge-Based Systems (KBS):** Where conventional programming intertwines domain knowledge with the software controlling the application of that knowledge, the knowledge-based systems separates them. The two explicitly separated system are the knowledge-base and the inference module (Hopgood [2012]).

2. **Computational Intelligence (CI):** Unlike the KBS, here the knowledge is not explicitly stated but it is represented by numerical values. As the system improves its accuracy, these values might be subject to change (Hopgood [2012]).

3. **Hybrid Systems (HS):** In many cases KBS and CI could work complementary to each other and thus be used together in a hybrid system (Hopgood [2012]).

These intelligent systems are also used in budget estimation. For the budget estimation of construction projects Elfaki et al. [2014] reviewed and analyzed proposals regarding budget estimation techniques for construction projects in a 10-year long survey. They found five main sorts of intelligent systems (see Fig. 2.1): Machine Learning [HS], Rule-Based Systems [KBS], Evolutionary Systems [CI], Agent-Based Systems [KBS] and Other Hybrid Systems [HS] (Elfaki et al. [2014]).

The following explains these systems:

Machine Learning

Machine Learning (ML) is when pre-solved data and the resulting output are fed to the computer. These two are used to create a program, which does the job of traditional programming (Sullivan [2017]). Machine learning can be seen as a hybrid system.

Rule-Based Systems

A rule-based system is a KBS where the knowledge base is represented in the form of a set, or sets, of rules. In order for the system to work, it also needs to have access to facts, unconditional statements which are assumed to be correct (Hopgood [2012]).

Evolutionary Systems

When it is hard to formulate a problem statement, evolutionary systems may come in handy. Miettinen et al. [1999] explain that at a high level of abstraction it is compared to the evolutionary process, where the more fit the individuals, the more influence there is in the future makeup of the population through the concept survival of the fittest. The most important components in evolutionary systems are: the population of the individuals, the notion of fitness, the bias, and the notion of inheritance (Miettinen et al. [1999]). Evolutionary systems fall under the CI group of intelligent systems.

Agent-Based Systems

Agent-Based Systems act analogous to human societies and organizations. The systems contain agents, which are intelligent computerized assistants, that are capable of achieving a goal in a way that is autonomous, cooperative and collaborative (Sugumaran [1998]). Agent-Based Systems can be seen as KBS.

The techniques described by Elfaki et al. [2014] all suffice for budget estimation. In order to understand which type of intelligent system to use for predicting whether the maintenance budget is priced right it is important to have an understanding of the different types of approaches to maintenance management and understand the techniques used there. Susto et al. [2014] explain three different types of approaches to maintenance management:

1. **Run-to-failure maintenance (R2F):** repair actions happen after the defect is detected.
2. **Preventive Maintenance (PvM):** maintenance is scheduled and carried out periodically with the aim of anticipating the process failures.
3. **Predictive Maintenance (PdM):** by continuous monitoring of the process health, maintenance is performed only when needed. PdM also uses prediction tools to assess when the future maintenance should be performed.

In the case of this thesis, we are dealing with Predictive Maintenance (PdM) since the condition of the bridge is continuously monitored through periodical inspection and maintenance is performed when needed. PdM-related solutions based on ML techniques seem to be among the most popular techniques (Susto et al. [2014]). Therefore this thesis is focusing on the use of ML as intelligent system. In theory any of these types of intelligent systems can be used for predictive budget estimation but given that ML is the most popular technique the focus is machine learning.

2.3 MACHINE LEARNING AS PREDICTIVE MODELING METHOD

Machine learning has three main different types of learning styles (Sullivan [2017]):

1. **Supervised Learning:** the input data and output data are known. Through training of the data a predictive model is built. This training process is repeated until it achieves the desired level of accuracy.

2. **Unsupervised Learning:** the output data is not known so a model is constructed by estimating the number of structures present in the input data in order to arrive at general rules. With unsupervised learning we therefore do not have a level of accuracy. Examples of unsupervised learning problems are e.g.dimension reduction, clustering and association rule learning.

3. **Semi-supervised Learning:** the input data includes a mixture of labeled as well as unlabeled data. The model needs to organize the data and besides that also make predictions.

The goal of this thesis is predictive modeling and for this reason supervised or semi-supervised learning method should be used. Since this thesis only contains labeled data, the use of a supervised learning suffices.

Predictive modeling is the art of building models that make prediction based on patterns found in historical data (Kelleher et al. [2015]). Overall, there are many different types of predictive algorithms for machine learning. The main groups are (Kelleher et al. [2015]):

- **Information based learning:** through the use of data, information is extracted and concepts such as most information gain and least information loss are most important.

- **Similarity based learning:** by looking what have worked well in the past, new predictions are made.

- **Probability based learning:** fundamentals of probability theory and Bayes' theorem are used e.g.calculating probabilities based on relative frequencies and conditional probabilities.

- **Error based learning:** a search for a set of parameters that minimizes the total error of the prediction is performed.

Currently, the technique that is used for making the prediction by the estimator is a similarity based prediction. By exploring various reference projects that are similar to the new case, the budgets for maintenance of new structures are predicted. The goal of this thesis is to develop a VA that verifies the prediction by the estimator. Therefore another type of learning method is used to eliminate any bias that can appear from using the same approach. This is information-based learning. The reason for this is that we have the following data to our disposal: the characteristics of the bridge, condition/state of the bridge, historical cost data, historical budget data. Due to the complexity of the relations between these data sets, an information based learning approach is most appropriate.

2.4 MACHINE LEARNING: CHOSEN TECHNIQUE FOR THIS RESEARCH

From the above it is apparent that this research deals with: a supervised predictive machine learning approach that is information based. For this kind of method a wide range of techniques have been developed (Kotsiantis et al. [2006]). The main techniques, as defined by Rokach and Maimon [2008] are shown in figure 2.2. In this research the chosen technique is Decision Tree Classification (DTC) because of 1) its applicability to this research in particular and 2) its ease of use and finally 3) because of the gap in academic literature.

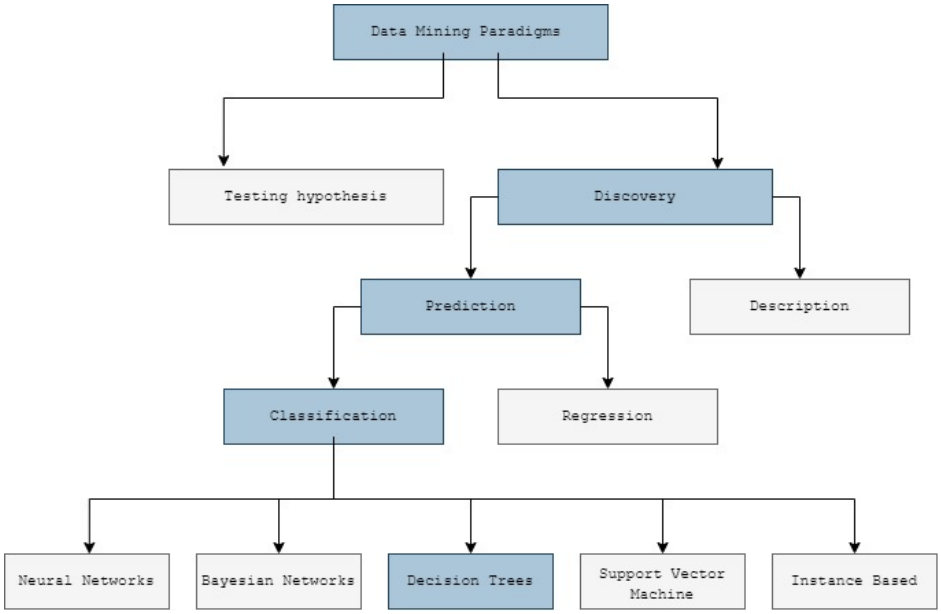


Figure 2.2: Taxonomy of data mining Methods as defined by Rokach and Maimon [2008]. The blue shows the chosen technique for this research.

2.4.1 Decision Tree Classification: Applicability

DTC is applicable because the standardization aspect within this research makes it easy to classify information.

The case for this research is to improve the objectivity of budget estimates, through machine learning using an information based approach that only takes into account objective data. The data that is dealt with is to a great extent standardized, making it easy to classify. The basis for the data are inspection reports, based on the Dutch Normative NEN2767, and cost data of different bridges in the Netherlands, provided by BAM. Documented data used to support bridge management vary from agency to agency (Sanford et al. [1999]) and this case is no different. The only thing that ties these data-sets together are the standardized methods and if a model is made on the basis of this standardization it will give an objective way to compare the different bridges.

DTC is applicable because the algorithm filters out all unnecessary information, making it easy for the user to deal with the complexity of the information.

There lays a complexity in understanding the factors affecting the price, especially the price of maintenance budgets. It is difficult to understand what the exact factors are that make up a budget estimation and the relations between these different factors. Therefore it is necessary to include as much input as possible and let the algorithm filter out the unnecessary information. This research is based the premise of using objective data and therefore minimize the assumption factor or human heuristic. Defining the factors that influence the price by ourselves defeats this purpose of minimizing bias. With DTC the model algorithm filters out the most important influence factors on the price.

2.4.2 Decision Tree Classification: Ease of use

DTC provides ease of use compared to the other methods, mainly because:

1. it is simple to understand, interpret and visualize, especially the smaller-sized trees (Tan et al. [2006]).
2. little effort is required for data preparation as DTC is a non-parametric approach for building classification models, in other words, it does not require any prior assumptions regrading the probability distribution satisfied by class and other attributes (Tan et al. [2006]).
3. can handle both numerical and categorical data as DTC can perform both classification, regression and multi-output tasks (Géron [2017]).
4. non linear parameters don't effect its performance since DTC works, unlike linear regression models. When there is a high non-linearity as well as a complex relationship between the independent dependent variables, a tree model will serve better than a regular method (Sullivan [2017]).

2.5 SUB-QUESTION 1

This chapter answered the first question:

What kind of data-driven approach should be applied?

The last few decades with the introduction of digitization and use of computers we are able to perform the same numerical and symbolic manipulations a person can, but faster and more reliable. Therefore there is a potential in using intelligent systems to at least verify the human estimators guess by computerized methods. There are several intelligent system models that can be used for this, but given the context of maintenance budgets, machine learning is the most popular technique that is currently used. Machine learning has different learning methods and since this thesis is focused on predictive modelling, the learning method used is supervised learning. There are several predictive machine learning techniques but for this research the chosen technique is Decision Trees because of 1) its applicability to this research in particular (see Ch. 2.4.1), 2) its ease of use (see Ch. 2.4.2).

3 METHODOLOGY

The research methodology is outlined in this chapter. First the DTC model algorithm is explained and hereafter the general methodology for the thesis is derived. Finally the workflow for answering the guiding sub-questions is discussed.

3.1 DECISION TREE CLASSIFICATION

Decision trees are used in data mining and in operations research. Although their form looks the same, there is a fundamental difference. In data mining a decision tree is a predictive model and in operations research it refers to a hierarchical model of decisions and their consequences in order to help decision making and strategy planning (Rokach and Maimon [2008]). The focus of this research is a decision tree as predictive model.

In general terms DTC uses a tree like structure where each internal node denotes a test on an attribute, each branch represents an outcome of the test and each leave/terminal node holds a class label (See Fig. 3.1). Decision trees find and identify the most significant variable as well as its value (Sullivan [2017]).The question arises how the splits are made, and the answer lays in the type of decision tree algorithm used.

There are several algorithms for DTC, of which the most popular ones are ID3, CART and C4.5 (Singh and Gupta [2014]). The difference between the algorithms are shown in Table 3.2. In order to select which algorithm it is important to note the characteristics of this project (see Table 3.1).

Input Data	Output Data	Missing Values	Outliers
Categorical/numerical	Categorical	Yes	Yes

Table 3.1: Characteristics research data

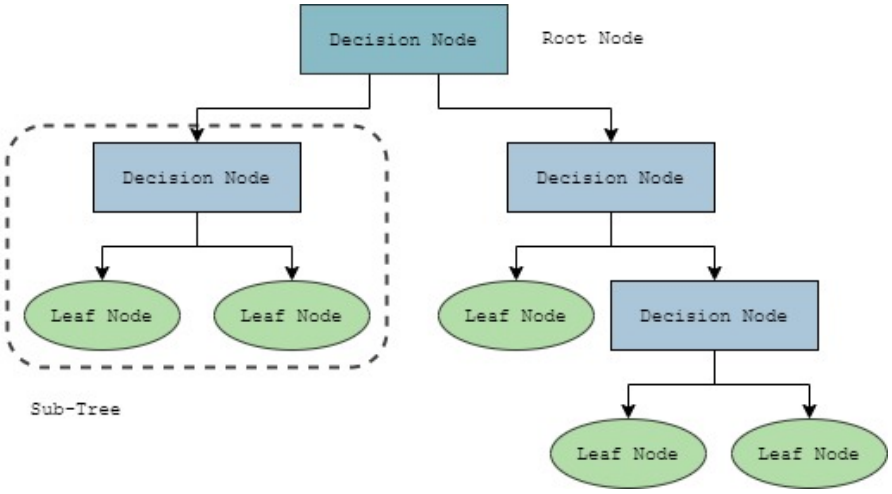


Figure 3.1: Decision Tree Structure

Characteristic(→) / Algorithm(↓)	Splitting Criteria	Attribute type	Missing values	Pruning Strategy	Outlier Detection
ID3	Information Gain	Handles only Categorical value	Do not handle missing values.	No pruning is done	Susceptible on outliers
CART	Towing Criteria	Handles both Categorical and Numeric value	Handle missing values.	Cost-Complexity pruning is used	Can handle Outliers
C4.5	Gain Ratio	Handles both Categorical and Numeric value	Handle missing values.	Error Based pruning is used	Susceptible on outliers

Table 3.2: Basic characteristic of decision tree algorithms Singh and Gupta [2014]

From the above it is apparent that there is no possibility for this thesis to use the ID3 algorithm since ID3 is not familiar with numerical values. In this thesis the input data mostly consists of numerical values as there is a mention of size and dimension, condition scores and finance. Therefore using ID3 is out of the question. Furthermore C4.5. is also not a fitting option in this case because of the difference in documentation, we are dealing with missing values. Finally, the most fitting algorithm to use for this thesis is the CART algorithm. CART stands for Classification and Regression Trees and it was developed by Breiman et al.in 1984. The key idea of CART is recursive partitioning. This means that the process begins by taking in to account all the data and all possible variables for growing a tree. From here it will select what the best split is considering the target attribute. The tree repeats this process until it cannot find another split (Boonamnuay et al. [2018]).

3.2 GENERAL METHODOLOGY

The general methodology for this thesis is the methodology of Géron [2017] for the development of the VA model using a machine learning approach (see Fig. 3.2). As mentioned in Chapter 2 machine learning uses pre-solved input and output as a basis for future predictions. This means that a VA model using a machine learning approach heavily relies on the data and the way it is trained by feeding the algorithm different types of dataset. Hereafter the solution is evaluated by defining the accuracy of the model. For machine learning the accuracy needs to be between 85-95 %. If that is the case then the model work and it is ready for launch. It does not reach this accuracy, the errors need to be analyzed and another iteration is done. This goes on until the required accuracy is achieved.

3.3 WORKFLOW

In order to answer the main research question, four guiding sub-questions are defined. The workflow for answering these questions is as follows:

1. **What kind of data-driven approach should be applied?**

   The first question explores the different types of data-driven approaches, applicable to estimation, through a literature study. This literature study leads to several options for data-driven approaches. According to the characteristics of the data input & output a fitting model is chosen. This sub-question is answered in Chapter 2.

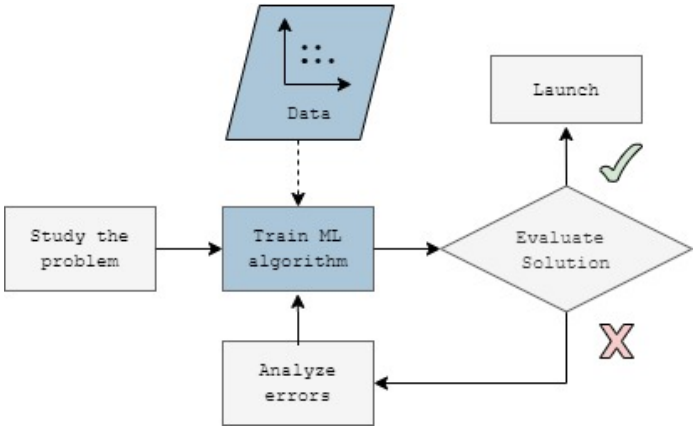


Figure 3.2: Machine learning modeling approach as defined by Géron [2017]

2. **How is the data-driven model trained and what is the output?**

   This chapter explores the way the model is trained and its output. This is done by firstly creating an artificial database and secondly training the data using the DTC algorithm. An artificial database is created because the real dataset, given by construction contractor BAM, does not contain enough data to train the model. The sub-question is answered in Chapter 5.

3. **How does this research compare to the current approach of estimating?**

   The added value of the new approach to estimating is explored through a comparison between the human way and the ML way of estimating. This is done through creating some fictional cases and comparing both methods to understand the differences. The answer to this sub-question is given in Chapter 6.

4. **What are the main elements of an estimation model for maintenance budgets?**

   The last sub-question explores what is generally needed for building a model for maintenance budgets using the DTC algorithm. This sub-question can be answered only after the model is built, verified and validated. This way it is ensured that all relevant maintenance budgets elements are present. The verification is done on the basis of an accuracy test and a sensitivity analysis and the validation is done on the basis of an expert review. The answer to this question is given in Chapter 6.

4 DATA EXPLORATION

From the previous chapters it is apparent that there is a need to develop a machine learning model using decision tree methodology. This chapter explores how to develop this model by establishing an understanding of the data through exploration.

4.1 DATA SOURCES

This thesis focuses on developing a model using a data-driven approach, therefore it relies heavily on the use of different data sources. The data sets are made available by the construction contractor BAM. The data sets include: the NEN2767 decomposition of bridges, the inspection reports of several bridges, the financial data of the bridges. However upon exploring these data-sets it was apparent that the provided data is not enough to build a working ML model using DTC methodology. Therefore an artificial database is made (see Ch. 5) to mimic how the model would work if the data-set was complete for modeling. In order to create this artificial database the input and output needs to be known. This chapter explores the data set as provided by BAM (see Fig. 4.3) and creates an understanding of what input and output to model.

4.2 DATA DOCUMENTATION: ABT TABLE

Machine learning works through documenting historical data in a systemic manner so that the machine learning algorithm can predict a new case, based on this historical data. Therefore there is a need for a database. The way this database is structured for predictive data analytic models is in the form of a Analytics Base Table (ABT) (See Fig. 4.1). The columns under the descriptive features describe the input (x) and the column under the target features describes the output (y). The whole classification model is dependent on these features (see Fig. 4.2). The following sections explain how this ABT table, so the database, is filled in for the case of this thesis.

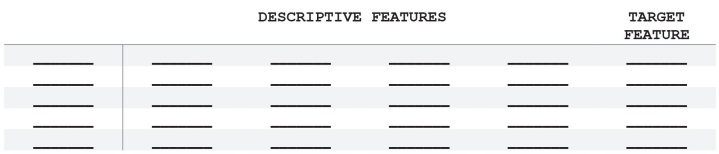


Figure 4.1: ABT table example

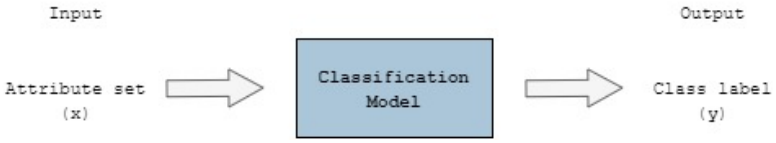


Figure 4.2: Classification as the task of mapping an input attribute set x into its class label y Tan et al. [2006]

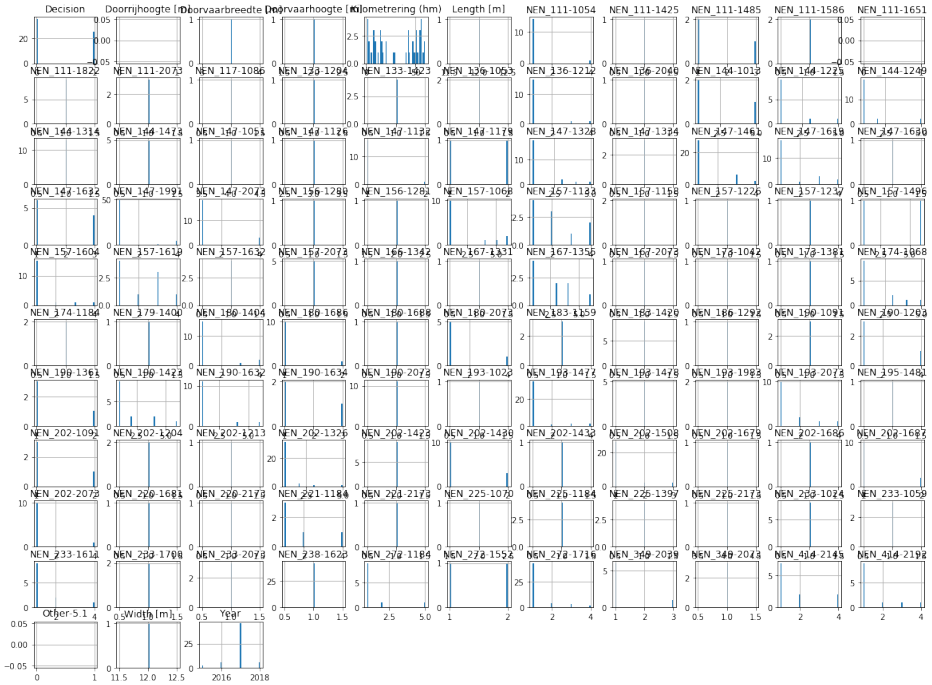


Figure 4.3: Dataset summary in histogram

4.3 DATA INPUT & OUTPUT

From the BAM data there is a total of 60 bridges and with that 60 accompanying documents that are used as a reference for the data input & output. The documents are inspection reports and some cases also include budgeting reports. The inspection reports are thoroughly read and the recurring elements of these reports are documented. These recurring elements are the basis of input (x) of the database. The budget reports are the basis of the output (y). Figure 4.3 depicts the summary of the recurring features in histograms, with the feature on the x-axis and the frequency of occurrence on the y-axis. It is apparent from figure 4.3 that the recurring features are mostly regarding the placing of the bridge, the size of the bridge, the building elements of the bridge and the condition of these building elements. Furthermore we can conclude that the data is non-parametric, some features are more often present than others and the data does not seem to behave following a distribution. Another aspect that is apparent is that there are many input features (112 features).

For this case and the amount of data as given by the contractor there is no possibility to create a working model. However, there is a possibility to create artificial data in order to prove the feasibility of the model (See Chapter 5). Beforehand it is important to specify the data needed for this model (input) and the maintenance budget (output).

4.4 INPUT

Based on the inspection reports provided by BAM the following main elements are at the basis of this model: the characteristics of the bridge, the NEN2767 condition data and the total duration of the maintenance.

4.4.1 Characteristics of the bridge & duration maintenance

The characteristics of the bridge that are relevant are mostly regarding the *dimensions* of the bridge and the *location* of the bridge. The dimensions can have an effect on the budget however the location does not amount to any impact on the budget so it can be disregarded.

The duration of the maintenance has a large effect on the final budget so this needs to be included. In the BAM dataset the duration is always the same which leads to this aspect being disregarded by the DTC algorithm. This is because the case duration has just one input (in this case 25 years) so it would be disregarded all in all. For this reason it is needed to note this as an critical factor for the development of the final model for the VA.

4.4.2 NEN2767: Decomposition & condition

The most important data set is the NEN2767, which provides 1) a decomposition of the building elements and parts of the structure and 2) the condition of these elements and parts. The Dutch Normative NEN2767 is developed with the goal of solving the problem of variation in methods of inspection. It makes it possible to measure the state of the structure and record defects in a unambiguous way. The initial purpose for the development of this inspection methodology was the prevention of subjectivity in the distribution of funds for urban development. This thesis case has a similar reasoning, the prevention of accounting too much money for maintenance.

Decomposition

The NEN decomposition needs to be included since it forms the key element to compare the different bridges to each other. The NEN2767 defines this decomposition by listing all the elements and building parts that a standard bridge should have (see Fig. 4.4).

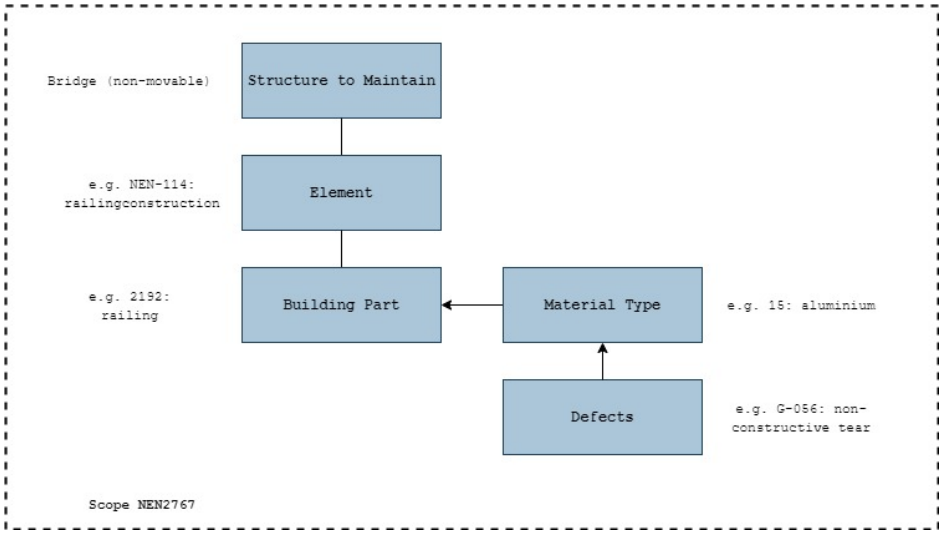


Figure 4.4: Scope NEN2767

Condition Score	Description	Explanation
1	Excellent	Occasionally minor defects
2	Good	Incipient aging
3	Reasonable	Locally visible aging, functional performance not at risk
4	Moderate	Functional performance at risk
5	Bad	Aging is irreversible
6	Very Bad	Technically ready for demolition

Figure 4.5: Condition Score

Condition

From the decomposition a standard condition score can be derived that corresponds to the amount and intensity of the defects. The defects are all standardized and begin with a G-code. An example of a way this coding system works is:

NEN-114-2192 with defect 15-G056 means *from the railing construction, the aluminum building part railing has the defect: non-constructive tear.*

The main principle is the more defects, the higher the condition score, the higher the maintenance budget should be.

Through the decomposition a condition score can be given to each of the elements. There are 6 condition scores (see Fig. 4.5). Appendix A explains these condition scores in more detail. For budgeting however we only use condition 1-5 because 6 refers to the demolition state and that is not included in a maintenance budget.

4.5 OUTPUT: BUDGET ESTIMATION

The output in this case is the historical data of the maintenance budgets for the 60 available bridges. An important implication in this case is that for only 6/60 bridges these reports are available. Therefore for the output artificial data is made based on the available budget reports.

5 DATA MODELING

This chapter explains the modeling process of this thesis. From Chapter 4 it is apparent that there is a need for an artificial database which will be elaborated in this chapter. Furthermore the final result of the model is compared to another algorithm. Finally an answer to the second sub-question is given in the conclusion: *How is the data-driven model trained and what is the output?*

5.1 DATA TYPES

From the previous chapter the following input and output data together with their data types can be distinguished: bridge size, total duration of the maintenance of the bridge, the NEN-norm which captures the decomposition and state of the bridge & the budget. The data with the accompanying data types, which show the way the data is processed in the ABT table, can be seen in Fig. 5.1.

Data	Bridge Size	Duration Maintenance	NEN-norm	Budget
Datatype	Categorical	Numerical	Categorical	Categorical

Table 5.1: Data types

5.2 SAMPLE SIZE

In order to make the synthetic data there needs to be an understanding of what sample size is needed to reach the adequate performance target. For DTC an adequate performance target is 85-95 %. This is important to define, not only to get an accurate estimate but also because in the real life application the gathering data can be difficult to obtain (Figueroa et al. [2012]).

Sug [2009] suggest that more data does note equate a better decision tree. Therefore a repeated sampling method is proposed using different sample sizes to decide the best sample size for the given problem. For this thesis the repeated sampling is done with three sample sizes (See Table 5.2 for the results). From this it can be derived that having a sample size of 300 (with overlap in data) already suffices for constructing a working model. To explain that this sample size already suffices, even though the tree can be more accurate, there are two reasons:

Reason 1: Performance target is reached with a sample size of 300

For decision tree classification the performance target is a model that has an accuracy of 85-95%, and the results of the synthetic data with a sample size of 300 is already in the range of the performance target.

Sample Size	300	600	1200
Accuracy	0.8889	0.9000	0.9144

Table 5.2: Sample size and corresponding accuracy

Reason 2: The most realistic sample size for this model is a size of 300

In The Netherlands there are around 3700 bridges and viaducts (Octrooicentrum Nederland [2008]). Of these structures, BAM maintains and operates 60 bridges. The largest player in The Netherlands is Rijkswaterstaat and they operate 885 bridges (Rijkswaterstaat [2020]). This means that even if BAM expands their inventory of bridges it will most likely not exceed 100 and in the near future it will stay around 60. This means that many of the bridges in our database are going to be of the same type, making it easy to classify and have overlap in the database. This also means that a sample size of 300 is a realistic target for this model to work.

5.3 MODEL INPUT

In order to create the model there is a need for artificial data, since the data at hand is not complete. For classification to work, we need to create overlap. A way to do this is by clustering the data that is similar at first and therefore using classes instead of numerical values. The following explains how every feature is recorded in the database.

The results of this artificial data is shown in Appendix B and an overview of the artificial data is given in a pairplot in Appendix C.

Bridge size

For the artificial data there are 3 size classes of bridges defined. A way to classify, is by looking at the amount of building elements a bridge has. The more building elements, the larger the bridge, the higher the size class.

Duration maintenance

In the BAM case the duration of the maintenance is 25 years for each case. Therefore this is not going to be classified in the decision tree and it is disregarded. However if the database is supplemented with more data, then the decision tree will take into account the different duration's, provided that: the parameter for the decision tree *min sample split* $<$ the amount of newly added data.

NEN-norm

The NEN-norm consists of 3 system levels: element - building part - material type (see Fig. 4.4). To create overlap in the model and thus increase the similarities in the different cases, the NEN-norm is defined on the element level. This will not lead to problems since the element level is a summary of the building parts and it will always take on the worst condition of the building part as the condition score. Figure 5.1 shows this, as *Leuning algemeen* (Railing general) with condition score 1 and *Beschermlaag* (Protection layer) with condition score 2 are the building parts and *Leuning* (Railing) is the element level which has condition score 2.

Size class NEN-norm

The condition relates to the defect of every element but it does not say anything about the extent of the defect. Therefore it is necessary to introduce a measure to

Figure 5.1: Example of condition element as defined in an inspection report

	Licht (L)	Matig (M)	Ernstig (E)
Geringe omvang (1)	L1	M1	E1
Enige omvang (2)	L2	M2	E2
Grote omvang (3)	L3	M3	E3

Figure 5.2: Reference: CROW's way of defining size classes

the defect. In many cases this measure is *defect per m2*. However this can lead to a variation in m2 of the defect. Again, to create overlap in the model there are 3 Size Classes for the defect introduced. To prove the applicability of classifying defect in size classes: the same methodology is used by CROW which is a renowned Dutch knowledge platform that provides uniform tools for practical implementation of existing legislation and regulations. Fig. 5.2 shows the way CROW defines defects in size classes.

Budget

To mimic the non-parametric budget data from the BAM case there needs to be a variation in the way the budget is composed. The budget is created based on data found from the 6 cases that had budgeting information. In order to create variation in the budget a fixed price is derived from these 6 cases. This fixed price is taken for a standard budget that matches condition n. Hereafter this standard budget is taken as the mean for every project that matches condition n and the variation is created by using the normal distribution with: $\mu = standardbudget$ and $\sigma = 20\%$.

5.4 DTC ALGORITHM

The way the model predicts future budgets is depending on the Decision Tree algorithm. As the analysis chapter concluded this is the CART algorithm. A pseudo-code is shown in Algorithm 5.1.

The main steps in CART is building the tree, stopping the tree building process, pruning the tree and choosing the best tree.

Building the tree & stopping the tree

The way the tree is built depends on the chosen algorithm. The difference between the algorithms is the way they split the data. This research uses the CART algorithm

which uses the gini-impurity metric. Gini impurity looks at what the probability is that a datapoint is classified incorrectly, and chooses the smallest gini impurity index to split on (Lewis [2000]).

To decide when to stop a tree, the parameters of the tree can be tweaked, especially the parameter maximum depth of the tree. This shows how large the tree is in depth. In this case by trial and error the max depth is set to 11. Every maximum depth will give another accuracy. As long as the accuracy of the model is not under 85% the model is fit for use.

Pruning the tree & choosing the best tree

Once a tree is grown it can reach an accuracy of 100%. This means that the tree is overfitting the data. Therefore the result needs to be generalized again, which means that the tree needs to be pruned. The pruning is in this case not needed since pruning comes into play by little variation and a large dataset and this case is missing a large dataset. Furthermore since there will be no pruning there are also no trade-offs to be made as far as choosing the best tree.

Algorithm 5.1: CART pseudo-algorithm VA maintenance budgets using scikit-learn decision tree classifier

Input: X = The characteristics of the different bridges, the condition data and the intensity of defect

Output: Y = The decision whether the budget falls in class 1,2,3,4,5 or 6

```
Upload database
Fill the missing values with 0
Define output (Y); Define input (X)
Split data in train & test, with train = 0.7 & test = 0.3
Apply CART decision tree classification algorithm from scikit-learn
Test accuracy using confusion matrix
if Accuracy = 100% then
    prune decision tree
Visualize tree
Predict maintenance budget class for new case using predict function
```

5.5 MODEL OUTPUT

For this thesis an artificial case is made and a decision tree is grown. The sample size of this artificial case is 300 bridges. The database for this case is shown in Appendix B and the results of the fully grown tree are shown in Appendix C. The python code can be found in Appendix D. The tree contains a depth of 11 and has an overall accuracy of 88.9 %. This means it reached the performance target and it is reliable enough to start making predictions.

5.6 BUDGET CLASS PREDICTION VS BUDGET PREDICTION

A question that often is posed is whether the model can make a prediction of a budget instead of a budget class. With the decision tree methodology this is also possible but the decision tree classifier needs to be changed into a decision tree regressor. Reason for this being: given the data set S, then the observation $\{x_i, y_i\}$

contains the information related to process iteration n Susto et al. [2014]. In its mathematical form:

$$S = \{x_i, y_i\}_{i=1}^{n} \tag{5.1}$$

Where:

- S: dataset
- $\{x_i, y_i\}$: observation
- $x_i \in \mathbb{R}^{1*p}$: contains information
- y: output

IF y assumes continuous variables THEN a regression problem is obtained, whereas IF y assumes categorical variables THEN a classification problem is obtained (Susto et al. [2014]).

Even though in theory it is possible to make an estimate more defined, the practice shows a different result. To change this problem and make it fit for the decision tree regressor the only thing that needs to be changed in the data set is the output (y) which goes from budget class to budget (see Appendix E and Appendix F). From the input (x) the size class of the defect needs to be changed into a actual m2 of the defect. Besides this, decision tree regressor often requires more data, but for the sake of comparison between the classification and regression model, the data set is kept the same. Now the decision tree regression can be modeled of which the algorithm is given in 5.2 and the output is shown in Appendix E.

There are several methods to evaluate the model, of which some are the Mean Square Error (MSE), root MSE and R2 error. For this model, since again we are dealing with non-parametric and noisy data, the most fitting method is R2 error. MSE tends to overestimate the badness of the model. The R-Square Error associated with the model is: 0.6756. This is significantly lower than the classification accuracy.

All in all, the previous shows that the classification method gives a better result for this case of estimating maintenance budgets.

Algorithm 5.2: CART pseudo-algorithm VA maintenance budgets using scikit-learn decision tree regressor

Input: X = The characteristics of the different bridges, the condition data and the intensity of defect
Output: Y = The decision of what the budget is

1 Upload database
2 Fill the missing values with 0
3 Define output (Y); Define input (X)
4 Split data in train & test, with train = 0.7 & test = 0.3
5 Apply CART decision tree regressor algorithm from scikit-learn
6 Test accuracy using R square error

5.7 SUB-QUESTION 2

This chapter answered the second sub-question:

How is the data-driven model trained and what is the output?

This thesis uses a data-set from the construction contractor BAM in order to create a DTC model. From Chapter 4 it is apparent that the data-set consists of data regarding the bridge size, duration of maintenance and the NEN norm as input and the budget class as the output. The input data and output data is divided into classes. A random sampling is done to estimate the sample size which reaches the performance target, 85-95 % accuracy. This target is reached with a sample size of 300. After this sampling, the data is modeled using two different DTC algorithms. It is apparent that the CART classifier (see Algorithm 5.1) has the best performance for this case. The output of this model is a prediction of a budget class.

6 VERIFICATION & VALIDATION

In this chapter the model for the VA is verified by showing the accuracy test and a sensitivity analysis in the first part of the verification. This is followed by a second part of the verification which compares the current practice and the practice using the VA. Based on the previous the third sub-question is answered: *How does this research compare to the current approach of estimating?* Hereafter the model is validated by an expert review. The full expert review can be found in Appendix H. Finally the last sub-question can be answered: *What are the main elements of an estimation model for maintenance budgets?*

6.1 VERIFICATION: PART I

6.1.1 Accuracy of model: confusion matrix

In order to evaluate the decision tree classification technique an accuracy metric called the confusion matrix is used. The general idea is to count the total amount an instance of class A is classified as class B in your testing data set (Géron [2017]). The formula to compute this is the following (Boonamnuay et al. [2018]):

$$Accuracy = \frac{(TP + TN)}{(TP + FN + FP + TN)} \tag{6.1}$$

Where:

- TP = the number of data points from a positive class that are rightfully predicted as a positive class
- TN = the number of data points from negative class that are rightfully predicted as a negative class
- FP = the number of data points that are in reality from negative class but the model incorrectly predicts these as a positive class
- FN = the number of data points that are in reality from positive class but the model incorrectly predicts these as a negative class

The confusion matrix for this case is given in Fig. 6.1. In this case there is no binary classification but a classification with 5 labels. This means there is no TP or TN, but True Classes or False Classes: True/False Class 1, True/False Class 2, True/False Class 3, True/False Class 4, True/False Class 5. The numbers on the diagonal show the number of correctly predicted classes. The other numbers are the falsely predicted classes. Therefore the accuracy of the VA model = 80/90 = 0.8889

6.1.2 Sensitivity analysis model

Gaps in our knowledge in this case are bridged by assumptions regarding the data set, such as the assumptions of different classes as well as assumptions regarding

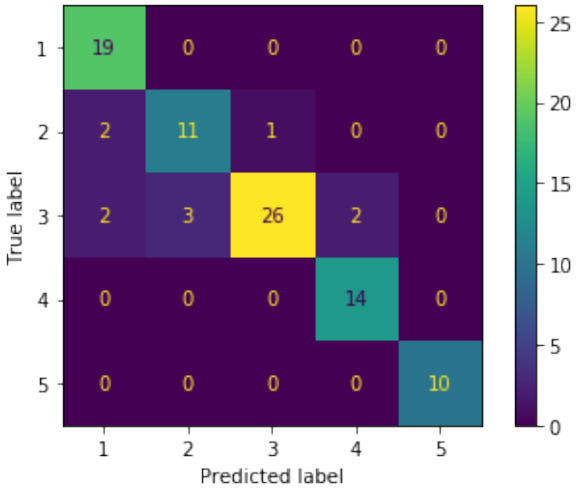


Figure 6.1: Confusion matrix

the decision tree model parameters. A sensitivity analysis can be used to systematically investigate the means by which assessors bridge these uncertainty gaps. It includes a what-if analysis for uncertain model parameters as well as the identification of the significant assumptions (Gorris and Yoe [2014]). The following identifies the significant assumptions and the shows sensitivity of the model parameters.

Significant Assumptions

The following assumptions are most significant and need to be provided for the VA model to work. Therefore these are not subject to change.

1. The model needs to have overlap in the data, regardless of the sample size. In this case this is done by e.g. introducing Size Classes to the NEN-elements.
2. In order for a new class to be included the number of times the class is present $>$ the minimum splitting criterion.
3. When a feature has only one class this feature is excluded in the classification, like in this case the feature 'Tot duration of maintenance.' These types of features need to be identified beforehand to be aware of this later on when the database is being supplemented with more data.

Model parameters

The VA model has several model parameters that are the basis of the model output. Using the scikit-learn module the following are the decision tree model parameters:

sklearn.tree.DecisionTreeClassifier(*, criterion = '', splitter = '', max_depth = '', min_samples_split = '', min_samples_leaf = '', min_weight_fraction_leaf = '', max_features = '', random_state = '', max_leaf_nodes = '', min_impurity_decrease = '', min_impurity_split = '', class_weight = '', presort = '', ccp_alpha = '')

For the VA model the parameters that are defined are shown below.

DecisionTreeClassifier(criterion = 'gini', max_depth = 11, min_samples_split = 2, random_state = 0)

A sensitivity analysis is performed on the model parameters: **max_depth** & **min_samples_split**. The parameter *criterion* does not have a sensitivity analysis because it is a fixed parameter. The whole CART algorithm depends on the splitting criterion being gini impurity. Furthermore random_state is disregarded for the same reason, it is a fixed parameter.

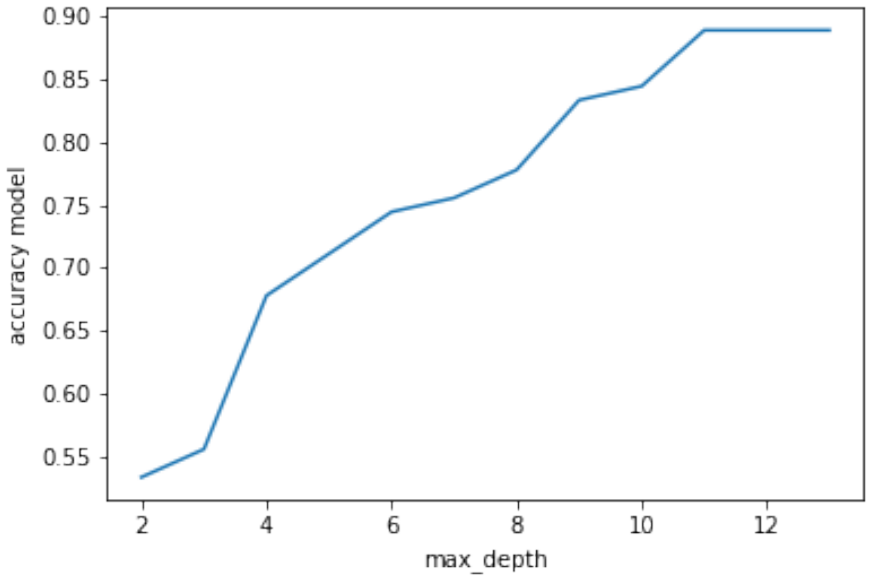


Figure 6.2: Sensitivity Analysis Graph max_depth & accuracy

max depth	accuracy model
2	0.5333
3	0.5556
4	0.6778
5	0.7111
6	0.7444
7	0.7555
8	0.7778
9	0.8333
10	0.8444
11	0.8889
12	0.8889
13	0.8889

Table 6.1: Sensitivity Analysis Table max_depth & accuracy

Max_depth

The max_depth is the depth of the tree, so how many sub-levels there are in a tree. Changing the depth of tree will change the accuracy of the model. The deeper the tree is allowed to grow, the more information it is allowed to capture. This means that there is a possibility for the tree to overfit, in other words, that the tree works perfect (= accuracy of 100%) on the training data but once new data is fed it is not going to work anymore. Therefore there needs to be a degree of generalization. This is where the max_depth is a useful parameter. It is important to also not overly generalize the tree, because that would cause underfitting which means a very low accuracy of the tree. For the VA model a sensitivity analysis is performed by filling in several criteria for the tree and see how it affects the accuracy (see Fig. 6.2). The graph shows the higher the max_depth, so the deeper the tree, the higher the accuracy of the model. It also shows that the model is not overfitting, since it does not achieve an accuracy of 100&. After a depth equal to 11 it will stay on an accuracy of 0.8889 percent. The VA model is trained on a depth equal to 11, so there will be no complication in the future.

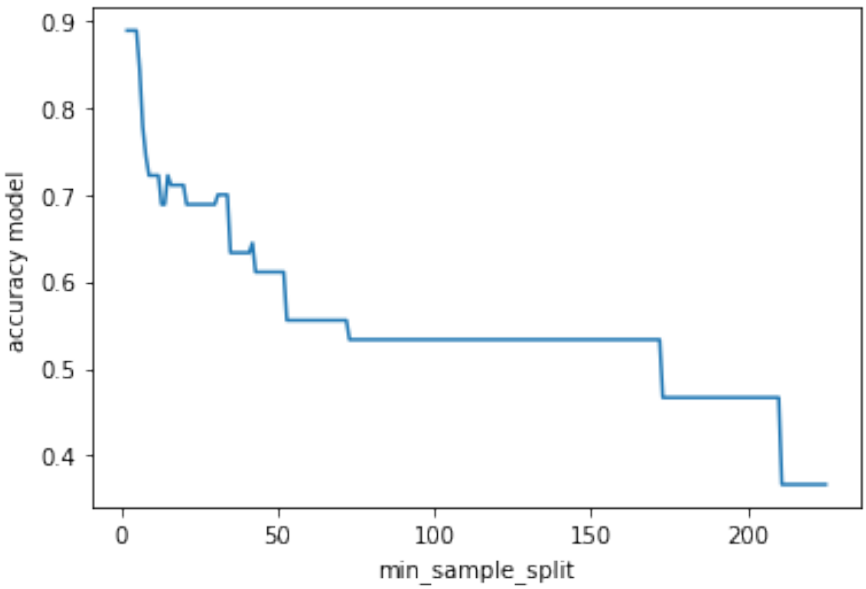


Figure 6.3: Sensitivity Analysis Graph min_sample_split & accuracy

min_sample_split	accuracy
2	0.8889
6	0.8444
10	0.7222
20	0.7111
30	0.6889
50	0.6111
100	0.6111
150	0.5333
200	0.4667
210	0.4667
211	0.3667
225	0.3667

Table 6.2: Sensitivity Analysis Table min_sample_split & accuracy

Min_sample_split

The min_sample_split parameter shows how many samples there should be in a node for it to split. The smaller the min_sample_split parameter, the deeper the tree can grow. From the previous (Fig. 6.2) it is apparent that a deeper tree has a higher accuracy. In the case of the VA the minimum split is set to 2, which allows the tree to grow. This is also the best way to start the tree. Only when the tree shows that it has an accuracy of 100% it is logical to start changing this parameter to a higher number. However, since that is not the case in the VA model (see Fig. 6.3), this parameter can stay the same and most likely will stay the same even if data is added.

6.2 VERIFICATION: PART II

6.2.1 Comparison to current way of estimating

The current way of estimating relies heavily on the expertise of a cost estimator. The cost estimator uses similar reference projects to determine the maintenance budget

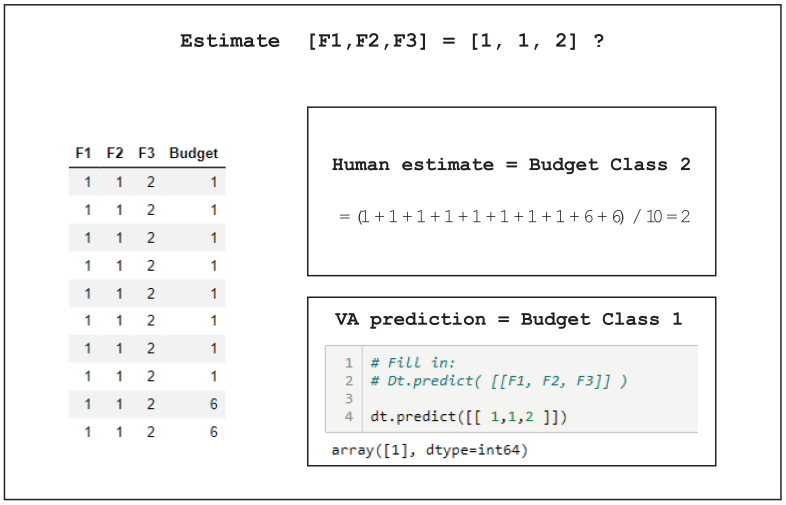


Figure 6.4: Proof 1

of a new case. Hereafter the average of the similar cases is taken to come to a budget. The human estimator uses a similarity-based approach and the VA uses information-based learning (see Chapter 2.3). The following few cases show the difference between the human estimate and the VA estimate and show how the VA provides 'better' estimates since it takes into account the whole database through the information-based learning. More examples can be found in Appendix I.

Case 1: "The VA does not take into account the average but looks at all the data and classifies where the prediction fits best"

Proof 1: Where the human estimate takes into account extreme values like in Fig. 6.4 and simply takes an average, the VA predicts a budget class that ends up in a leaf node by classifying it. In this example this leaf node corresponds to budget class 1 as this is where most of the cases are classified in. Therefore it filters out the extremes and takes information from the whole data set into account without estimating budgets that are unrealistic/false.

Case 2: "The VA considers, besides the same exact case, also similar cases to classify where the prediction fits best"

Proof 2:

In Fig. 6.5 the combination [1,1,2] has two different outcomes: budget class in 2 cases and budget class 4 in 2 cases. Because the VA takes into account the information of other cases as well, the model eventually classifies it as budget class 4. A human would take the average (budget class 3) and that is most definitely not right.

Case 3: "The VA can predict never seen before cases by classifying into already known tree"

Proof 3:

The last case shows that even if there is a totally new prediction to be made, the VA will classify it somewhere in a budget range. This is done on the basis of other

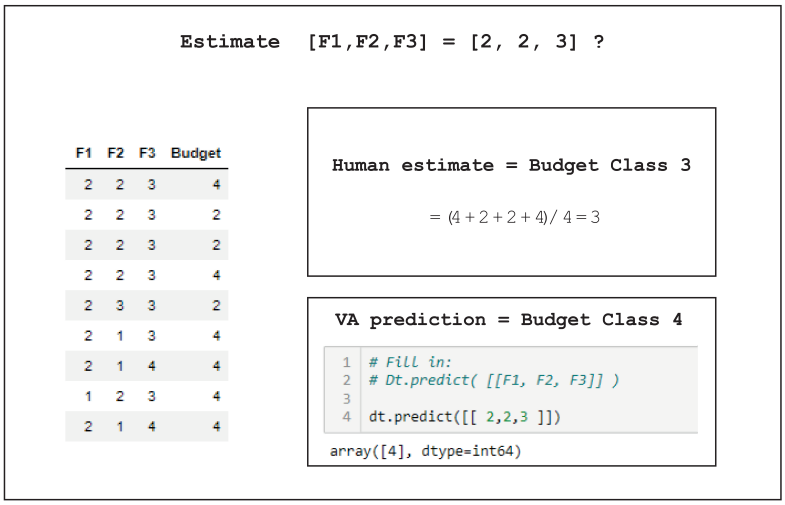


Figure 6.5: Proof 2

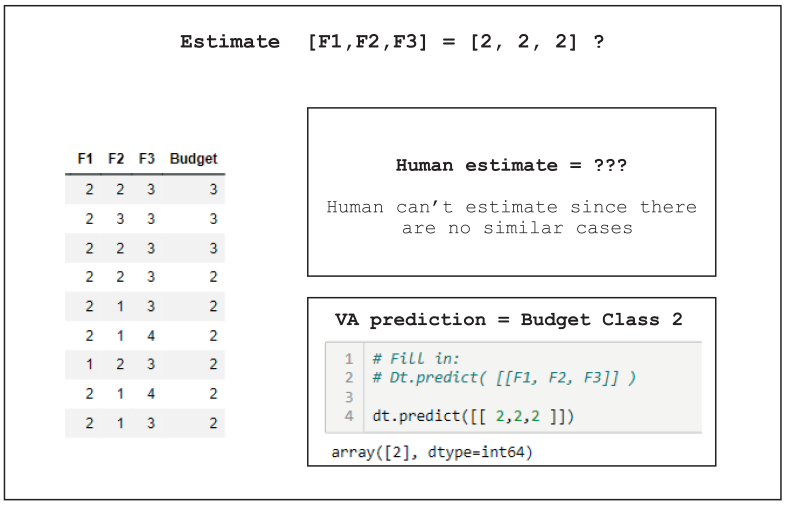


Figure 6.6: Proof 3

information from other projects that were found before. A human would not know where to start in this case (see Fig. 6.6.

6.3 SUB-QUESTION 3

The third sub-question can be answered now:

How does this research compare to the current approach of estimating?

This research compares to the current way of estimating by introducing a VA and creating three main differences by doing so:

1. The VA does not take into account the average but looks at all the data and classifies where the prediction fits best.
2. The VA considers, besides the same exact case, also similar cases to classify where the prediction fits best.
3. The VA can predict never seen before cases by classifying them into the already known tree.

Furthermore from current academic literature it is apparent that there does not exist a VA that uses DTC for prediction in the cases of maintenance budgets. This thesis proved that this is a fitting strategy, provided that the organization that uses the methodology aligns their data storage strategy with the VA model.

6.4 VALIDATION

The validation of the model is performed to evaluate whether it reached its development objectives and is therefore functional for industry use. This is done through an expert review. In Appendix H the full overview is given of the findings of the expert review at BAM infra. The expert review was conducted through demonstrating the VA model and asking for feedback. The team of experts at BAM consists of: an asset manager, a cost estimator and a project manager for inspections. All experts were asked for their feedback on the VA model by explaining their point of view on the VA's applicability to current practice and the feasibility of creating a VA. The question on applicability tests whether the VA model reached its desired development objectives according to the experts, which are also the users of the tool. The question on feasibility tests whether there are limitations that occur by using this technique. This is important to test because at the moment the model is based on artificial data so the expert perspective on the feasibility of a real model needs to be included.

6.4.1 Applicability of VA model to estimation

First the applicability of the VA model is tested through the review. The applicability means the extent to which the model reached its development objectives. These objectives can be derived from the development gap (see section 1.3) and are the following:

1. **Clear definition of project level:** The VA model has no ambiguity in the definition of the project level. It is clear how each project is defined.
2. **Independent prediction:** The VA model does not depend on the opinion of the cost estimator and can make a prediction independently.
3. **Clear definition of used tools and references:** The VA model clarifies what tools and references are used for the estimation.

Clear definition of project level

All of the experts agreed that the project level is defined clearly by using the VA. Each line in the database includes a description of each of the bridges by defining its size and decomposition. For every next bridge the same characteristics should be filled in to predict a new budget. Therefore the definition of project level is always ensured to be the same. Currently the accuracy of a budget estimate relies on the level to which the cost estimator defines a project. This approach is changed by using the VA because the extent to which a project needs to be defined is already preset.

Independent prediction:

When focusing on the dependence on the cost estimators opinion the general consensus was that the VA can predict independently. In theory anyone can use the tool to insert the data that is asked for and arrive at a prediction, without using the cost estimator. However it is to be noted that there was also an agreement that the VA prediction should be a support to the cost estimator and not a replacement, which is also the case here since it is a virtual *assistant*. The VA is used as a verification tool.

Clear definition of used tools and references

The objective of clear definition of used tools and references is also seen as achieved. The used tool for the VA model is the model itself and the used references are clear because there is a whole database which stores historical information about past bridges. Furthermore the way the way the VA shows a tree, containing the trade-offs that are made for arriving at a decision, make it very understandable. This helps the users to communicate better to outside parties what the most critical components are of the budget estimate.

6.4.2 Feasibility of VA model to be develop

Since the VA model used for the demo case relies on artificial data, the question regarding feasibility of development for a real case was posed. The model was presented at two levels of detail: VA that predicts a budget range, Level of Detail 1 (LOD1) and a VA that predicts the final budget, Level of Detail 2 (LOD2).

The general consensus was that the VA is feasible for industry use when developed at LOD1, predicting a budget class/range. This has less to do with the model and more to do with the way data is stored at the moment within the organization and the amount of data available at hand. Within the organization there needs to be more standardization and more standard procedures to generate objective data. When this is done the budget classes can be determined and a multidisciplinary team can look into whether to add more input features.

6.5 SUB-QUESTION 4

The previous chapters answered the fourth question:

What are the main elements of an estimation model for maintenance budgets?

The main elements of an estimation model for maintenance budget using the CART algorithm can be retrieved from an initial data exploration specific to the case at hand, verified and validated by expert review. From the case it is apparent

that the main elements are characteristics of the bridge, which refer mostly to the dimension of the bridge, the NEN2767, which capture the decomposition and condition of the bridge and to a lesser extent the duration of the maintenance. In this case the duration of the maintenance is always the same (25 years). Therefore this duration aspect is not included in the tree structure. Whenever new data is fed, this aspect will automatically be included if and only if the parameter for the decision tree min_sample_split is $<$ the amount of newly added data.

Furthermore other relevant data can also be included as long as they do not conflict with the significant assumptions, which are:

1. The model needs to have overlap in the data, regardless of the sample size. In this case this is done by e.g. introducing Size Classes to the nen-elements.
2. In order for a new class to be included the number of times the class is present $>$ the minimum splitting criterion.
3. When a feature has only one class, this feature is going to not be included in the classification, like in this case the feature 'Tot duration of maintenance.' These types of features need to be identified beforehand to be aware of this later on when the database is supplemented with more data.

Finally, in order to assure the model to function and make the input as complete as possible, there is a need for identifying the applicability of this model to the industry. From expert review it is apparent that a VA model is feasible to develop if data is documented in a standardized way using standard procedures set up by a multidisciplinary team of experts.

7 CONCLUSIONS & RECOMMENDATIONS

This chapter concludes the thesis by answering the main research question. Furthermore it proposes recommendation for future use. This is done in a process diagram for future steps to be implemented by a contractor if they choose to use this methodology for budget verification.

7.1 CONCLUSION

In today's practice the budget estimation of the maintenance of civil structures relies on the opinion of a cost estimator only. This results in estimations that are not accurate enough with a possibility of deviations from the estimated budget in the future which in turn will effect the economic performance of a contractor or construction company. However, with the increase of documentation of data there is a potential to make these estimates more objective. In order to do so, this thesis looked into developing a Virtual Assistant (VA) to verify the maintenance budget guess of the cost estimator using a data-driven approach. This VA fills the gap of: ambiguity in the definition of the project level, high dependence on expert-opinion and the ambiguity in the tools and references used by estimator. The main research question for this thesis follows:

How can we improve the objectivity of a preliminary budget-estimate, with regards to the maintenance of civil engineering structures?

The way the objectivity is improved in this thesis is through a data-driven approach. By conducting a literature study it is found that there are several intelligent systems that can be used for this. In current literature and given the context of maintenance budgets, machine learning is the most popular technique that is currently used. Machine learning is an approach where historical data is documented together with the results and on the basis of this new predictions are made. The machine will try to find patterns in the new data that correspond to what it already 'knows' in order to come to the best prediction. There are several predictive machine learning techniques but for this thesis the chosen technique is Decision Trees Classification.

In order to develop the VA model using Decision Tree Classification, data is retrieved from the construction contractor BAM. On the basis of this data an artificial (or mock) database, with a sample size of 300 bridges, is created. Only the objective elements are filtered out and fed to the model, these elements are: the bridge size, duration of maintenance and the NEN norm as input and the budget class as the output. The reason that the prediction of the VA is a budget class and not an actual budget is because of the accuracy of a budget class model (=0.85%) being higher than the accuracy of an actual budget model (=0.68%). This is proven by comparing both predictions and the accompanying algorithms. It is important to note that for the VA model to work some assumptions need to be included which are that: 1) there needs to be overlap in the data so that it allows the machine to find patterns and create rules and 2) the parameters of the model need to be re-checked when data is added.

Furthermore the VA can be used as verification for the estimators' guess since the approach of the VA is information-based and the approach of the human estimator

is similarity-based. This means that an estimator will use the cases only known from past experience whereas the VA uses a more holistic approach by taking into account all historic cases that ever existed and that are documented in its database. This fills the gap that currently exists in the way of estimating project budgets. By using a VA there is no need for a definition of the project level, all project levels together with their information are included in the definition of the database. Secondly there is less dependence on the expert-opinion alone since there is now a two-step verification, meaning computerized estimator and a human estimator. Thirdly the ambiguity by tools and references that are used by the estimators can also be handled since decision tree methodology actually forms a tree which shows all trade-offs made to arrive to a certain decision.

Finally from current academic literature it can be concluded that a similar VA, which uses DTC for prediction in the cases of maintenance budgets, does not exist yet. Nevertheless this thesis proved that this is a fitting strategy to implement for assisting the cost estimator, provided that the organization that uses the methodology aligns their data storage strategy with the VA model database.

7.2 RECOMMENDATIONS

From the expert review some remarks were made about the need for standardization within the organization to make this tool work in the future. Figure 7.1 shows a process diagram to use when trying to implement this VA methodology in an organization. This is based on the outcomes of the VA model made for this thesis and the expert review.

1. **Identify the domain of the maintenance budget, which structure do we have to maintain?**

   First it is necessary to find out which structure it is that we need to maintain. The VA for every structure is different and when comparing the structures should be of the same type. There are a total of 64 different structures as defined in the nen-norm.

2. **Create a multidisciplinary team within the organization of domain experts related to this type of maintenance.**

   Based on the structure to maintain, a team of experts needs to be formed that have knowledge on the maintenance of the structure as well as other aspects related to the structure. It is recommended to start with a large team and slowly narrow it down to a team of specialists, once the scope is known.

3. **The multidisciplinary team decides on which features to include in the database.**

   Together the multidisciplinary team can discuss on which features to include or exclude. It is expected that this takes several meetings and iterations.

4. **Create data classes by standardization of the features.**

   Now it is important to create classes for the defined features. The way these classes are defined by the domain experts as well as other computerized methods such as finding patterns in data and clustering the data based on these patterns.

5. **Make sure the data quality of each project is the same by creating a standard procedure on how to classify.**

   It is recommended to keep a data quality report to ensure that the data is always retrieved via the same method. This standardization does not only contribute to the objectivity of the final estimate but it also ensures that the

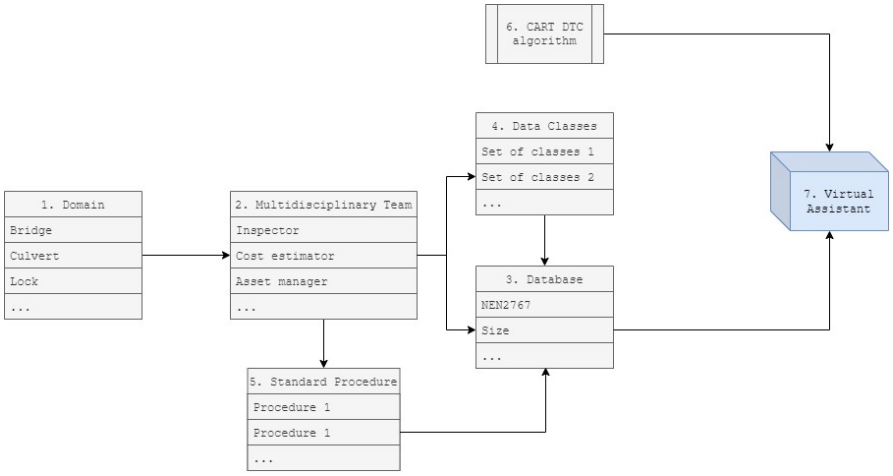


Figure 7.1: Process diagram industry use

comparison between the different structures is as complete as possible and no data is missing.

6. **Use the decision tree classification model as described in this research and run the model.**

   The findings of this research can help build the final model for the VA.

7. **You have successfully created a VA for maintenance budget estimation.**

BIBLIOGRAPHY

Boonamnuay, S., Kerdprasop, N., and Kerdprasop, K. (2018). Classification and regression tree with resampling for classifying imbalanced data. *International Journal of Machine Learning and Computing*, 8(4):336–340.

Elfaki, A. O., Alatawi, S., and Abushandi, E. (2014). Using intelligent techniques in construction project cost estimation: 10-year survey. *Advances in Civil Engineering*, 2014.

Evdorides, H. T., R. Kerali, H., Rivière, N., and Ørnskov, J. (2002). Condition-based method for programming road infrastructure maintenance. *Transportation research record*, 1816(1):10–15.

Figueroa, R. L., Zeng-Treitler, Q., Kandula, S., and Ngo, L. H. (2012). Predicting sample size required for classification performance. *BMC medical informatics and decision making*, 12(1):8.

Géron, A. (2017). Hands-on machine learning with scikit-learn & tensorflow.

Gorris, L. G. M. and Yoe, C. (2014). Risk analysis: risk assessment: principles, methods, and applications.

Haroun, A. E. (2015). Maintenance cost estimation: application of activity-based costing as a fair estimate method. *Journal of Quality in Maintenance Engineering*.

Hopgood, A. A. (2012). *Intelligent systems for engineers and scientists*. CRC press.

Humphreys, K. K. (1991). *Jelen's cost and optimization engineering*. McGraw-Hill Science, Engineering & Mathematics.

Kelleher, J. D., Mac Namee, B., and D'arcy, A. (2015). *Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies*. MIT press.

Kim, H.-J., Seo, Y.-C., and Hyun, C.-T. (2012). A hybrid conceptual cost estimating model for large building projects. *Automation in construction*, 25:72–81.

Kotsiantis, S. B., Zaharakis, I. D., and Pintelas, P. E. (2006). Machine learning: a review of classification and combining techniques. *Artificial Intelligence Review*, 26(3):159–190.

Lewis, R. J. (2000). An introduction to classification and regression tree (cart) analysis. In *Annual meeting of the society for academic emergency medicine in San Francisco, California*, volume 14.

Miettinen, K., Makela, M. M., Neittaanmaki, P., and Pkriaux, J. (1999). *Evolutionary algorithms in engineering and computer science: recent advances in genetic algorithms, evolution strategies, evolutionary programming, GE*. John Wiley & Sons, Inc.

Octrooicentrum Nederland (2008). Grond weg- en waterbouw: Bruggen en viaducten. Last accessed: 22.10.2020.

Raftery, J. (1987). The state of cost/modelling in the uk construction industry: a multi criteria approach. *Building Cost Modeling and Computers*, pages 49–71.

Rijkswaterstaat (2020). Bruggen. Last accessed: 22.10.2020.

Rokach, L. and Maimon, O. Z. (2008). *Data mining with decision trees: theory and applications*, volume 69. World scientific.

Sanford, K. L., Herabat, P., and McNeil, S. (1999). Bridge management and inspection data: Leveraging the data and identifying the gaps. In *8th International Bridge Management Conference*, pages 26–28. Transportation Research Board, National Research Council Washington, DC.

Scarf, P. A. (2007). A framework for condition monitoring and condition based maintenance. *Quality Technology & Quantitative Management*, 4(2):301–312.

Singh, S. and Gupta, P. (2014). Comparative study id3, cart and c4. 5 decision tree algorithm: a survey. *International Journal of Advanced Information Science and Technology (IJAIST)*, 27(27):97–103.

Srivastava, A. K., Kumar, G., and Gupta, P. (2020). Estimating maintenance budget using monte carlo simulation. *Life Cycle Reliability and Safety Engineering*, 9(1):77–89.

Staub-French, S., Fischer, M., Kunz, J., and Paulson, B. (2003). A generic feature-driven activity-based cost estimation process. *Advanced Engineering Informatics*, 17(1):23–39.

Sug, H. (2009). An effective sampling method for decision trees considering comprehensibility and accuracy. *W. Trans. on Comp*, 8(4):631–640.

Sugumaran, V. (1998). A distributed intelligent agent-based spatial decision support system. *AMCIS 1998 Proceedings*, page 136.

Sullivan, W. (2017). *1: Machine learning Beginners Guide Algorithms Supervised & Unsupervised learning, Decision Tree & Random Forest Introduction*. CreateSpace Independent Publishing Platform.

Susto, G. A., Schirru, A., Pampuri, S., McLoone, S., and Beghi, A. (2014). Machine learning for predictive maintenance: A multiple classifier approach. *IEEE Transactions on Industrial Informatics*, 11(3):812–820.

Tan, P.-N., Steinbach, M., and Kumar, V. (2006). Classification: basic concepts, decision trees, and model evaluation. *Introduction to data mining*, 1:145–205.

Wang, H.-J., Chiou, C.-W., and Juan, Y.-K. (2008). Decision support model based on case-based reasoning approach for estimating the restoration budget of historical buildings. *Expert Systems with Applications*, 35(4):1601–1610.

A APPENDIX A: DETERMINING CONDITION SCORE

As the defined in the NEN2767-4-1, a condition score is determined by the extent, intensity and severity of the defects found. In the descriptions below, the condition is described in general terms.

Condition score 1 - Excellent condition

- No or very limited aging
- Installations operate smoothly
- Defects are usually in the form of slight damage or of an aesthetic nature
- Repairs can be performed immediately and bring the building part back to the intended basic quality
- With regard to the overall appearance of defects, building components are in an excellent condition.

Condition score 2 - Good condition

- Incipient aging
- Installations operate nearly fault-free
- Defects to building components in the form of material degradation
- Defects, such as weathering symptoms, are only detected locally
- Building parts can have visible dirt infestation
- With regard to the overall appearance of defects, the building components can be considered as good.

Condition score 3 - Reasonable condition

- The aging process has started locally
- The functioning of the installations can be a single time disrupted without harming the business process
- Defects, in the form of weathering, etc., can occur locally or regularly
- Building parts show local defects to finishes, materials and components
- A building component may show a visible aging in its entirety.
- With regard to the overall appearance of the defects, the technical condition is qualified as reasonable. The quality of the materials used and/or the basic quality, detailing and execution can play a significant role in this.

Condition score 4 - Moderate condition

- The aging process is observed on a regular basis
- The operational reliability of installations is moderately guaranteed

- Building components regularly show defects in finishes, materials and components
- Locally, malfunctions in the functioning of the building component may act and regular (serious) defects may occur that can lead to loss of function
- Operating interruptions may occur
- With regard to the overall appearance of defects, the components are assessed as moderate. This can include are caused by errors in choice of materials, poor basic quality and/or execution.

Condition score 5 - Poor condition

- The ageing process has become more or less irreversible
- The functioning of the installations are no longer guaranteed. Many (serious) defects can occur that lead to loss of function. Business interruptions may occur on a regular basis.
- Building components exhibit to a considerable extent defects in finishes, materials and components
- functioning of building components is no longer guaranteed
- The overall appearance of faults in the components is poor. The cause may be: structural defects in the materials, the originally defective basic quality and/or the execution.

Condition score 6 - Very poor condition

- Maximum defect image
- The condition of construction parts is so bad that it can no longer be classified under Condition 5
- There is a maximum defect image and faults constantly occur in the function fulfillment of building components
- The building component is unusable and technically ripe for demolition

B APPENDIX B: DATABASE DECISION TREE CLASSIFIER

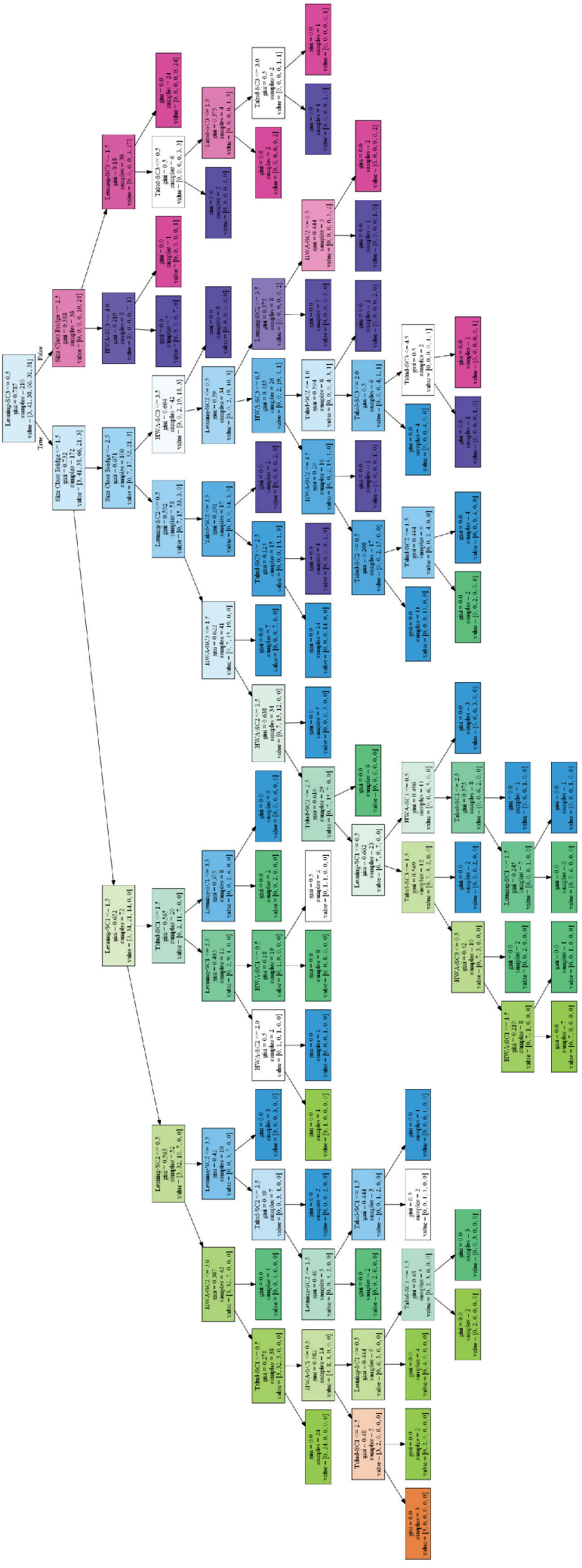
The following shows a snippet of the database used for the VA model. The full database is added to this thesis in a separate file.

Case	Tot duration	Size Class	Bridge	Leuning-SC1	Leuning-SC2	Leuning-SC3	Talud-SC1	Talud-SC2	Talud-SC3	HWA-SC1	HWA-SC2	HWA-SC3	Budget	Budget Class
1	25		1	1									3501	2
2	25		1	1									3217	2
3	25		1				1			5			4381	2
4	25		1				1						631	1
5	25		1							1			1404	2
6	25		1							1			807	2
7	25		1								1		2193	2
8	25		1								1		2146	2
9	25		1					1					1186	2
10	25		1					1					676	2
11	25		1		1								6508	3
12	25		1		1								5424	3
13	25		1	1			1			1			5687	2
14	25		1	1			1			1			4567	2
15	25		1	2				3			4		14369	4
16	25		1	1			2			4			6284	3
17	25		1							2			1702	2
18	25		1					1	1				929	2
19	25		1	1				1	1				5336	2
20	25		1	1				1	1				5174	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
290	25		3			1			1			1	95819	6
291	25		3									1	19232	4
292	25		3			2			1				99722	6
293	25		3					3				3	65898	5
294	25		3			1							60988	5
295	25		3			3						3	243756	6
296	25		3						3				28984	4

Figure B.1: Database VA model, sample size = 300

C APPENDIX C: DECISION TREE CLASSIFIER MODEL

In order to see a clear picture of the tree the code accompanied by this thesis can be copy & pasted to the website: http://www.webgraphviz.com/. There the tree will be generated to fit your pc screen.



D APPENDIX D: PYTHON CODE CLASSIFIER MODEL

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

[2]:
```
ds = pd.read_excel('DTC_voorbeeld.xlsx',sheet_name = 'LOD1')
```

[3]:
```
ds.shape
```

[3]:
```
(300, 14)
```

[4]:
```
ds.head()
```

[4]:

	Case	Tot duration	Size Class Bridge	Leuning-SC1	Leuning-SC2	\
0	1	25	1	1.0	NaN	
1	2	25	1	1.0	NaN	
2	3	25	1	NaN	NaN	
3	4	25	1	NaN	NaN	
4	5	25	1	NaN	NaN	

	Leuning-SC3	Talud-SC1	Talud-SC2	Talud-SC3	HWA-SC1	HWA-SC2	HWA-SC3	\
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2	NaN	1.0	NaN	NaN	5.0	NaN	NaN	
3	NaN	1.0	NaN	NaN	NaN	NaN	NaN	
4	NaN	NaN	NaN	NaN	1.0	NaN	NaN	

	Budget	Budget Class
0	3501.052960	2
1	3217.189524	2
2	4381.078144	2
3	630.963522	1
4	1403.593848	2

[5]:
```
ds_new = ds.fillna(0)
ds_new.head()
```

[5]:

	Case	Tot duration	Size Class Bridge	Leuning-SC1	Leuning-SC2	\
0	1	25	1	1.0	0.0	
1	2	25	1	1.0	0.0	
2	3	25	1	0.0	0.0	
3	4	25	1	0.0	0.0	
4	5	25	1	0.0	0.0	

	Leuning-SC3	Talud-SC1	Talud-SC2	Talud-SC3	HWA-SC1	HWA-SC2	HWA-SC3	\
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

```
2          0.0        1.0        0.0        0.0        5.0        0.0        0.0
3          0.0        1.0        0.0        0.0        0.0        0.0        0.0
4          0.0        0.0        0.0        0.0        1.0        0.0        0.0

         Budget  Budget Class
0  3501.052960              2
1  3217.189524              2
2  4381.078144              2
3   630.963522              1
4  1403.593848              2
```

[6]:
```
Y = ds_new['Budget Class ']
X = ds_new.drop(['Case', 'Tot duration', 'Budget', 'Budget Class '], axis = 1)
```

[7]:
```
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
```

[8]:
```
X_train,X_test,Y_train,Y_test = train_test_split( X, Y, test_size = 0.3,␣
↪random_state = 10)
```

[9]:
```
dt = DecisionTreeClassifier(criterion = 'gini', max_depth=11,␣
↪min_samples_split=2,random_state = 10)
```

[10]:
```
dt.fit(X_train,Y_train)
```

[10]:
```
DecisionTreeClassifier(max_depth=11, random_state=10)
```

[11]:
```
dt.score(X_train, Y_train)
```

[11]:
```
0.9904761904761905
```

[12]:
```
dt.score(X_test,Y_test)
```

[12]:
```
0.8888888888888888
```

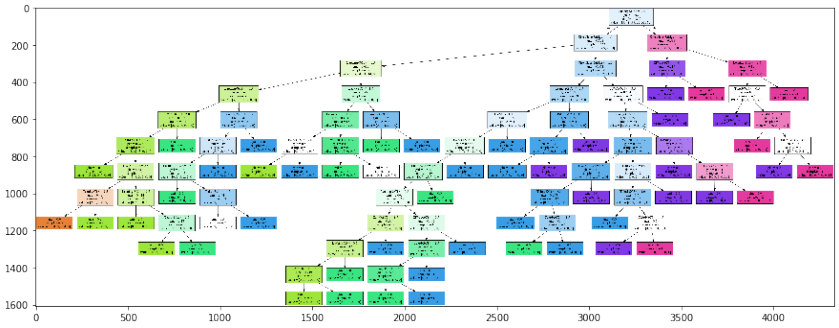
[13]:
```
from sklearn import tree
```

[14]:
```
dtc = tree.export_graphviz(dt, out_file = 'tree.dot', feature_names = X_train.
↪columns, max_depth = 20, filled = True)
```

[15]:
```
!dot -Tpng tree.dot -o tree.png
```

[16]:
```
image = plt.imread('tree.png')
plt.figure(figsize = (15,15))
plt.imshow(image)
```

[16]:
```
<matplotlib.image.AxesImage at 0x18ade297408>
```



[17]:
```
# Vul in:
# DtReg.predict( [[Size Class, Leuning-1, Leuning-2, Leuning-3, Talud-1,␣
 ↪Talud-2, Talud-3, HWA-1, HWA-2, HWA-3 ]] )


dt.predict([[ 1, 1, 0, 0, 0, 0, 0, 0, 0, 0 ]])
```

[17]:
```
array([2], dtype=int64)
```

[18]:
```
# OPTIONAL -> use if acc = 100%

# PRUNING

from sklearn.tree import DecisionTreeClassifier
path = dt.cost_complexity_pruning_path(X_train, Y_train)
ccp_alphas, impurities = path.ccp_alphas, path.impurities
```

[19]:
```
ccp_alphas
```

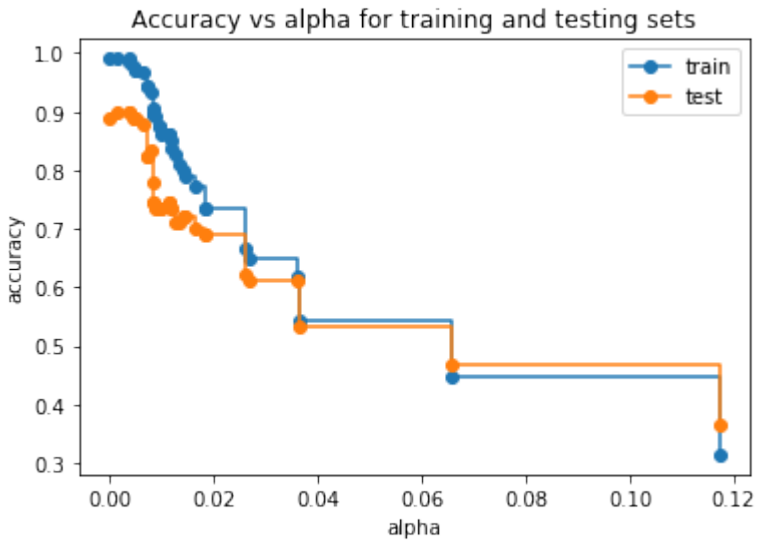
[19]:
```
array([0.        , 0.0015873 , 0.00357143, 0.00380952, 0.00455026,
       0.0047619 , 0.0047619 , 0.00634921, 0.00714286, 0.00714286,
       0.00793651, 0.00818071, 0.00833333, 0.00833333, 0.00888889,
       0.00952381, 0.00968254, 0.00986395, 0.01142857, 0.01160043,
       0.01166667, 0.01168831, 0.01253968, 0.01327188, 0.01428571,
       0.01464052, 0.01650794, 0.01821459, 0.01853074, 0.02607537,
       0.02694832, 0.03612782, 0.03627173, 0.0656428 , 0.11717572])
```

[20]:
```
clfs = []
for ccp_alpha in ccp_alphas:
    clf = DecisionTreeClassifier(random_state=0, ccp_alpha=ccp_alpha)
    clf.fit(X_train, Y_train)
    clfs.append(clf)
```

```
print("Number of nodes in the last tree is: {} with ccp_alpha: {}".format(
      clfs[-1].tree_.node_count, ccp_alphas[-1]))
```

```
Number of nodes in the last tree is: 1 with ccp_alpha: 0.11717571892077938
```

[21]:
```
train_scores = [clf.score(X_train, Y_train) for clf in clfs]
test_scores = [clf.score(X_test, Y_test) for clf in clfs]

fig, ax = plt.subplots()
ax.set_xlabel("alpha")
ax.set_ylabel("accuracy")
ax.set_title("Accuracy vs alpha for training and testing sets")
ax.plot(ccp_alphas, train_scores, marker='o', label="train",
        drawstyle="steps-post")
ax.plot(ccp_alphas, test_scores, marker='o', label="test",
        drawstyle="steps-post")
ax.legend()
plt.show()
```



[22]:
```
clf = DecisionTreeClassifier(random_state=0, ccp_alpha=0.02)
clf.fit(X_train,Y_train)
```

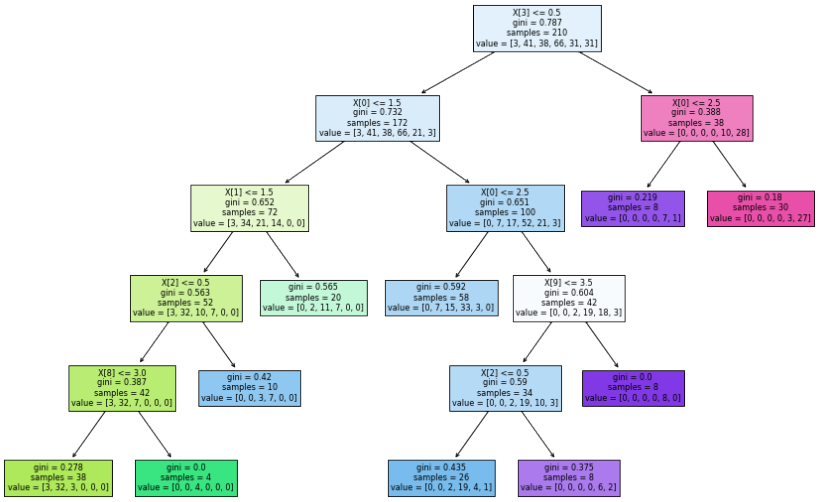
[22]:
```
DecisionTreeClassifier(ccp_alpha=0.02, random_state=0)
```

[23]:
```
pred=clf.predict(X_test)
from sklearn.metrics import accuracy_score
accuracy_score(Y_test, pred)
```

[23]:
```
0.6888888888888889
```

[24]:
```
plt.figure(figsize=(15,10))
tree.plot_tree(clf,filled=True)
```

[24]:
```
[Text(547.2692307692308, 498.3, 'X[3] <= 0.5\ngini = 0.787\nsamples = 210\nvalue
= [3, 41, 38, 66, 31, 31]'),
 Text(386.3076923076923, 407.70000000000005, 'X[0] <= 1.5\ngini = 0.732\nsamples
= 172\nvalue = [3, 41, 38, 66, 21, 3]'),
 Text(257.53846153846155, 317.1, 'X[1] <= 1.5\ngini = 0.652\nsamples = 72\nvalue
= [3, 34, 21, 14, 0, 0]'),
 Text(193.15384615384616, 226.5, 'X[2] <= 0.5\ngini = 0.563\nsamples = 52\nvalue
= [3, 32, 10, 7, 0, 0]'),
 Text(128.76923076923077, 135.89999999999998, 'X[8] <= 3.0\ngini =
0.387\nsamples = 42\nvalue = [3, 32, 7, 0, 0, 0]'),
 Text(64.38461538461539, 45.299999999999955, 'gini = 0.278\nsamples = 38\nvalue
= [3, 32, 3, 0, 0, 0]'),
 Text(193.15384615384616, 45.299999999999955, 'gini = 0.0\nsamples = 4\nvalue =
[0, 0, 4, 0, 0, 0]'),
 Text(257.53846153846155, 135.89999999999998, 'gini = 0.42\nsamples = 10\nvalue
= [0, 0, 3, 7, 0, 0]'),
 Text(321.9230769230769, 226.5, 'gini = 0.565\nsamples = 20\nvalue = [0, 2, 11,
7, 0, 0]'),
 Text(515.0769230769231, 317.1, 'X[0] <= 2.5\ngini = 0.651\nsamples = 100\nvalue
= [0, 7, 17, 52, 21, 3]'),
 Text(450.69230769230774, 226.5, 'gini = 0.592\nsamples = 58\nvalue = [0, 7, 15,
33, 3, 0]'),
 Text(579.4615384615385, 226.5, 'X[9] <= 3.5\ngini = 0.604\nsamples = 42\nvalue
= [0, 0, 2, 19, 18, 3]'),
 Text(515.0769230769231, 135.89999999999998, 'X[2] <= 0.5\ngini = 0.59\nsamples
= 34\nvalue = [0, 0, 2, 19, 10, 3]'),
 Text(450.69230769230774, 45.299999999999955, 'gini = 0.435\nsamples = 26\nvalue
= [0, 0, 2, 19, 4, 1]'),
 Text(579.4615384615385, 45.299999999999955, 'gini = 0.375\nsamples = 8\nvalue =
[0, 0, 0, 0, 6, 2]'),
 Text(643.8461538461538, 135.89999999999998, 'gini = 0.0\nsamples = 8\nvalue =
[0, 0, 0, 0, 8, 0]'),
 Text(708.2307692307693, 407.70000000000005, 'X[0] <= 2.5\ngini = 0.388\nsamples
= 38\nvalue = [0, 0, 0, 0, 10, 28]'),
 Text(643.8461538461538, 317.1, 'gini = 0.219\nsamples = 8\nvalue = [0, 0, 0, 0,
7, 1]'),
 Text(772.6153846153846, 317.1, 'gini = 0.18\nsamples = 30\nvalue = [0, 0, 0, 0,
3, 27]')]
```



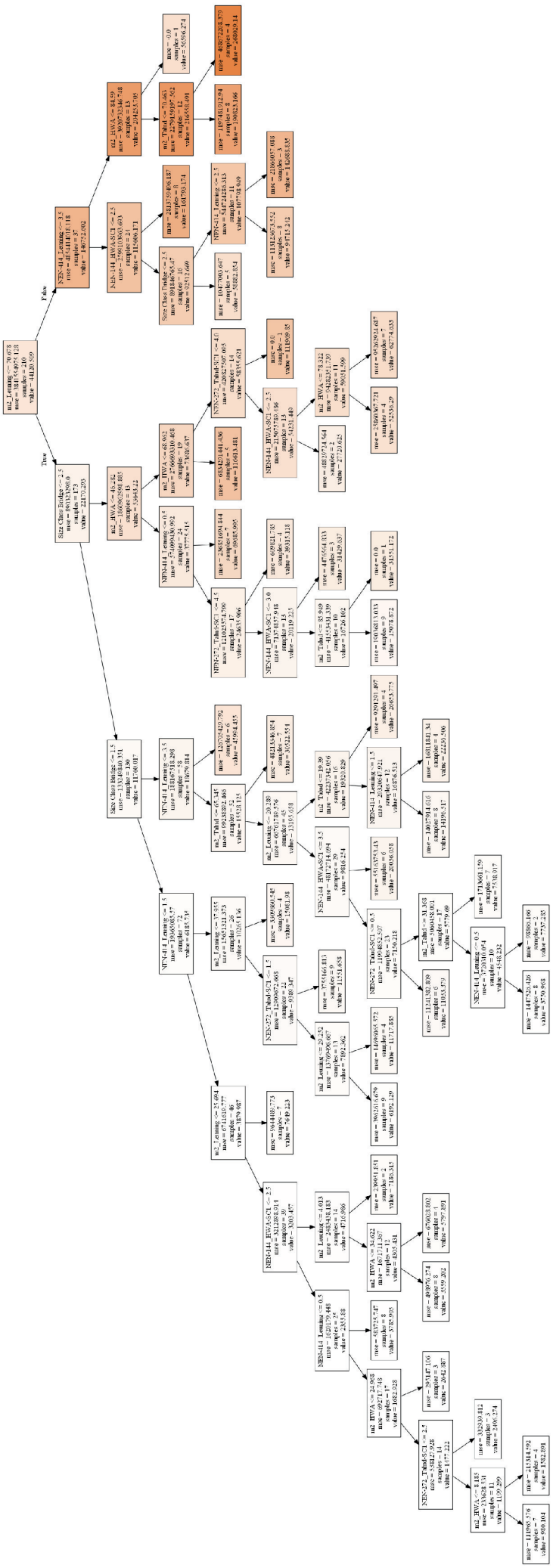
[25]:
```
#plot confusion matrix() will run the test data down the tree and draw a␣
 ↪confusion matrix

from sklearn.datasets import make_classification
from sklearn.metrics import plot_confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC


a = plot_confusion_matrix(dt, X_test, Y_test)
# you can see 5/10 are correctly classified which is 50%
```

E APPENDIX E: DECISION TREE REGRESSOR

This appendix shows the database for the VA using the decision tree regressor and the model output.

Case	Tot duration	Size Class Bridge	NEN-414_Leuning	m2_Leuning	NEN-272_Talud-SC1	m2_Talud	NEN-144_HWA-SC1	m2_HWA	**Budget**
1	25	1	1	23					3314
2	25	1	1	23					3411
3	25	1			1	48	5	11	4377
4	25	1			1	28			663
5	25	1					1	24	1027
6	25	1					1	25	903
7	25	1					1	46	2235
8	25	1					1	16	2285
9	25	1			1	40			1266
10	25	1			1	48			1347
...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...
290	25	3	1	81	1	63	1	87	96071
291	25	3					1	78	20733
292	25	3	2	76	1	84			94727
293	25	3			3	22	3	79	71598
294	25	3	1	60					64036
295	25	3	3	82			3	77	231257
296	25	3			3	88			31551
297	25	3			5	80			38929
298	25	3			5	77			39290
299	25	3			5	74			39318
300	25	3			5	83			40571

Figure E.1: Database VA model, sample size = 300



F APPENDIX F: PYTHON CODE DECISION TREE REGRESSOR

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

[2]:
```
ds = pd.read_excel('DTC_voorbeeld.xlsx',sheet_name = 'LOD2')
```

[3]:
```
ds.shape
```

[3]:
```
(300, 10)
```

[4]:
```
ds.head()
```

[4]:

	Case	Tot duration	Size Class Bridge	NEN-414_Leuning	m2_Leuning	NEN-272_Talud-SC1	m2_Talud	NEN-144_HWA-SC1	m2_HWA	Budget
0	1	25	1	1.0	23.011598	NaN	NaN	NaN	NaN	3313.741068
1	2	25	1	1.0	22.518598	NaN	NaN	NaN	NaN	3411.330344
2	3	25	1	NaN	NaN	1.0	47.714599	5.0	10.542897	4376.799959
3	4	25	1	NaN	NaN	1.0	27.916250	NaN	NaN	662.699204
4	5	25	1	NaN	NaN	NaN	NaN	1.0	23.873793	1026.568586

[5]:
```
ds_new = ds.fillna(0)
ds_new.head()
```

[5]:

	Case	Tot duration	Size Class Bridge	NEN-414_Leuning	m2_Leuning	NEN-272_Talud-SC1	m2_Talud	NEN-144_HWA-SC1	m2_HWA	Budget
0	1	25	1	1.0	23.011598	0.0	0.000000	0.0	0.000000	3313.741068
1	2	25	1	1.0	22.518598	0.0	0.000000	0.0	0.000000	3411.330344
2	3	25	1	0.0	0.000000	1.0	47.714599	5.0	10.542897	4376.799959
3	4	25	1	0.0	0.000000	1.0	27.916250	0.0	0.000000	662.699204
4	5	25	1	0.0	0.000000	0.0	0.000000	1.0	23.873793	1026.568586

[6]:
```
Y = ds_new.iloc[:, 9].values
X = ds_new.iloc[:, 2:9].values
```

[7]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test = train_test_split( X, Y, test_size = 0.3,␣
 ↪random_state = 10)
```

[8]:
```
from sklearn.tree import DecisionTreeRegressor

DtReg = DecisionTreeRegressor(max_depth=20, min_samples_split=10)

DtReg.fit(X_train, Y_train)
```

[8]:
```
DecisionTreeRegressor(max_depth=20, min_samples_split=10)
```

[9]:
```
Y_predict_dtr = DtReg.predict((X_test))

#Model evaluation using R-square for DTR
from sklearn import metrics
r_square = metrics.r2_score(Y_test, Y_predict_dtr)

print('R-Square Error associated with Decision Tree Regressor is:', r_square)
```

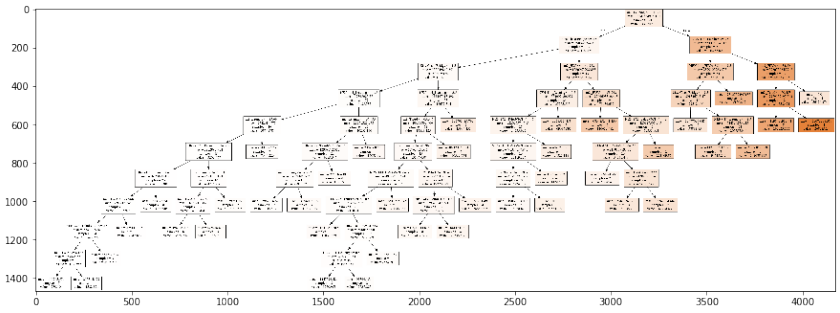
```
R-Square Error associated with Decision Tree Regressor is: 0.6755756006403919
```

[10]:
```
from sklearn.tree import export_graphviz

dtc = export_graphviz(DtReg, out_file = 'dtregtree.dot', feature_names = ds_new.
 ↪columns[2:9], max_depth = 20, filled = True)
```

[11]:
```
!dot -Tpng dtregtree.dot -o dregtree.png
```

[12]:
```
image = plt.imread('dregtree.png')
plt.figure(figsize = (15,15))
plt.imshow(image)
```

[12]:
```
<matplotlib.image.AxesImage at 0x212b7dc5808>
```



[13]:
```
export_graphviz(DtReg, out_file = 'dtregressor.dot',
                feature_names = ds_new.columns[2:9])
```

[14]:
```
DtReg.predict( [[1, 1, 22, 0, 0, 0, 0]] )
```

[14]:
```
array([3785.90517025])
```

G APPENDIX G: PAIRPLOT SYNTHETIC DATA

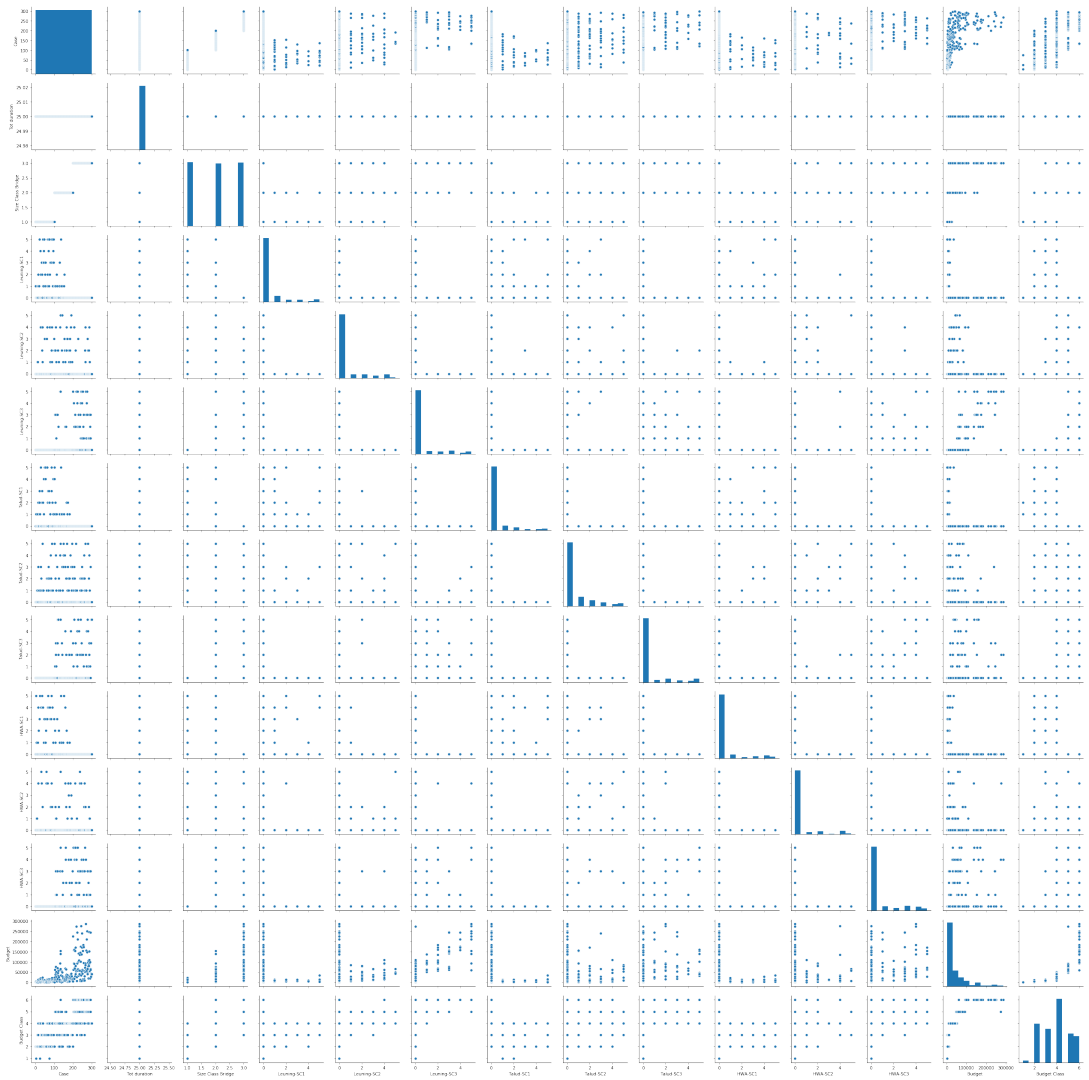


Figure G.1: Pairplot Synthetic Data VA model, sample size = 300

H | APPENDIX H: EXPERT REVIEW

This chapter shows an overview of the findings of the expert review conducted at BAM infra. The expert review was conducted through demonstrating the VA model and asking for future remarks. The VA model was presented in two levels of detail: VA that predicts a budget range (LOD1) and a VA that predicts the final budget (LOD2). The team of experts at BAM consists of: an asset manager, a cost estimator and a project manager for inspections.

All experts were asked for their feedback on the VA model by explaining their point of view on the VA's applicability to current practice, the feasibility of creating a VA and other remarks.

- ***Applicability VA***

  How well do you think the VA is working?

- ***Feasibility VA***

  How do you think a working VA can be realized, given the development objectives?

H.0.1 Cost Estimator

Applicability VA

The VA seems easy to use at first glance. The functions to fill in are preset which makes it easy to use. I would not opt for an independent prediction, which in theory this seems to be possible. The database allows for retrieving past information which at the moment is not there in this structured manner.

Feasibility VA

The VA is feasible when using LOD 1: predicting a budget class. This is because it works for a rough estimate for the first stages of the project, but it is dangerous to use for a final budget price (*inschrijfprijs*). This has less to do with the model and more to do with the way data is stored at the moment within the organization. Therefore the prediction of an exact price is not applicable but prediction of a budget class would be. The human still needs to be involved in the cost estimation process. A reference to prove this point: within our organization the section asphalt already uses intelligent systems for predicting prices. In one case 5 models were used to create a budget for asphalt costs and because of the presence of the models the cost estimator simply used these models to predict the budget without also predicting a price himself. This led to budget that was not realistic, because apparently there were other factors that influenced the budget as well that were not taken into account in the model. For this reason the human needs to be involved in the budget estimation procedure and the model can be of assistance for verification. With this assistance it would also be ideal if there would be a bandwidth of accuracy of the estimate. Furthermore the VA can be feasible if the way data is documented is standardized. The way in this case the data is divided into classes is something that

needs to be looked at within the organization. How these classes are determined and the possibility to add more features need to be discussed within a multidisciplinary team. Also the way the data is delivered needs to be the same exact way, for it to be stored easily in the database. These are arrangements that need to be communicated not only in the cost estimation branch but within all branches that have anything to do with the estimation of budgets for bridges. Only with a multidisciplinary team there can be a template made for the database by looking what need to be included and standardized.

H.0.2 Project Manager Inspections

Applicability VA

This model seems to have a good potential for future use. The model's name is also fitting for its purpose. It is a VA: an assistant for the cost estimator and not a replacement. Therefore from the perspective of ease of use it is very comfortable to use but we should not disregard the cost estimators opinion.

Feasibility VA

The model is probably going to work well once realized, provided we introduce standardization in the way of working. There needs to be more standardization and more clearer standards to generate objective data. So when an inspector is looking at a bridge and classifies an element in class 2, it needs to be also classified by another inspector in the same category. Therefore the judgement should be the same. This way the input data will be more reliable. Once we have good quality of data, a standardized way of working, we might even replace the cost estimator with a model. Until that time the right measures need to be taken to come to an initial budget range. Therefore the VA is feasible if the quality of the data is documented as well as standardization is introduced. There is already a digital strategy for the future in the works and this model really shows the potential of the use of machine learning, decision tree methodology.

H.0.3 Asset Manager

Applicability VA

In the contracting world there is a need to use digitization and intelligent tools in order to use the assets in the most efficient way. At the moment BAM already uses tools as such for smaller works. The VA, on this conceptual level, shows that machine learning can be applicable as an intelligent tool. It also shows that intelligent tools can also be used for projects with a larger. The tool seems easy to use and if data keeps being stored in the right manner it has a potential to be changed from an assistant for the cost estimator to an actual estimator. If we know the outcomes are reliable then it very efficient because of its ease of use and standard input features.

Feasibility VA

There is the need for a multidisciplinary team and more communications with the company tho realize this model. Different disciplines have different ideas on what is most important to include. It is also necessary to sit with disciplines that already make use of machine learning or other intelligent systems to use their expertise as a comparison. This model is a good first step to provide insight on where to start with the composition of the multidisciplinary team.

I APPENDIX I: COMPARISON VA & HUMAN ESTIMATION

Case 1: "The VA does not take into account the average but looks at all the data and classifies where the prediction fits best"

Proof 1:
See Fig. I.1.

Case 2: "The VA considers, besides the same exact case, also similar cases to classify where the prediction fits best"

Proof 2:
See Fig. I.2.

Case 3: "The VA can predict never seen before cases by classifying into already known tree"

Proof 3:
See Fig. I.3.

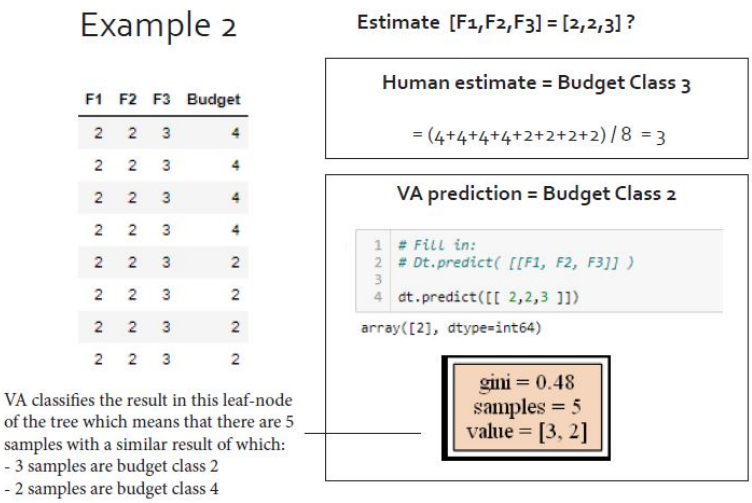


Figure I.1: Proof 1b

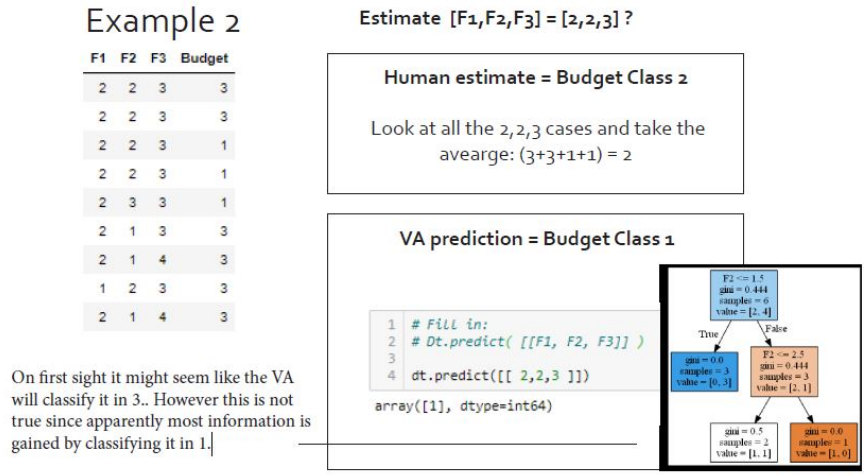


Figure I.2: Proof 2b

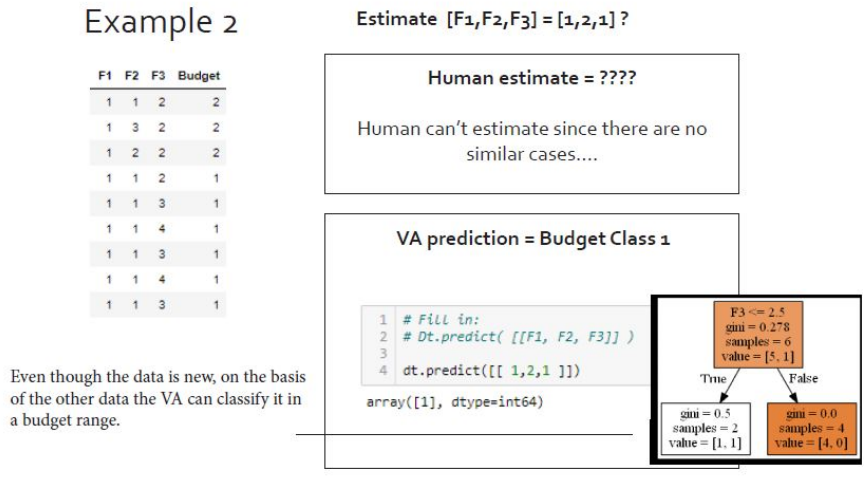


Figure I.3: Proof 3b

COLOPHON

This document was typeset using LaTeX. The document layout was generated using the arsclassica package by Lorenzo Pantieri, which is an adaption of the original classicthesis package from André Miede.