# Enhanced Model Tree Application Framework for Developing Interpretable AI in Construction Engineering

by

Serhii NAUMETS

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science

in

Construction Engineering and Management

Department of Civil & Environmental Engineering

University of Alberta

# *Abstract*

The construction industry has been and continues to be overflown with data. Scholars have no problems dealing with this phenomenon through the incorporation of artificial intelligence (AI) methods like neural networks or random forests. However, when the time comes to practical application, the industry professionals show very little interest in these predictive models. Most of the best-performing methods are too complex and packed in "black boxes". In my view, the user's trust in a computer program is analogous to the user's trust in a co-worker: if there is no understanding—there is no trust, if there is no trust—there is no cooperation.

In collaboration with a steel fabrication company in western Canada, this research investigated the cost estimation department in regards to preparing pre-bid estimates. I found that most of the estimators fall under the "baby boomer" cohort. In my view, it was imperative to capture their experience and know-how before they retire and pass it on to the next generation of engineers and managers. Another finding showed that the professionals in this company were not eager to use AI techniques. They needed something that could be easily interpreted and trusted.

A data set sourced from this steel fabrication company was used to compare various AI algorithms and search for candidate for interpretable AI. Firstly, I identified interpretable performance metrics the meaning of which can be easily explained to a user. Secondly, these metrics were put together in a color scheme that could help to decide on the credibility of the AI model.

As a testing case study, I used the Compressive Concrete Strength data set (Yeh, 1998a) to illustrate that the developed framework could build an interpretable AI model in a different problem domain. Linear regressions provided

by the Model Tree can serve as formula sheets to customize concrete mix or to calculate the compressive strength of concrete at a certain point of curing.

After the comparison of Artificial Neural Network, Support Vector Machine, Random Forest, and Model Tree, the last was determined as a potential candidate to generate interpretable AI for practical applications. The enhanced M5P algorithm with three-colored performance scheme has no analogous concepts and functions in any existing software.

As supporting material, Appendix C provides a manual of how to setup a Model Tree in WEKA and Appendix D contains the configurations of all of the discussed AI models.

# Preface

I, Serhii NAUMETS, declare that this thesis titled, "Enhanced Model Tree Application Framework for Developing Interpretable AI in Construction Engineering" and the work presented in it are my own. The content is based on the following research papers:

1) Naumets, S., and Lu, M. (2020). "Using Model Trees to Represent Knowhow of Experienced Estimators in Steel Fabrication Industry." *ASCE Construction Research Congress*. Phoenix, AZ. In press.

2) Naumets, S., and Lu, M. (2020). Interpretable Artificial Intelligence Models for Estimating Steel Fabrication Projects. *Journal of Construction Engineering and Management*. Submitted for revision.

I confirm that:

- This work was done wholly while in candidature for a master degree at the University of Alberta.
- Where any part of this thesis has previously been submitted for a publication or any other institution, this has been clearly stated.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.

Date: August 12, 2020

*Dedicated to my supervisor and mentor Dr. Ming Lu.*
*Thank you for giving me a chance.*

*"Imagination is more important than knowledge. Knowledge is limited. Imagination encircles the world."*

Albert Einstein

# *Acknowledgements*

I had a privilege to be funded by the National Science and Engineering Research Council and Supreme Group through a Collaborative Research and Development grant.

Great appreciation to Dalip Prasad, Rakesh Sabharwal, and Arash Mohsenijam for contributing their time and precious know-how.

Separate gratitude to Dr. Evan Davies, Dr. Yuxiang Chen, Dr. Simaan Abourizk, and Dr. Ming Lu for the comments and feedback, which helped me improve the organization and clarity of this thesis.

Thank you to my family and friends. Without you, there is no me.

# Contents

**Abstract** **ii**

**Preface** **iv**

**Acknowledgements** **vii**

**List of Tables** **xi**

**List of Figures** **xii**

**List of Abbreviations** **xiv**

**1 Introduction** **1**

1.1 Research Motivation . . . . . . . . . . . . . . . . . . . . . . . . . 1

1.1.1 Demographic situation . . . . . . . . . . . . . . . . . . . . . . 1

1.1.2 Estimators' Know-How in Steel Fabrication . . . . . . . . . 2

1.2 Problem Statement . . . . . . . . . . . . . . . . . . . . . . . . . . 4

1.2.1 Explainable vs. Interpretable AI . . . . . . . . . . . . . . . . 4

1.3 Research Objective . . . . . . . . . . . . . . . . . . . . . . . . . . 6

1.4 Thesis Structure . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

**2 Literature Review** **8**

2.1 Predictive Methods . . . . . . . . . . . . . . . . . . . . . . . . . . 8

**3 Methods** **13**

3.1 Data Collection . . . . . . . . . . . . . . . . . . . . . . . . . 13

3.2 Performance Metrics . . . . . . . . . . . . . . . . . . . . . . . 15

3.2.1 Absolute errors . . . . . . . . . . . . . . . . . . . . . . 15

3.2.2 Relative errors . . . . . . . . . . . . . . . . . . . . . . 16

3.2.3 R-squared . . . . . . . . . . . . . . . . . . . . . . . . . 16

3.2.4 Pearson Correlation Coefficient . . . . . . . . . . . . . . . 18

3.3 How M5P Works . . . . . . . . . . . . . . . . . . . . . . . . . 19

3.3.1 Growing the initial tree . . . . . . . . . . . . . . . . . . 19

3.3.2 Pruning . . . . . . . . . . . . . . . . . . . . . . . . . . 21

3.3.3 Smoothing . . . . . . . . . . . . . . . . . . . . . . . . . 22

3.4 Using WEKA for Building Predictive Models . . . . . . . . . . . 23

**4 Results** **24**

4.1 Attribute Selection . . . . . . . . . . . . . . . . . . . . . . . . 24

4.2 Performance of Each Model . . . . . . . . . . . . . . . . . . . . 25

**5 Discussion** **27**

5.1 Comparison of the Models' Interpretability . . . . . . . . . . . . 27

5.1.1 ANN . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27

5.1.2 SVM . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28

5.1.3 RF . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29

5.1.4 M5P . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30

5.2 3-Colored Scheme for M5P . . . . . . . . . . . . . . . . . . . . 33

**6 Practical Test Case** **37**

**7 Conclusion** **43**

7.1 Findings . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 43

7.2 Industry Contribution . . . . . 44
7.3 Academic Contribution . . . . . 45
7.4 Next Steps . . . . . 46

**Bibliography** **47**

**A Regressions for Concrete test case** **51**
A.1 Linear regression 1 . . . . . 51
A.2 Linear regression 2 . . . . . 52
A.3 Linear regression 3 . . . . . 53
A.4 Linear regression 4 . . . . . 54
A.5 Linear regression 5 . . . . . 55
A.6 Linear regression 6 . . . . . 56
A.7 Linear regression 7 . . . . . 57
A.8 Linear regression 8 . . . . . 58
A.9 Linear regression 9 . . . . . 59
A.10 Linear regression 10 . . . . . 60

**B Steel fabrication dataset sample** **61**

**C WEKA Concrete example setup manual** **64**

**D WEKA model setups from Chapter 5** **72**
D.1 ANN setup . . . . . 72
D.2 SVM setup . . . . . 76
D.3 RF setup . . . . . 80
D.4 M5P setup . . . . . 84

# List of Tables


3.1 Steel data set attributes . . . . . . . . . . . . . . . . . . . . . . . . 14

4.1 Selected attributes . . . . . . . . . . . . . . . . . . . . . . . . . . . 25

4.2 10-fold cross-validation . . . . . . . . . . . . . . . . . . . . . . . . 26

5.1 Leaf training performance . . . . . . . . . . . . . . . . . . . . . . . 33

6.1 HPC dataset attributes . . . . . . . . . . . . . . . . . . . . . . . . 38

6.2 Concrete example training and cross-validation performance . . . 38

6.3 Concrete example only "Green" leaves performance . . . . . . . . 39

B.1 Steel dataset characteristics . . . . . . . . . . . . . . . . . . . . . . 61

B.2 Steel dataset sample . . . . . . . . . . . . . . . . . . . . . . . . . . 62

B.3 Steel dataset sample continuation . . . . . . . . . . . . . . . . . . . 63

# List of Figures


1.1 The projection of Canadian population . . . 2

2.1 Abstract illustration of ANN and SVM . . . 9
2.2 Abstract illustration of Decision Tree . . . 10
2.3 Abstract illustration of Random Forest . . . 11
2.4 Abstract illustration of Model Tree (M5P) . . . 12

3.1 Pearson Correlation Coefficient . . . 19
3.2 Splitting Criteria calculation example . . . 20
3.3 Smoothing of Predicted Value . . . 22

5.1 ANN correlation scatter-plot . . . 28
5.2 SVM correlation scatter-plot . . . 29
5.3 RF correlation scatter-plot . . . 30
5.4 M5P correlation scatter-plot . . . 31
5.5 M5P tree . . . 32
5.6 M5P tree with correlation graphs for each node . . . 34
5.7 Leaf leave-one-out cross-validation performance . . . 35
5.8 Revised M5P tree . . . 36

6.1 Concrete example model . . . 40
6.2 Concrete example leave-one-out cross-validation performance . . . 40
6.3 Revised concrete example model . . . 41

6.4 Strength development of concretes at different water-cementitious materials ratios . . . 42

C.1 Applications window . . . 64
C.2 Loading a dataset . . . 65
C.3 Attribute features . . . 66
C.4 Loading an algorithm . . . 67
C.5 Setting the algorithm features . . . 68
C.6 Training a model . . . 69
C.7 Classifier error visualization . . . 70
C.8 Tree visualisation . . . 71

D.1 WEKA ANN preprocess . . . 72
D.2 WEKA ANN performance . . . 73
D.3 WEKA ANN setup . . . 74
D.4 WEKA ANN regression error . . . 75
D.5 WEKA SVM preprocess . . . 76
D.6 WEKA SVM performance . . . 77
D.7 WEKA SVM setup . . . 78
D.8 WEKA SVM regression error . . . 79
D.9 WEKA RF preprocess . . . 80
D.10 WEKA RF performance . . . 81
D.11 WEKA RF setup . . . 82
D.12 WEKA RF regression error . . . 83
D.13 WEKA M5P preprocess . . . 84
D.14 WEKA M5P performance . . . 85
D.15 WEKA M5P setup . . . 86
D.16 WEKA M5P regression error . . . 87

# List of Abbreviations

| | |
|---|---|
| **AI** | **A**rtificial **I**ntelligence |
| **NN** | **N**eural **N**etwork(s) |
| **ANN** | **A**rtificial **N**eural **N**etwork(s) |
| **SVM** | **S**upport **V**ector **M**achine(s) |
| **RF** | **R**andom **F**orest(s) |
| **DARPA** | **D**efense **A**dvanced **R**esearch **P**roject **A**gency |
| **XAI** | e**X**plainable **A**rtificial **I**ntelligence |
| **CART** | **C**lassification **A**nd **R**egression **T**ree(s) |
| **PCC** | **P**earson **C**orrelation **C**oefficient |
| **HPC** | **H**igh **P**erformance **C**oncrete |

# Chapter 1

# Introduction

## 1.1 Research Motivation

### 1.1.1 Demographic situation

The baby boomers – that massive bubble of people born in the two decades following the Second World War, perhaps the most important cohort for economical, technological and social development in human history – have begun the transition into old age (Calabrese, 2015). According to Figure 1.1 (Canada, 2017), the projection of the population aged 65 years and older starting from 2015 does not look promising for the current generation. In my view, each and every year more and more professionals will be retiring together with their valuable experience and knowledge. It may affect every industry including construction and many countries not limited to Canada. Darren Calabrese (2015) suggested that some of the impacts of Canada's ageing work-force can be mitigated by its relatively large immigration program: about 250,000 new immigrants arrive in Canada each year, roughly double of the country's natural growth through births and deaths. While it is relatively easy to replace the work-force, it is more complicated to save the experience of retiring experts.

I was privileged to take part in Collaborative Research Development with

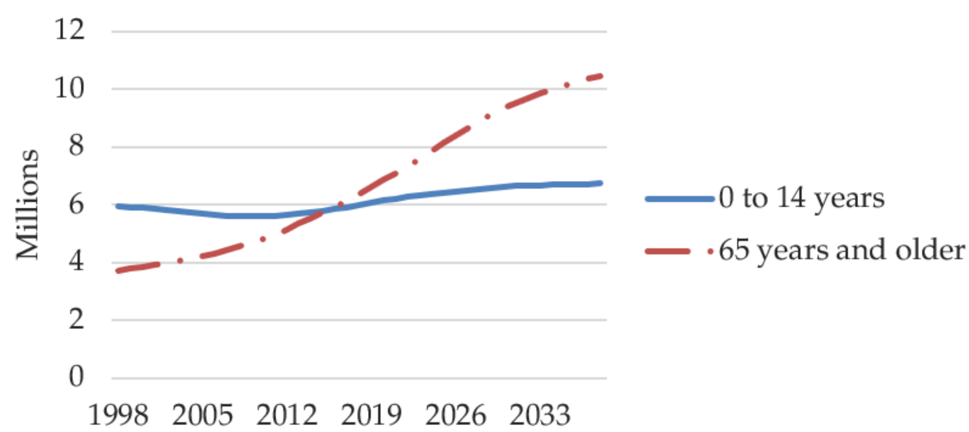


FIGURE 1.1: The projection of Canadian population 1998 to 2038

one of the biggest steel fabrication companies in western Canada. The team I was part of closely interacted with company's critical cost estimation department. After a comprehensive investigation of its age group, I found that at least half of all the estimators fall under the "baby boomer" cohort. Thus, I decided to study the techniques to convert raw estimation data into meaningful knowledge and save it for the company's newcomers and professionals in training.

### 1.1.2 Estimators' Know-How in Steel Fabrication

From my observations, the cost estimating department is always engaged in a high-intensity, never-ending "battle" in the bidding process. Firstly, for a potential new project, estimators along with executives must make a critical decision whether "to bid" or "not to bid". It requires completing a feasibility estimate to know the rough number of labour hours by the fabrication shop and approximate the total price of the project. Secondly, if the decision is made "to bid", estimators break the project down to packages and initiate the take-off process. And it takes time. For the steel fabricator, conceptual estimating entails the skills to review performance specifications and the footprint of a structure and develop a budget for all the activities related to steel fabrication, namely: detailing, fabricating, painting, transporting and erecting (Liddy and Cross, 2002).

It is noteworthy that two critical components are missing from the list above: material cost and buyouts/subouts[1] cost. While the unit cost for materials is relatively constant, the cost for the buyouts/subouts category is highly unpredictable. I suppose that this uncertainty considerably decreases the correlation between the attributes of the project and the estimated total project cost. That is why I chose the total labor-hours required to fabricate the project as an output for the current study with the consideration of the fact that labor-hours data show less variation than the recorded labor cost in dollars, which is susceptible to inflation and time-dependent labor rate and exchange rate fluctuation.

Through embedded graduate student training based in the estimating department of the industry partner, estimators' decision process is thoroughly studied in terms of how the total labor-hours are predicted by experience. Firstly, total weight, the total length of each piece and the total quantity of pieces on a given steel fabrication job are identified. Secondly, the ratio of total weight over total length is multiplied by a certain factor, which can be biased and fully depends on estimator's experience. I used this logic as a baseline to compare against the model results.

Liddy (2002) describes an old rule of thumb: "no estimate is ever forgotten!". Each estimate needs to be fully documented and retained. The challenge for the steel fabricator is to track past project costs and organize them in a way that allows the creation of an accurate conceptual estimate in a minimal amount of time. Unfortunately, it is not exactly true in practice. The lion's share of the

[1]Buyouts/subouts refer to miscellaneous parts of the structure which are not typical for conventional steel fabrication or the work that cannot be handled in the contractor's steel shop.

conceptual bid estimates does not go through to win the bid. As a result, lessons learned, and success factors experienced would often be left in the shared folders or databases. Certainly, estimators can recall most, if not all, estimates they have ever done themselves and use the experience to benchmark and inform on bidding new projects. But what would happen after the experienced estimator retires? How would a novice estimator take over this challenging task without learning from scratch? To address such age-old questions provides one of the main motivations for conducting this research.

Another long-standing problem of data insufficiency served as an additional motivator. From the words of Dr. Ming Lu (personal communication, July 22, 2020), "The data quality in the construction industry is much the same as twenty years ago and far from ideal; there is a need for developing AI methods to accommodate the imperfect data and solve the problem that is defined based on such data".

## 1.2 Problem Statement

### 1.2.1 Explainable vs. Interpretable AI

Machine learning algorithms for data-driven predictive analytics, including neural networks (NN), support vector machines (SVM), random forests (RF), have been widely utilized by researchers in the past couple of decades. According to the recent discussion by Frank Emmert-Streib et al. (2020) such statistical models and machine learning methods have been introduced due to the lack of general theories outside of physics as they allow a quantitative analysis of experimental evidence. In the past, this experimental evidence could only be produced in the laboratories. Nowadays, enabled with technological advancement, this so-called evidence is falling on us from everywhere. Nearly every domain is overflowed with data surge which might have created an impression that every research should start with data collection and end with AI application. The construction realm is surely one of them. Hojjat Adeli (2001) conducted a review of the journal of Computer-Aided Civil and Infrastructure Engineering from 1989 (first publication on NN topic) to 2000 and found over one hundred and eighty NN use cases, not counting alternative stand-alone algorithms like decision tree or fuzzy logic. A recent review paper by Preeti Kulkarni (2017) described over seventy NN applications in construction management alone. While scholars keep widening the boundaries of what machines can learn, practitioners do not go hand in hand. Many of the best performing methods feature highly complex mathematical algorithms, prohibiting a straightforward explanation of the obtained results in simple terms (Emmert-Streib, Yli-Harja, and Dehmer, 2020). For professionals who make high-stake decisions, these explanations are worth their weight in gold. The user's trust in a computer program is analogous to the user's trust in a co-worker: if there is no understanding—there is no trust, if there is no trust—there is no cooperation. Addressing this issue led the United States Defense Advanced Research Projects Agency (DARPA) to initiate a new field called Explainable Artificial Intelligence (XAI) (Gunning, 2016). Nevertheless, as per XAI strategies, developing and validating the second "non-black-box" model, which is built to describe the "black box" of the initial model, presents a special challenge: if the explanation is completely faithful to what the original model computes, the explanation would equate with the "black box" model, as such, one would not need the original model in the first place, but the explanation. In other words, this is a case where the original model would be **interpretable** (Rudin, 2019). From the perspective of applied research, the "no-black-box" model represents exactly what decision-maker needs.

## 1.3 Research Objective

The first objective of my research is to identify which out of 4 tested models (ANN, SVM, RF, and M5P) can be called “Interpretable” in the context of practical application in construction engineering and management.

The second objective is not to create a brand new algorithm but to establish a framework to apply existing ensemble algorithms such as Model Tree (M5P) (Quinlan, 1992; Wang and Witten, 1997) in the context of producing interpretable AI for construction industry professionals. The enhancement of M5P with 3-colored scheme is intended to be able to identify which regression is worth using and which should not be used at all. The enhancements made on Model Tree are not aimed to increase the prediction accuracy, but to improve the interpretability of the model’s internal logic.

## 1.4 Thesis Structure

Literature Review follows the Introduction chapter where I discuss four AI algorithms namely artificial neural networks, support vector machine, random forest and model tree. Next follows the Methods chapter where I provide a description of the used data set and performance metrics chosen for the three-color scheme. Also, this chapter contains an explanation of how M5P works in lay terms as well as a short note on using WEKA for building predictive models. The Results chapter illustrates the attribute selection and the performance of each model. The Discussion comes next with the comparison of models’ interpretability and the description of M5P enhancement (three-color scheme). In

Chapter 6 I present a case study to test newly developed three-color scheme. Finally, the conclusion and contributions are summarized in Chapter 7.

# Chapter 2

# Literature Review

## 2.1 Predictive Methods

Frank Rosenblatt (1961) was the first who pioneered the research of recreating the human brain in the form of perceptron—a machine he designed for image recognition[1]. Since then, researchers in a wide range of scientific fields adopted this artificial neural networks (ANN) approach of teaching the machine to recognize the output based on a set of inputs. A significant departure from Rosenblatt's perceptron happened when Vladimir Vapnik and Corinna Cortes (1995) combined Vapnik's optimal hyperplanes developed in 1965 (Vapnik and Kotz, 2006) with ANN design into the concept of Support-Vector Machines (SVM, also support-vector networks). ANN and SVM usually perform marginally better in comparison with simpler models like multiple linear regression or decision tree. In particular, SVM is capable to separate categories in high dimensional space (simply put, SVM can cluster points in unlimited dimensions); ANN excels at distinguishing data that is not linearly separable (in other words, ANN can connect dots with "curly" line that simple linear regression can not achieve). See

[1] Originally, the term "Perceptron" was intended as a generic name for a variety of theoretical nerve nets (Rosenblatt, 1961).

Figure 2.1 for visual illustration. It is challenging for human brains to comprehend a space described with more than three dimensions and nonlinear transfer functions. To a certain degree, attempting to explain how these neural nets reason is analogous to trying to explain the mechanisms of thought process and consciousness in the human brain.

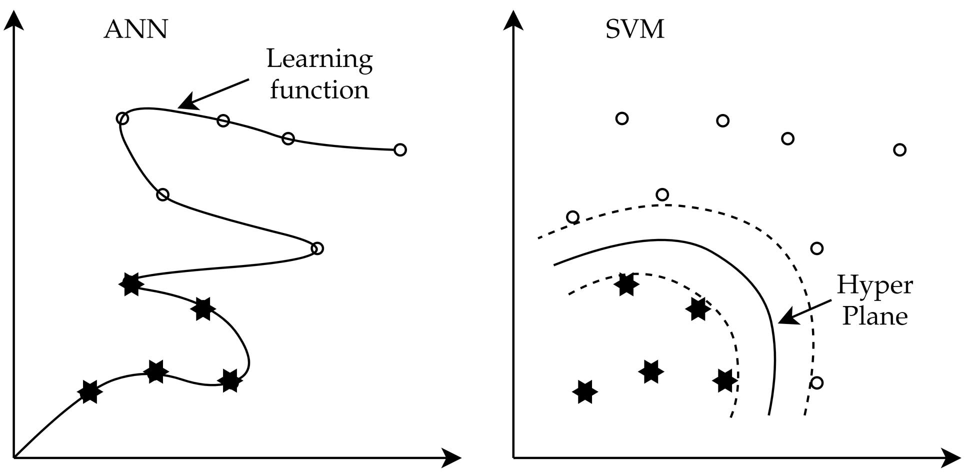


FIGURE 2.1: Abstract illustration of ANN and SVM

Leo Breiman (1984) developed analytical algorithms of the decision tree model for classification and regression (CART). This model acts like an upside-down tree, growing its branches from the root node down to the leaf nodes at the bottom. Each split in a branch represents a numeric or categorical condition. The expansion of the tree ends at "leaf" nodes (Figure 2.2). The interpretability of this model is high, but it has some drawbacks. To quote from Elements of Statistical Learning (Tibshirani and Friedman, 2008), "Trees have one aspect that prevents them from being the ideal tool for predictive learning, namely inaccuracy".

As an enhanced version of the decision tree, Random Forests were developed by Ho Tin Kam (1995). This algorithm builds as many random trees as possible. From Figure 2.3 we can observe how random forest arbitrarily categorizes data points using decision trees and simple yes/no conditions. After all trees are

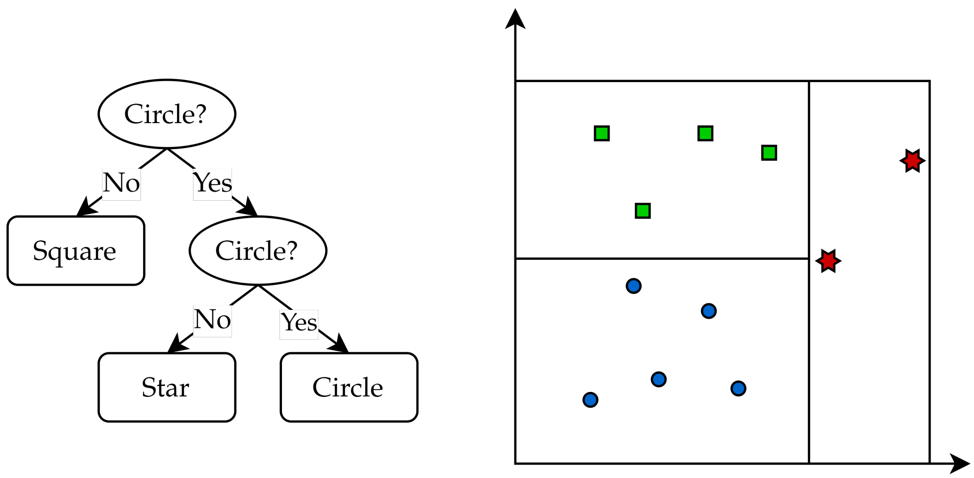


FIGURE 2.2: Abstract illustration of Decision Tree

grown, each of them is evaluated using the data kept for testing[2]. Based on this evaluation, random forest chooses the most accurate tree as the final solution.

Another parallel endeavour to embellish decision tree models resulted in integration with regression algorithms. M5P or Model Tree was first designed by John Quinlan (1992) and then enhanced by Yong Wang and Ian Witten (1997). M5P grows a decision tree-like CART but instead of providing one value at a leaf node it builds a linear regression for the instances which reach that node (Figure 2.4).

It is noteworthy that research in deciphering those "black box" models has achieved limited success in specific application domains. For instance, Lu, AbouRizk and Hermann (2001) created a tornado-like sensitivity graph that can analyze ANN input parameters and measure their impact on the output. Domain experts could use this interpretation tool to validate the model based on their experience and common sense. Stefan Ruping (2006) investigated how to interpret SVM and how to measure the interpretability of the machine learning

[2]Bootstrap aggregating, also called bagging (from **B**ootstrap **AGG**regat**ING**), is used for random forest ensembling (Breiman, 1996).

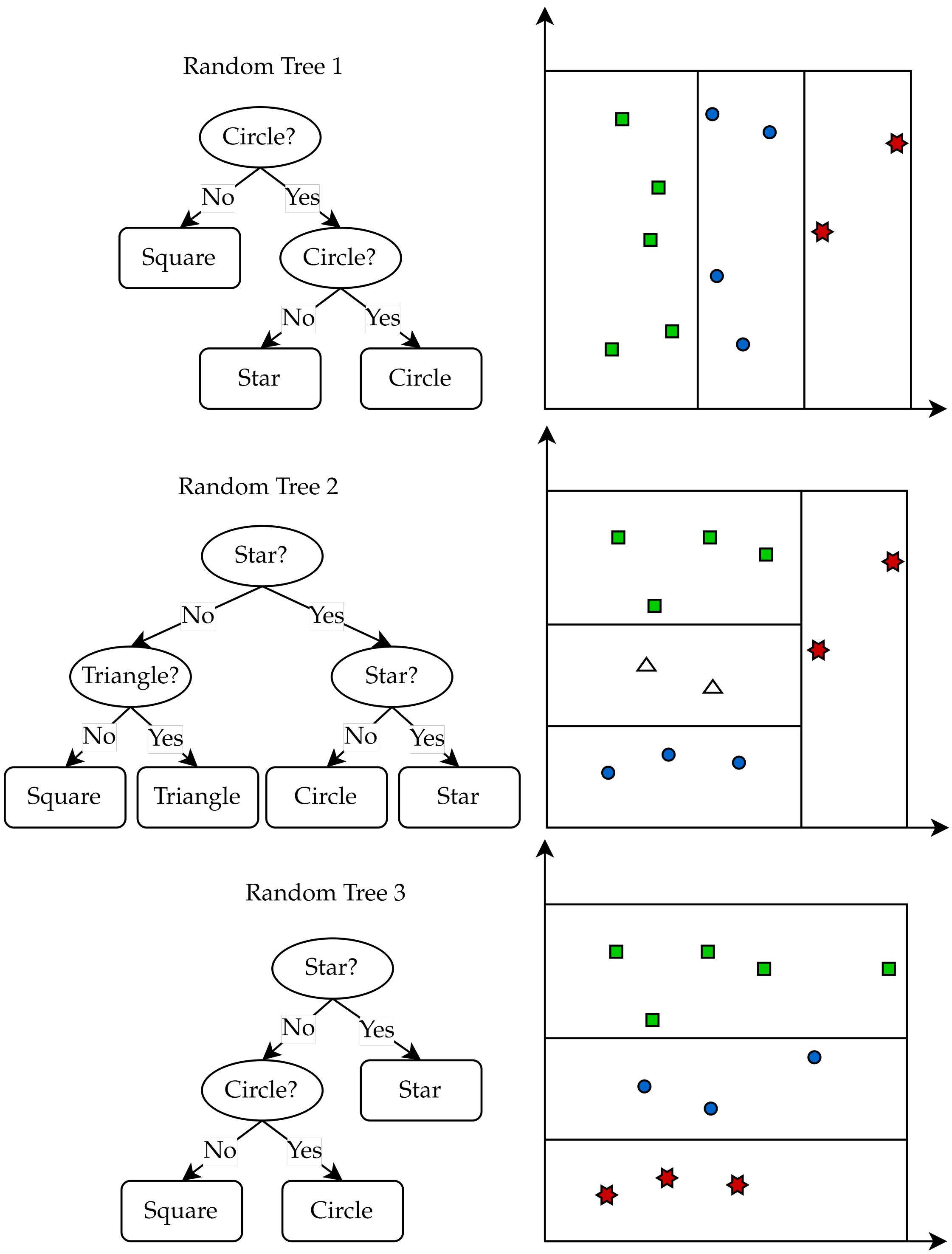


FIGURE 2.3: Abstract illustration of Random Forest

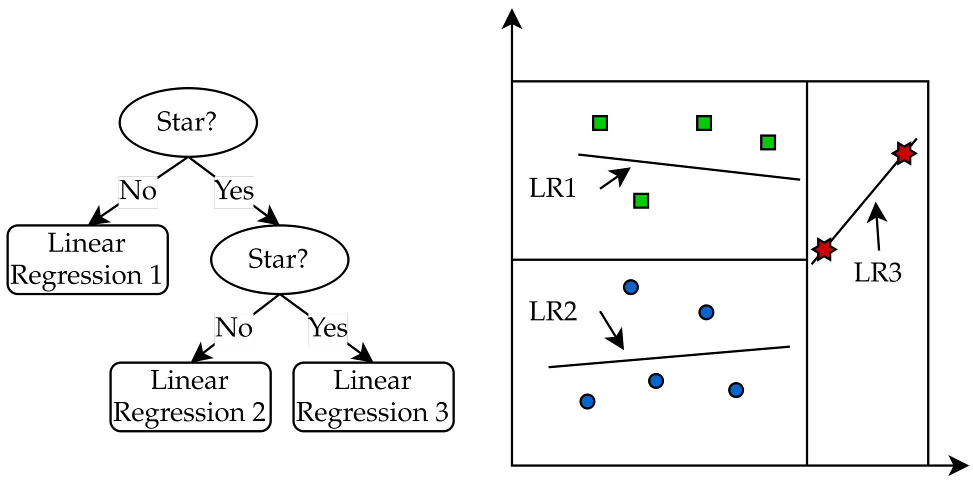


FIGURE 2.4: Abstract illustration of Model Tree (M5P)

algorithm itself. He argued that in order for a model to be comprehensible to the user it must be accurate and efficient so that interpretability does not become a performance bottleneck. In general, M5P holds the potential to provide an interpretable AI model, in contrast with the three "black box" models being tested in the case study of this research (ANN, SVM and RF).

# Chapter 3

# Methods

## 3.1 Data Collection

The groundwork for the research is laid by the historical data that is available to the industry partner and had been garnered throughout the two years of joint industry-academia research efforts.

Mehmed Kantardzic (2011) suggested that all raw data sets initially prepared for data mining are often large and messy. One should not be surprised to find missing values, distortions, misrecordings, inadequate sampling, and so on in these initial data sets. This description exactly characterizes data collection in construction.

Initially, I collected 935 instances each representing a separate pre-bid estimate of a steel project or its revision. Each estimate contained a take-off of the steel profiles[1] (length, weight, and quantity) listed in a project. Additionally, I identified relevant project attributes as follows: location of fabrication (6 different locations), sector (oil & gas, industrial, commercial, infrastructure), scope (supply & erection or supply only) and complexity (light, medium, heavy, and very heavy). The complexity feature is defined based on expert knowledge. For example, light complexity projects are found in the commercial sector where

[1]Steel profile referrers to a type and shape of a cross-section of structural steel.

they use hollow structural steel (so-called "stick build"). A heavy complexity project generally refers to a massive structure made of plates, thus requiring considerable handling and welding operations. A very heavy complexity project is most likely a bridge or an oil rig. The totals for weight, quantity, and length of all the pieces were calculated as a sum of all profiles (for example, Total weight = Hollow structural steel weight + Wide flange weight + C-shape weight + and so on). All the data was extracted from the company's shared folder using Visual Basic code embedded in a master spreadsheet in Excel. Table 3.1 shows the input features.

TABLE 3.1: Steel data set attributes

| Data set input attributes | |
|---|---|
| 1. Scope of work | 20. Round bar weight |
| 2. Sector | 21. Round bar quantity |
| 3. Location | 22. Round bar length |
| 4. Complexity | 23. Miscellaneous weight |
| 5. Hollow structural steel weight | 24. Miscellaneous quantity |
| 6. Hollow structural steel quantity | 25. Miscellaneous length |
| 7. Hollow structural steel length | 26. S-shape weight |
| 8. Wide flange weight | 27. S-shape quantity |
| 9. Wide flange quantity | 28. S-shape length |
| 10. Wide flange length | 29. Wide T-shape weight |
| 11. C-shape weight | 30. Wide T-shape quantity |
| 12. C-shape quantity | 31. Wide T-shape length |
| 13. C-shape length | 32. Pipe weight |
| 14. L-shape weight | 33. Pipe quantity |
| 15. L-shape quantity | 34. Pipe length |
| 16. L-shape length | 35. Total weight of pieces |
| 17. Plate weight | 36. Total quantity of pieces |
| 18. Plate quantity | 37. Total length of pieces |
| 19. Plate length | 38. Total labor-hours (output) |

The common practice in data mining is to clean the data set to perfection getting rid of all the noise and outliers in order to achieve the least possible error. However, a lot of valuable "experience" could be lost in the cleaning process. A decision was made to leave as many instances as practically feasible. 218 instances were left to build a model. Out of 8284 data values, 3505 are absent[2] and 5 are missing. Absent values are expressed as 0 while missing values denoted as "blank". A sample of the dataset can be found in Appendix B.

## 3.2 Performance Metrics

Effective and straightforward metrics are selected based on those commonly applied to evaluate regression models. For the researcher, it is not very important which evaluating metrics to use because in most practical situations the best numeric prediction method is still the best no matter which error measure is used (Witten and Frank, 2011). On the other hand, for practitioners, these metrics need to indicate whether the model is worthwhile or not. Thus, selecting proper metrics for model accuracy evaluation is vital. Next, three general types of errors for evaluating regression or classification algorithms are described, namely: absolute or mean errors, relative errors, and correlation coefficients.

### 3.2.1 Absolute errors

Absolute errors are the most intuitive. For example, Mean Absolute Error is an average of the differences between actual and predicted values. Mean Absolute Percentage Error indicates by how much on average the model under or over predicts the target value. In practical applications, the percentage error as in Equation (3.2) is usually avoided because it tends to be distorted by outliers.

[2] A single project usually does not contain all of the steel profiles.

$$Mean\ Absolute\ Error = \sum_i \frac{|predicted_i - actual_i|}{number_{instances}} \tag{3.1}$$

$$Absolute\ Percentage\ Error = \sum_i \frac{|predicted_i - actual_i|}{|actual_i|} \cdot 100\% \tag{3.2}$$

### 3.2.2 Relative errors

Relative errors can be good metrics to compare AI algorithms. The error is normalized by the error of the simple predictor (the differences between actual values and mean of actuals) that always predicts mean. Furthermore, Relative Squared Error and Root Relative Squared Error will often result in higher numerical values than absolute errors.

$$Relative\ Squared\ Error = \sum_i \frac{(predicted_i - actual_i)^2}{(actual_i - actual_{mean})^2} \tag{3.3}$$

$$Root\ Relative\ Squared\ Error = \sqrt{\sum_i \frac{(predicted_i - actual_i)^2}{(actual_i - actual_{mean})^2}} \tag{3.4}$$

### 3.2.3 R-squared

R-squared is a widely used metric to estimate the accuracy of a model. Ironically, this coefficient is often confusing and can be misused. In statistics, R-squared refers to the Coefficient of Determination and is simply a square of the Pearson Correlation Coefficient (PCC). The PCC measures linear correlation between two variables with a metric ranging between -1 and +1. Mathematicians square the PCC and derive Equation (3.5) to explain the percentage of variation between two variables. Note, this equation is given only to facilitate the interpretation of R-squared and should not be used to calculate the Pearson Correlation Coefficient (Witte and Witte, 2017).

$$R^2 = \frac{Variance_{mean} - Variance_{(actual,predicted)}}{Variance_{mean}} \tag{3.5}$$

In machine learning, R-squared also refers to the Coefficient of Determination that indicates how much variation of the target value is explained by the predicted value, as in Equation (3.6). In other words, if $R^2$ is equal to 0.78 I can say that the model only accounts for 78% of the variation and 22% remains hidden.

$$R^2 = \frac{\sum_i baseline\ error_i^2 - \sum_i error_i^2}{\sum_i baseline\ error_i^2} \tag{3.6}$$

where:

$$error_i = actual_i - predicted_i \tag{3.7}$$

$$baseline\ error_i = actual_i - actual_{mean} \tag{3.8}$$

Equations (3.5) and (3.6) are essentially identical. The other interpretation of the formula (3.6) can be put in the following way: if $R^2$ is equal to 0.78 then the model performs 78% better than a zero rule predictor[3]; or if $R^2$ is equal to -0.11 one can suggest that the model performs 11% worse than a zero rule predictor. Note, this version of R-squared definition can be negative ($R^2 \in (-\infty, 1]$) in case of poor prediction performance (error is much higher than baseline error). This metric can be of great value to the user for evaluating model performance. The name, however, can be changed to Coefficient of Explained Variation to avoid confusion.

[3] A model that always predicts mean.

### 3.2.4 Pearson Correlation Coefficient

As mentioned earlier, the Pearson Correlation Coefficient measures statistical correlation between two variables, denoted with $R \in [-1, 1]$. In other words, this coefficient can tell whether the dependency between two parameters is weak $(R \to 0)$ or strong ($R \to -1$ or $R \to 1$).

$$R = \frac{Covariance_{actual,predicted}}{\sqrt{Variance_{actual}} \cdot \sqrt{Varience_{predicted}}} \tag{3.9}$$

where:

$$Covariance_{actual,predicted} = \frac{\sum_i^n (actual_i - actual_{mean}) \cdot (predicted_i - predicted_{mean})}{n-1} \tag{3.10}$$

Despite PCC having great value for statisticians, in machine learning, it also causes confusion. This metric is scalable meaning that if we multiply all predicted values by any number and leave actual values intact, the correlation stays the same (Figure 3.1 b, c). It implies the possibility that if an algorithm consistently underperforms on all the predictions by a considerable margin, the correlation coefficient can still stay high. An intuitive indicator of ideal prediction accuracy is the correlation line intersecting X and Y axes at the origin with 45° tilt angle (Figure 3.1 a). Thus, it is advisable to apply the correlation coefficient to justify model's prediction performance only if it is supported by graphical visualization of the tilt angle of the correlation line.

For the current study, I select (a) correlation coefficient $R$, (b) coefficient of explained variation $R^2$ and (c) mean absolute percentage error as AI model performance evaluating metrics. Although the last error measure emphasizes the existence of outliers, it is the most intuitive and the easiest to interpret. Dealing

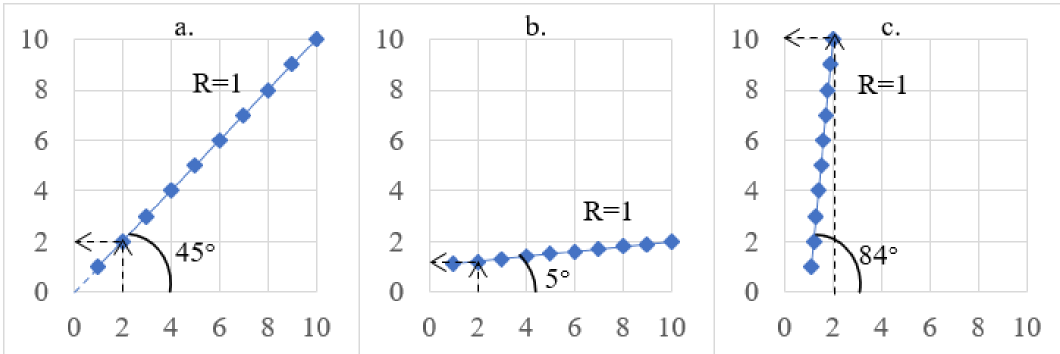


FIGURE 3.1: Pearson Correlation Coefficient

with the outliers in practical applications is a crucial task as these outliers can be minority representatives of the population that do not land in the sampled dataset or indicate certain errors inherent in the data.

## 3.3 How M5P Works

Ensemble top-down trees are usually grown to the maximum size and then pruned backwards replacing poor-performing subtrees with leaves (see Figure 5.5). Then the smoothing procedure adjusts the performance of each leaf node to compensate for sharp discontinuities that would inevitably occur between adjacent linear models (Wang and Witten, 1997). These internal mechanisms are employed to achieve the highest feasible prediction accuracy for M5P model as a whole.

### 3.3.1 Growing the initial tree

To build the upside-down tree, M5P uses a *Splitting Criteria*, as in Equation (3.11), to find the attribute and the value at which to begin growing branches.

$$Splitting\ Criteria = sd(Output) - sd(Output_{split})_{weighted} \tag{3.11}$$

where:

$$\begin{aligned} sd(Output_{split})_{weighted} = {} & sd(Output_{split\_1}) \cdot \frac{\sum |Output_{split\_1}|}{\sum |Output|} + \\ & + sd(Output_{split\_2}) \cdot \frac{\sum |Output_{split\_2}|}{\sum |Output|} \end{aligned} \tag{3.12}$$

The algorithm evaluates all possible splits and measures the magnitude by which the standard deviation (*sd*) of the output is reduced. The reduction is represented as the sum of the weighted standard deviations of the output values of evaluated splits. For example, if we have two attributes (one input and one output) and twelve instances, M5P would sort values for each attribute and find an average between adjacent points (potential splitting values). Then, Equation (3.11) is calculated for each of the possible splits (in our example—eleven splits) and a splitting value with the smallest *Splitting Criteria* is chosen.

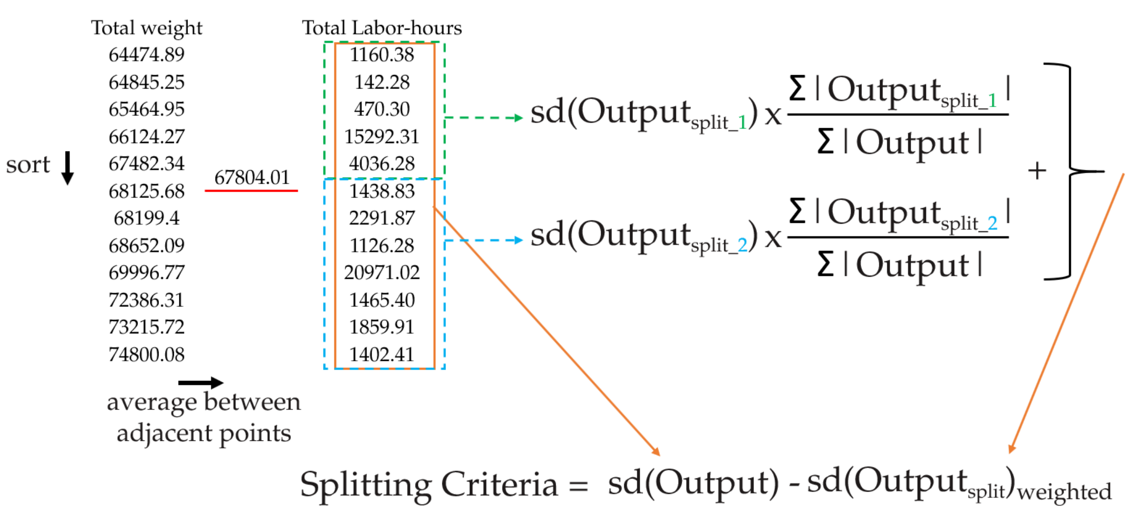


FIGURE 3.2: Splitting Criteria calculation example

This procedure continues until the tree is grown to the maximum size and stopping condition is met. In the case of M5P, the tree stops growing when the Leaf node has less than three instances or the standard deviation of Leaf's output is less than 5% of the standard deviation of the output of the entire set (3.13).

$$sd(Output_{leaf}) < 0.05 \cdot sd(Output) \tag{3.13}$$

### 3.3.2 Pruning

After the tree is grown, M5P builds multiple linear regressions for each leaf as well as each subtree using standard regression and greedy search attribute selection. Then the algorithm tests each instance (training process) and averages the difference between predicted and actual values (expected error) for each leaf and subtree. The error of every entity is then multiplied by *Compensation Factor*, as in Equation (3.14), to account for the fact that the model is not tested on unseen cases. The lower the number of instances—the more error increment is expected.

$$Compensation\ Factor = \frac{number_{instances} + number_{attributes}}{number_{instances} - number_{attributes}} \tag{3.14}$$

The pruning itself is a process of comparing the expected error of the lower leaves with the expected error of the upper subtree. If regression in the subtree performs better than the regressions in the leaves, then they are pruned and subtree becomes a leaf (bottom-up pruning).

### 3.3.3 Smoothing

Finally, smoothing is employed to calibrate the *Predicted Value* of the leaf by passing it to higher subtrees and eventually to the root node. Equation (3.15) is calculated at each level of the tree (from leaf to subtree, from subtree to next level subtree...to the root node). The goal is to combine the prediction power of the leaf with the prediction power of subtrees.

$$Predicted\ Value_{upper\ node} = \frac{Predicted\ Value_{node} \cdot number_{inst.} + Predicted\ Value_{lower\ node} \cdot k}{number_{instances} + k} \tag{3.15}$$

In the Equation (3.15) $k$ is a constant and in M5P it is equal to 15 whereas $number_{instances}$ refers to the subtree which is denoted as "node" (blue in Figure 3.3).

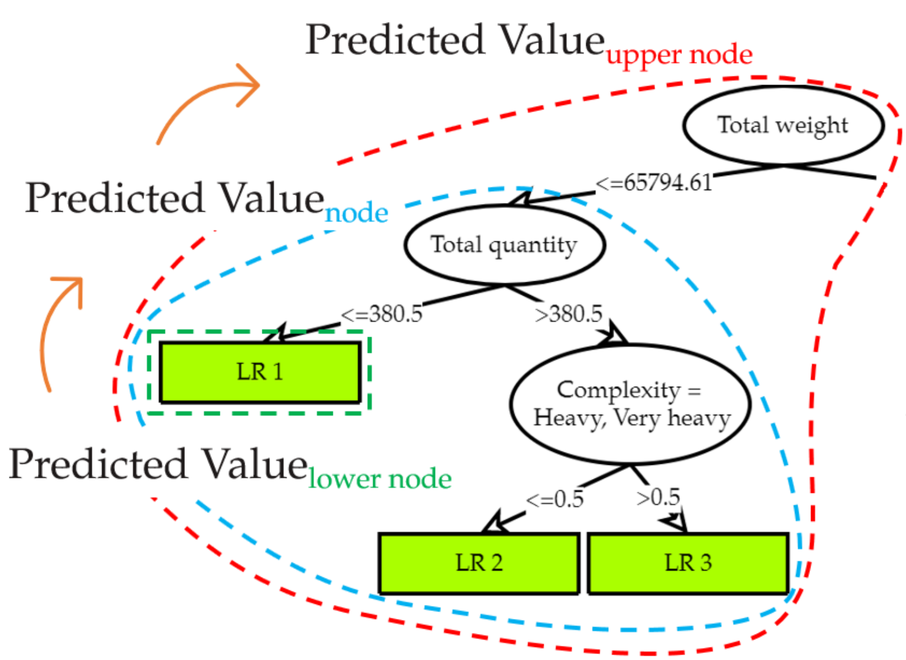


FIGURE 3.3: Smoothing of Predicted Value

## 3.4 Using WEKA for Building Predictive Models

In order to devise an interpretable AI model, I used WEKA to develop independent M5P, ANN, RF, and SVM models from the same steel fabrication dataset. WEKA provides implementations of various learning algorithms. You can preprocess a dataset, feed it into a learning scheme, and analyze the resulting classifier and its performance—without writing any program code at all (Witten and Frank, 2016). This application was chosen because it is an open-source and can be accessed through a graphical user interface. It is widely used for teaching, research, and industrial applications, contains a vast number of built-in tools for standard machine learning tasks, and additionally, it is written in Java and distributed under General Public License which means that it is free to use on operating systems such as Windows, Linux, and Macintosh by anyone.

As supporting material, Appendix C provides a manual of how to setup a Model Tree (M5P) in WEKA and Appendix D contains the configurations of all the discussed AI models.

# Chapter 4

# Results

## 4.1 Attribute Selection

As it was mentioned earlier, Model Tree includes in its algorithm Greedy Search that selects most valuable attributes for each regression. In the current study, to make the comparison fair, I used a Bi-directional Greedy Wrapper Subset Evaluator described by Ron Kohavi and George John (1997) to perform attribute selection for the rest of the counterpart models (Table 4.1). The resulting optimal feature subsets are tailored to a particular regression algorithm by training the model with different subsets of attributes and choosing the subset with the highest accuracy.

To better understand the impact of selected attributes I calculate their total proportional correlation using Equation (4.1) where $R_i$ is the correlation coefficient between output and each attribute.

$$R_{proportional} = \frac{\sum R_i}{number_{attributes}} \tag{4.1}$$

As we can observe from Table 4.1, some attributes, namely Total weight and Complexity (double underlined), and Plate quantity, Plate weight, and Hollow steel quantity (underlined), were selected by all the models, or by three out of

TABLE 4.1: Selected attributes and their correlation to the output (w–weight, q–quantity, l-length)

| # | ANN | | SVM | | RF | | M5P | |
|---|---|---|---|---|---|---|---|---|
| 1 | Total w | 74% | Total w | 74% | Total w | 74% | Total w | 74% |
| 2 | Total q | 60% | Plate q | 58% | R-nd bar q | 8% | Total q | 60% |
| 3 | Plate q | 58% | Wide fl. q | 56% | R-nd bar w | 8% | Plate q | 58% |
| 4 | Plate w | 48% | Plate w | 48% | S-shape q | 8% | Wide fl. q | 56% |
| 5 | Hollow st. q | 41% | Hollow st. q | 41% | Complexity | 7% | Wide fl. w | 54% |
| 6 | T-shape q | 29% | Plate l | 32% | | | Plate w | 48% |
| 7 | C-shape q | 26% | Wide fl. l | 25% | | | Hollow st. q | 41% |
| 8 | Location | 12% | L-shape l | 25% | | | Hollow st. w | 41% |
| 9 | Sector | 12% | Sector | 12% | | | L-shape w | 36% |
| 10 | Complexity | 7% | R-nd bar w | 8% | | | C-shape w | 27% |
| 11 | S-shape w | -3% | S-shape q | 8% | | | C-shape q | 26% |
| 12 | S-shape l | -5% | Complexity | 7% | | | Sector | 12% |
| 13 | | | | | | | Complexity | 7% |
| $R_{proportional}$(4.1) | | 30% | | 33% | | 21% | | 41% |

four models respectively. It may indicate that these attributes play a crucial part in the labour-hours prediction.

## 4.2 Performance of Each Model

The performance of each model is contrasted in Table 4.2 in terms of their prediction metrics. The correlation graphs based on the testing data are provided for each model in Section 5.1. I used 10-fold cross-validation to estimate the accuracy of the learning schemes. In 10-fold cross-validation the dataset is randomly split into 10 mutually exclusive subsets (the folds) of approximately equal size. The algorithm is trained on 90% of the dataset and tested on 10% of the dataset ten times. The cross-validation estimate of accuracy is the average of ten tests (Kohavi, 1995).

TABLE 4.2: 10-fold cross-validation

| Metric | ANN | SVM | RF | M5P |
|---|---|---|---|---|
| Absolute Percentage Error (want low) | 150% | 62% | 50% | 69% |
| Coefficient of Explained Variation, $R^2$ (want high) | 89% | 56% | 87% | 74% |
| Correlation Coefficient, $R$ (want high) | 94% | 87% | 95% | 88% |
| Correlation line tilt angle (want 45°) | 45.9° | 33.5° | 39.9° | 39.1° |

Next, let us examine the models that were obtained by running each algorithm on the same steel dataset.

# Chapter 5

# Discussion

## 5.1 Comparison of the Models' Interpretability

### 5.1.1 ANN

With $R^2$ equal to 89%, this algorithm is able to explain the most variation in the model. On the other hand, the absolute percentage error is the highest which could mean that ANN had "overlearned[1]" the dataset by memorizing noise instead of generalizing patterns in data. From Figure 5.1 we can see that the correlation line is the closest to the ideal 45°. The subset of chosen attributes is a reasonable representation of the problem, having a total proportional correlation of 30%. The highest accuracy was achieved with the following ANN parameters: number of hidden layers = 5, transfer function—sigmoid, learning rate = 0.2, momentum = 0.1. Having 5 layers and 12 attributes, which became 24 after the transition from nominal (Location, Sector and Complexity) to binary, we have $5 \cdot 24 = 120$ coefficients (weights) plus 6 bias weights between -1 and 1. In addition, the initialization of those 126 neuron weights is randomly set. See model setup in the Appendix D.1

[1] Over-learning refers to an event when a machine learning algorithm fits its function to each data point too precisely. For visualization see Figure 2.1.

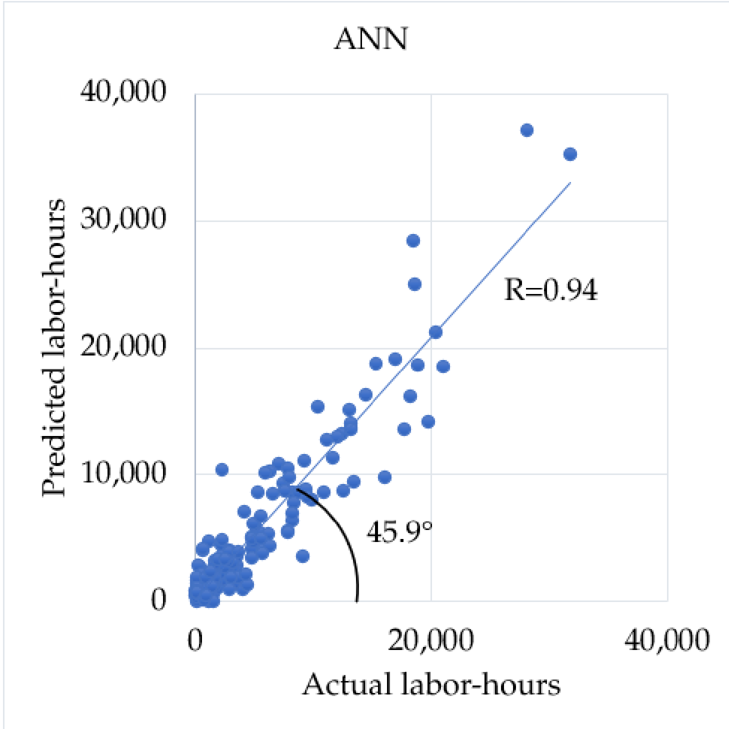


FIGURE 5.1: ANN correlation scatter-plot

### 5.1.2 SVM

SVM performed the worst out of 4 tested models with 56% explained variation and 33.5° correlation line tilt angle. From Figure 5.2 we can observe that those seven instances located under the correlation line drag it away from the ideal tilt. The settings for the SVM model are: transfer function (kernel)—radial basis function (RBF), SVM type—$\nu$-SVR (Chang and Lin, 2019), $\nu$[2] = 0.5. Unfortunately, the interpretability of the resulting model is extremely challenging due to the fact that the radial kernel function mathematically transforms data points into infinite dimensions to find their high-dimensional relationship on each other and identify a relative distance between these observations. See model setup in the Appendix D.2

[2] $\nu \in (0, 1]$—a parameter that controls the number of support vectors.

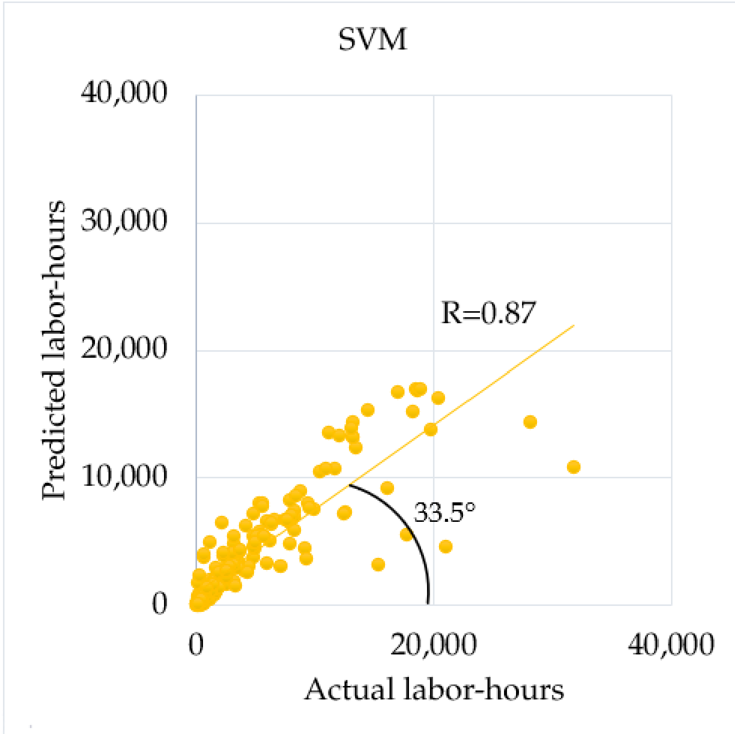


FIGURE 5.2: SVM correlation scatter-plot

### 5.1.3 RF

I was able to achieve the highest accuracy at 100 iterations with the average size of each tree equal to 169 (number of leaves). The RF model performed the best out of 4 tested machine learning models in terms of the lowest absolute percentage error of 50% and the highest correlation coefficient. It is worth mentioning that the tilt angle of the correlation line is 39.9° that indicates an "under predicting" trend (Figure 5.3). The attribute selection in the final model missed some critical factors (e.g. steel profiles like wide flange, plate and hollow steel) and hence is deemed insufficient. In my view, an expert would not rely on a model that does not conform to existing know-how or common sense. However, if interpretability was not an issue, Random Forest would be the algorithm of my choice. See model setup in the Appendix D.3

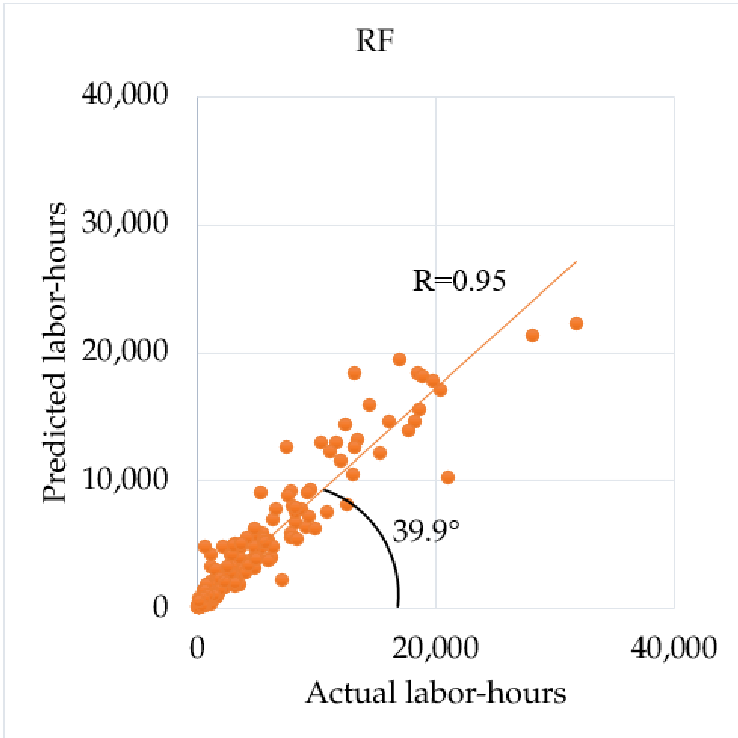


FIGURE 5.3: RF correlation scatter-plot

### 5.1.4 M5P

M5P did not stand out as the winner by any of the tested performance metrics. Given the four models being evaluated, it only outperformed SVM with correlation coefficient equal to 88% (Figure 5.4).

But from the interpretability standpoint, this model has marked advantages. First, the selected attributes (Table 4.1) align the best with experts' know-how and common sense in the application domain. Note the attribute selection is embedded in the algorithm and does not require additional computational workload. The resulting model is totally transparent to the user, the reasoning logic of the model can be intuitively validated, and the model can be used manually (Figure 5.5). Numerical variable-based splitting conditions in M5P model are

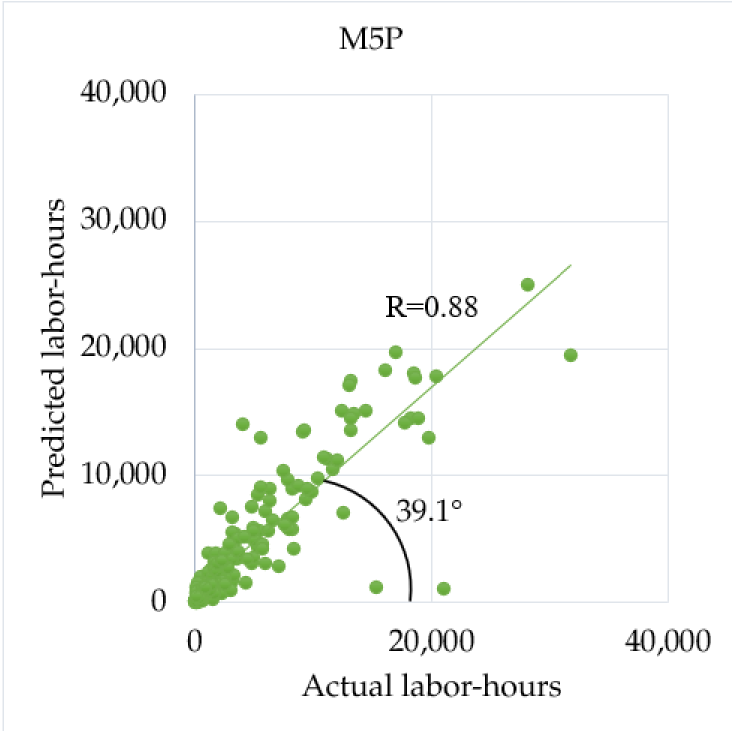


FIGURE 5.4: M5P correlation scatter-plot

straight-forward and self-explanatory. Nominal variable-based splitting conditions (such as complexity) can be interpreted as follows: if the project has complexity as heavy (all three splits contain heavy complexity), then its binary value equals 1 and at the split, it follows "> 0.5" condition. If the complexity is medium (none of the three nominal splits contain medium complexity), then its binary value equals 0 and at the split it follows "$\leq 0.5$" condition. In the regression formula, same logic applies: if sector is oil & gas or infrastructure, then multiplier equals 1, otherwise multiplier equals 0 (Figure 5.5, second line of Linear Regression 3). As such, if I had a new project with the total weight equal to 6000 kg, total quantity equal to 500 pieces, and complexity equal to heavy, I would use regression No. 3 to predict the total amount of labor-hours. See model setup in the Appendix D.4

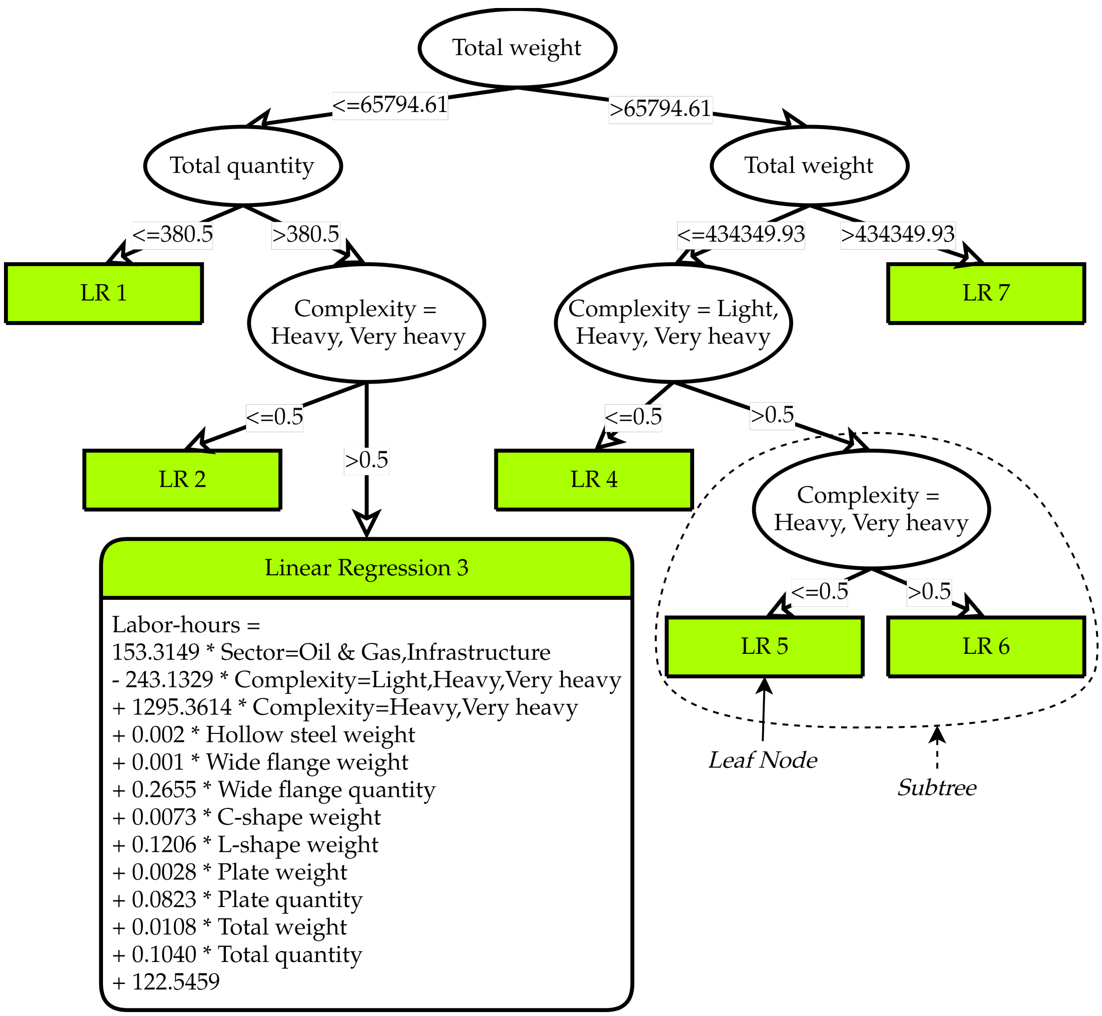


FIGURE 5.5: M5P tree (weight in kilograms)

The above discussion is provided to justify my choice of M5P as an interpretable AI model for practical application. The prediction accuracy only served to make a statement that Model Tree is as accurate as others. In the same time, the interpretability of the model's internal logic is only possible on Model Tree. Next, I provide an enhancement to M5P that increases the interpretability of the model as a whole and of the performance of each linear regression.

## 5.2 3-Colored Scheme for M5P

Incorporating pruning and smoothing features to boost accuracy leads to reduction of the interpretability of the regressions and the model as a whole. Another obstacle to interpretability is the way we verify and validate models. The performance results provided in Table 4.2 are an average of 10 different models which are not the same as the model in Figure 5.5. This implies that I have no idea how good or bad each linear regression performs because they are not tested in cross-validation. I only know the training performance for each leaf (Table 5.1).

TABLE 5.1: Leaf training performance

| Metric | Full | LR1 | LR2 | LR3 | LR4 | LR5 | LR6 | LR7 |
|---|---|---|---|---|---|---|---|---|
| Absolute Percentage Error (want low) | 54% | 112% | 99% | 72% | 42% | 48% | 33% | 26% |
| Coefficient of Explained Variation, $R^2$ (want high) | 93% | -76% | -396% | 39% | 10% | 59% | 43% | 86% |
| Correlation Coefficient, R (want high) | 97% | 19% | 23% | 70% | 58% | 83% | 79% | 93% |
| Correlation line tilt angle (want 45°) | 46.0° | 9.8° | 6.2° | 37.1° | 32.2° | 36.3° | 65.3° | 47.1° |
| Number of instances | 218 | 49 | 35 | 18 | 47 | 22 | 8 | 39 |

Given the fact that the overall training and testing accuracy shows acceptable results, when we look closely into each linear regression, the conclusions may differ. From visual inspection (Figure 5.6) and performance analysis (Table 5.1), only one leaf (Linear Regression 7) aligns well with common sense.

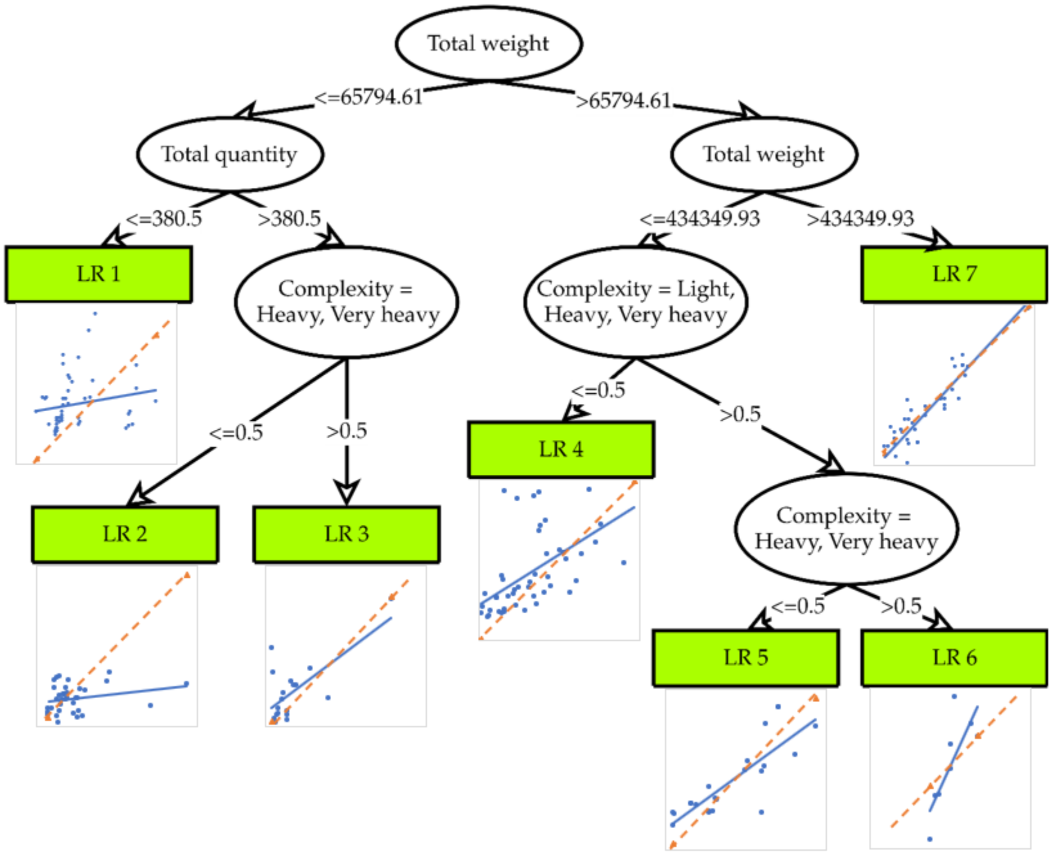


FIGURE 5.6: M5P tree with correlation graphs for each node (dash line – full model)

Knowing that the accuracy of all seven leaves had been altered by smoothing, I decided to test each node separately using leave-one-out cross-validation where the number of folds is equal to the number of data points. By doing so I ensure that every linear model is treated on the same ground, regardless of the number of available instances. In this case, every regression performs independently without impact from the higher level subtrees. In Figure 5.7 I depict the performance of each leaf applying three-colored schemes. The interpretation is

| Metric | LR1 | LR2 | LR3 | LR4 | LR5 | LR6 | LR7 | Ideal |
|---|---|---|---|---|---|---|---|---|
| Absolute Percentage Error (want low) | 57% | 47% | 74% | 26% | 69% | 45% | 23% | 0% |
| Coefficient of Explained Variation, $R^2$ (want high) | 52% | -11% | 57% | 76% | -360% | 6% | 71% | 100% |
| Correlation Coefficient, R (want high) | 75% | 54% | 77% | 88% | 57% | 48% | 86% | 100% |
| Correlation line tilt angle (want 45°) | 38° | 40° | 25° | 42° | 13° | 28° | 41° | 45° |
| Number of instances (and folds) | 49 | 35 | 18 | 47 | 22 | 8 | 39 | |

FIGURE 5.7: Leaf leave-one-out cross-validation performance

as follows: for $R$ and $R^2$ red marking represents their values of 0% (negative values are as red as 0% to make $R$ comparable to $R^2$), yellow marking represents the value of $R$ and $R^2$ equal to 50% and green marking represents their values of 100%. For absolute percentage error, the condition is the opposite: green is 0%, yellow is 50%, and red 100%. Tilt angle becomes green at 45°, red at 0° and 90°, and yellow at mid-point of the range between 0° to 45° or 45° to 90°(22.5° and 67.5°, respectively). The coloring scheme merges in between the above mentioned thresholds namely, green to yellow and yellow to red. I recognize that this framework is not exact. I also know that there is no such thing as "silver bullet" in machine learning that can determine rigid boundaries for the minimum acceptable error. There is always uncertainty and risk in using predictive models. This concept provides the user with a visual aid to decide which regression is valid to use (i.e. *Green Leaf*), which regression should be used with caution (i.e. *Yellow Leaf*) and finally which regression should not be used at all (i.e. *Red Leaf*). It is assumed that the Green Leaf must have at least three metrics satisfying "green" condition and one satisfying "yellow" condition, Yellow Leaf can satisfy either "yellow" or "green" conditions and if the leaf has at least one red flag on the four metrics, then it is deemed as Red Leaf.

From Figure 5.7 I can suggest that leaves with ID No. 4 and 7 are Green and leaf No. 1 is Yellow. The other ones are Red and should not be used for prediction. Figure 5.8 depicts the revised model with a corresponding leaf color scheme applied.

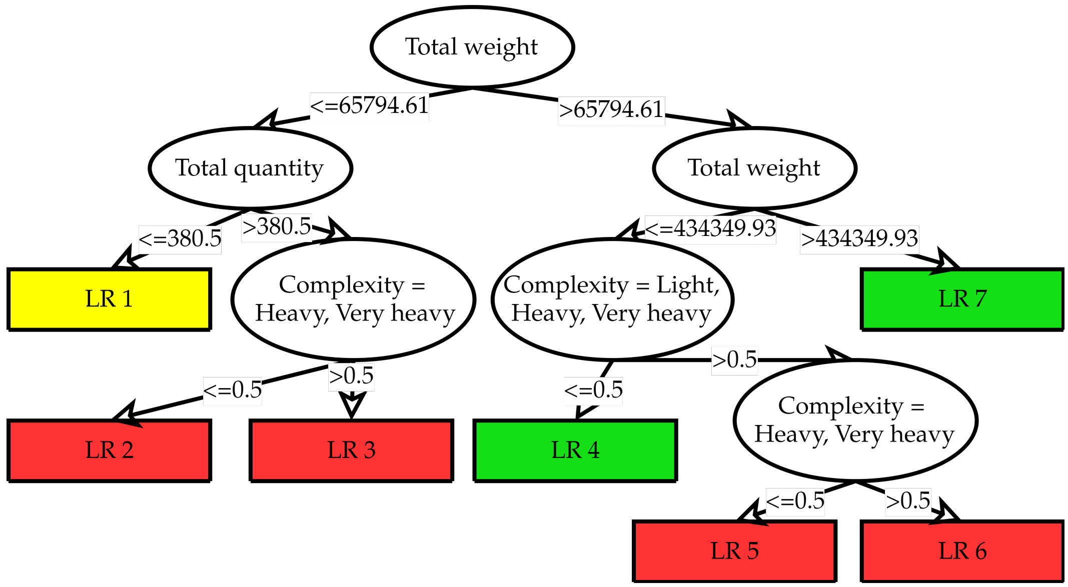


FIGURE 5.8: Revised M5P tree

The Enhanced Model Tree is preferred not because it predicts more accurate, that is not the point! It is because its logic is transparent to the modeler and user. Considering a small dataset associated with sub-models at each leaf node, the enhanced Model Tree application framework assesses the quality of regression at each leaf node with a selection of regression performance metrics and N-fold testing regimen. A color scheme denoting the quality of regression is intuitive and effective to guide the practical application.

The used data set and derived models contain confidential information of our partner company and hence are not presented in its entirety. A sample of the steel data set can be found in the Appendix B. Therefore, in the next chapter, I provide a test case for further validating and presenting the current framework application in detail. It is noted that the application problem is not estimating steel fabrication labor-hours, but still falls in the construction engineering domain.

# Chapter 6

# Practical Test Case

To test the newly developed framework I chose publicly available and well studied Concrete Compressive Strength data set which can be found through University of California Irvine Machine Learning Repository (Yeh, 1998a). It may be confusing that the problem domain switched from steel fabrication to high-performance concrete. Taking into account that a good quality data set with a numeric target (regression prediction type) is difficult to find in the construction field for new algorithm performance bench-marking, I consider this change acceptable for the current study. The data set was first described by I-Cheng Yeh (1998b). He gathered experimental data of High Performance Concrete (HPC) mix proportions and corresponding compressive strength from 17 different sources to build an ANN model. The components of the dataset are described in Table 6.1.

In his experiments, Yeh achieved the coefficient of explained variance ($R^2$) equal to 91.4% (4-fold cross-validation) which is considered acceptable for practical application.

Using the described concrete dataset I built the M5P model (Figure 6.1) utilizing the default smoothing and pruning features in WEKA. The performance of the tree is shown in Table 6.2.

TABLE 6.1: HPC dataset attributes

| Attribute | Minimum | Maximum | Average |
|---|---|---|---|
| | Inputs | | |
| Cement ($kg/m^3$) | 102 | 540 | 281.2 |
| Blast Furnace Slag ($kg/m^3$) | 0 | 359.4 | 73.9 |
| Fly Ash ($kg/m^3$) | 0 | 200.1 | 54.2 |
| Water ($kg/m^3$) | 121.8 | 247 | 181.6 |
| Superplasticizer ($kg/m^3$) | 0 | 32.2 | 6.2 |
| Coarse Aggregate ($kg/m^3$) | 801 | 1145 | 972.9 |
| Fine Aggregate ($kg/m^3$) | 594 | 992.6 | 773.6 |
| Age ($days$) | 1 | 365 | 45.7 |
| | Output | | |
| Compressive strength ($MPa$) | 2.33 | 82.6 | 35.8 |

TABLE 6.2: Concrete example training and cross-validation performance

| Metric for M5P | Training | 10-fold | 4-fold | 4-fold(Yeh, 1998b) |
|---|---|---|---|---|
| Absolute Percentage Error | 13% | 15% | 15% | - |
| Coefficient of Explained Variation, $R^2$ | 89% | 86% | 86% | 91% |
| Correlation Coefficient, $R$ | 94% | 92% | 92% | - |
| Correlation line tilt angle°) | 45.6° | 45.4° | 45.4° | - |

Just as with the steel fabrication model it is not understood what is the testing accuracy of each leaf in the resulting tree. As in Section 5.2 I tested each node independently using leave-one-out cross validation and depicted the outcome in Figure 6.2. To comment on the figure, leaves with ID No. 7, 8 and 9 had poor performance measures and were deemed as "Red". All other nodes showed acceptable results and were colored "Green" for practical use. It is worth mentioning that if I remove "Red" leaves resulting from performance and only consider "Green" ones, the accuracy improves significantly (Table 6.3).

TABLE 6.3: Concrete example only "Green" leaves performance

| Metric for M5P | Leave-one-out |
|---|---|
| Absolute Percentage Error (want low) | 14% |
| Coefficient of Explained Variation, $R^2$ (want high) | 98% |
| Correlation Coefficient, $R$ (want high) | 94% |
| Correlation line tilt angle (want 45°) | 44.6° |

While Yeh's ANN model showed good prediction results, I doubt that laboratory technicians or assistants would be comfortable training and using neural networks on a day-to-day basis. They need something as simple as formula sheets to quickly get the results on the spot, even without using a computer. These formula sheets can be found in Appendix A where I provide linear regressions for each node in the revised tree (Figure 6.3).

There are two universal patterns about compressive concrete strength: the lower the ratio between water and binder the higher the strength, and as concrete gets older in the curing stage-the strength gets higher. From the tree in Figure 6.1 we can observe that eight out of nine splitting attributes that identify the distribution of the data across the tree are either concrete age, or cement portion, or water portion. In the study conducted by Kadri et al.(2012) they tested a compressive strength of high-performance concrete having different ratios of water to cement with silica fume additive. Their findings are depicted in Figure 6.4. I plotted the value of the first splitting attribute ($Age = 21\,days$) on their graphs to visualize how the model separates two different stages of curing. The first stage on the left has a rapid increase in strength whereas the second stage on the right has a lower increasing trend. In this manner, these two simple examples support the idea that M5P can build a highly interpretable model that aligns with common knowledge in a specific problem domain.

Linear regressions provided by M5P could serve as a practical use case for

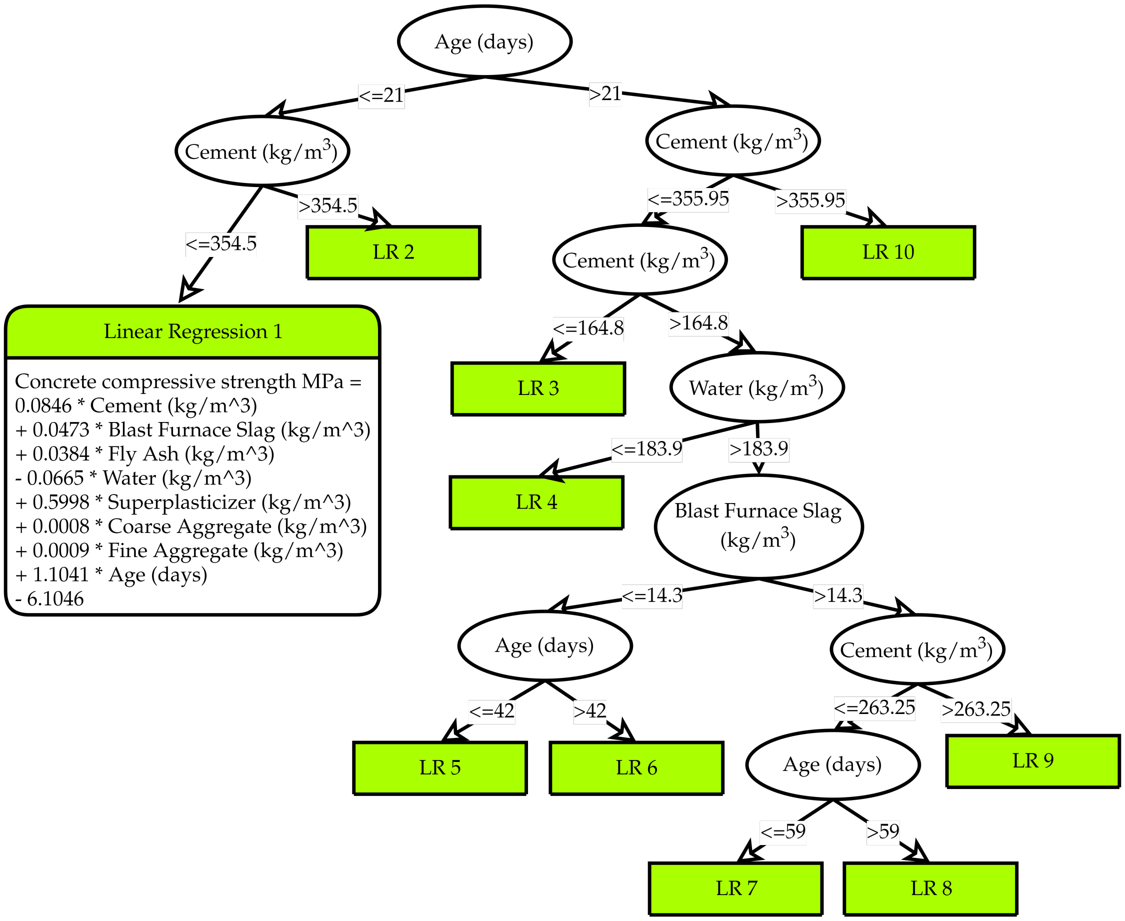


FIGURE 6.1: Concrete example model

| Metric | LR 1 | LR 2 | LR 3 | LR 4 | LR 5 | LR 6 | LR 7 | LR 8 | LR 9 | LR 10 | Ideal |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Absolute Percentage Error (want low) | 18% | 17% | 14% | 13% | 9% | 7% | 15% | 18% | 14% | 11% | 0% |
| Coefficient of Explained Variation, $R^2$ (want high) | 79% | 70% | 66% | 65% | 72% | 77% | -7% | -80% | 3% | 66% | 100% |
| Correlation Coefficient, R (want high) | 89% | 84% | 82% | 81% | 85% | 88% | 35% | -14% | 33% | 81% | 100% |
| Correlation line tilt angle (want 45°) | 44.5° | 43.7° | 40.9° | 43.9° | 43.9° | 43.2° | 23.9° | -10.3° | 28.6° | 44.1° | 45° |
| Number of instances (and folds) | 230 | 94 | 126 | 193 | 64 | 60 | 37 | 20 | 47 | 159 | |

FIGURE 6.2: Concrete example leave-one-out cross-validation performance

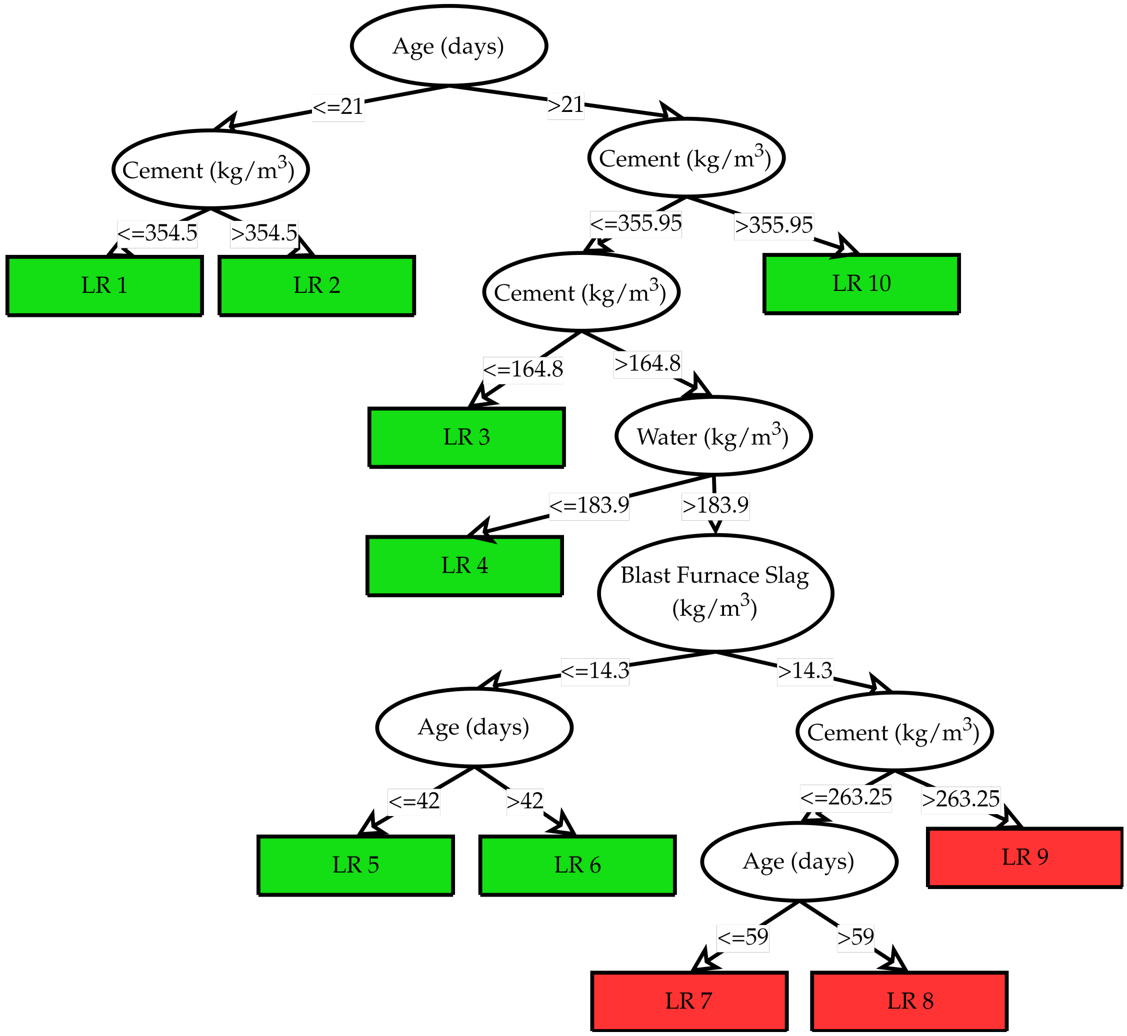


FIGURE 6.3: Revised concrete example model

building design. Let us assume that I wanted to configure my concrete mix to a certain strength at a certain point of ageing. While I could use graphs of experimental results that are usually provided for fixed mixes of concrete, using the equations in Appendix A, I could adjust individual components whilst keeping the other ones constant. That would allow me to create a bespoken concrete mix tailored to specific circumstances.

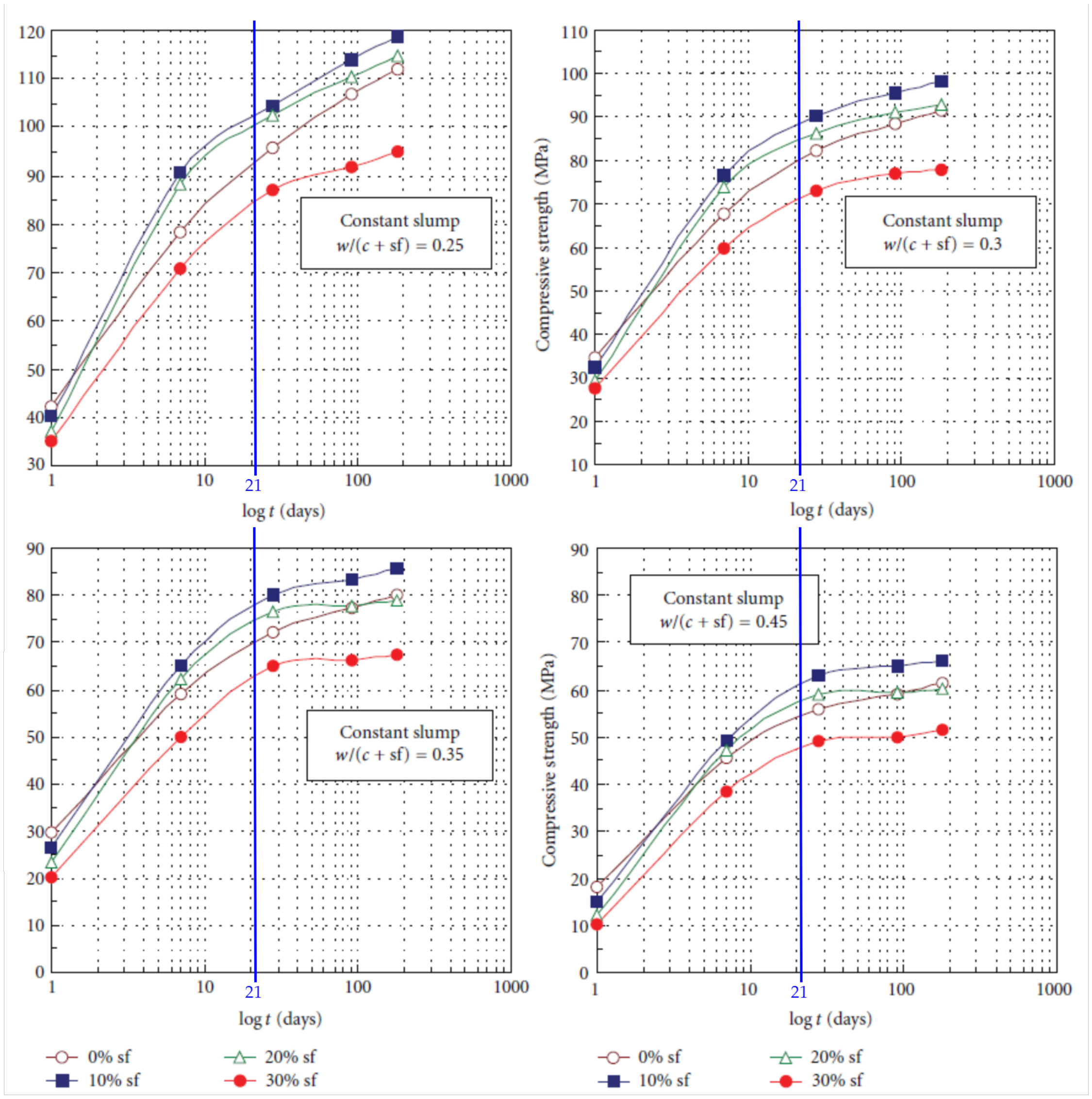


FIGURE 6.4: Strength development of concretes at different water-cementitious materials ratios, w-water, c-cement, sf-silica fume (Kadri et al., 2012)

# Chapter 7

# Conclusion

## 7.1 Findings

In collaboration with a steel fabrication company in western Canada, my research investigated the cost estimation department in regards to preparing prebid estimates. I found that most of the estimators fall under the "baby boomer" cohort. It is imperative to capture their experience know-how before they retire and pass it on to the next generation of engineers and managers. Another finding showed that the professionals in this company were not eager to use AI techniques. From my point of view, they needed something that could be easily interpreted and trusted.

A dataset sourced from the real world was used to compare various AI algorithms and search for candidate for interpretable AI. I identified that Model Tree (M5P) is the candidate. The above hypothesis is proven through comparing Model Tree against ANN, SVM, and RF using steel fabrication project estimating case. Considering this, I revealed hidden patterns and decision rules encoded in the M5P model. Furthermore, I distinguished valid sub-models from invalid ones by using a three-colored scheme on the regressions at each leaf node of the tree. The enhanced Model Tree application framework was successfully implemented on the Concrete Compressive Strength data set. It was incorporated due to the confidentiality issue with the steel fabrication data set which restricted me from providing all resulting regressions and their coefficients. The resulting model was elaborated inside out explaining every internal parameter, branching variable, threshold, and the sub-model at each leaf node including coefficients for each regression.

## 7.2 Industry Contribution

I presented a real-world case study of implementing Model Tree (M5P), along with three commonly applied AI algorithms such as artificial neural networks (ANN), support vector-machine (SVM), and random forest (RF) to predict project labor-hours based on pre-bid estimate data. Through reviewing performance measures for regression algorithms I selected (a) correlation coefficient and its tilt angle, (b) coefficient of explained variation and (c) mean absolute percentage error as performance metrics for evaluating model accuracy. I not only calibrated the Model Tree to the lowest feasible error but also created a model representing estimators' know-how, making it accessible to the company's newcomers and professionals in training.

For further research validation and application demonstration, I used the developed color scheme to build a model that learns from an established machine learning dataset for algorithm performance bench-marking and predicts compressive strength of high performance concrete as a test case. The resulting tree and linear regressions can be used manually in laboratories even without having a computer.

Both of the implemented case studies are aimed to promote artificial intelligence in the construction industry.

## 7.3 Academic Contribution

To a certain extent, excellence is a consequence of continuous learning from predecessors. Despite that, we reached the point where conventional learning is simply not enough to get to the next level. The construction industry constantly suffers from losing precious knowledge when experts leave or retire. There is an obvious disconnect in expertise transfer. Many studies have been done regarding incorporating machine learning techniques in the construction field and in fact, it is becoming more acceptable to practitioners. However, there is a huge barrier in the way of artificial intelligence entering the industry's day-to-day practice. Humans tend to avoid things they do not understand.

The resulting AI model is the simplest form yet still sufficient to represent the complexity in the practical problems and tolerate the limitations in available data (limited quantity, noise, and incompleteness) in construction engineering.

The enhancement in the form of the three-colored model performance scheme to M5P algorithm has no analogous concepts and functions in any existing software, hence it is considered an academic contribution of this research.

To remark on the limitations, my research relies on a well-established data mining tool such as WEKA. While it is widely used open-source educational software, some data processing tools and hypotheses remain hidden in complex code. I assume that based on the reputation of the University of Waikato, where this application was developed and continually maintained, and a track record of related academic publications and successful use cases over the last two decades, this tool is reliable enough to support my research.

## 7.4 Next Steps

From the interpretability standpoint, further steps can be made towards revealing data set attributes' meaning and contribution to AI models. For now, attribute selection is made by applying existing regression modeling tools with the objective mostly set as to increase model's prediction performance.

Throughout application of M5P on construction engineering problems I noticed its capability to cluster instances in the available data set that contribute very little to the prediction results. In other words, Model Tree has yet to identify data points which may have poor quality or are outliers. Further research could be conducted to enhance this capability, for example, in comparison with other clustering techniques like K-means or Hierarchical clustering.

Another branch of follow up research can be done on further improving and applying the proposed three-colored model performance scheme on machine learning algorithms of classification type where performance metrics slightly differ from regression ones.

# Bibliography

Adeli, Hojjat (2001). “Neural Networks in Civil Engineering: 1989–2000”. In: *Computer-Aided Civil and Infrastructure Engineering* 16.2, pp. 126–142. ISSN: 1093-9687, 1467-8667. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/0885-9507.00219.

Breiman, Leo (1996). “Bagging predictors”. In: *Machine Learning* 24.2, pp. 123–140. ISSN: 0885-6125, 1573-0565. URL: http://link.springer.com/10.1007/BF00058655.

Breiman, Leo et al. (1984). *Classification And Regression Trees*. Boca Raton, FL: Chapman & Hall/CRC. ISBN: 978-0-412-04841-8.

Calabrese, Darren (2015). “Boom, bust and economic headaches”. In: *The Globe and Mail*. URL: https://www.theglobeandmail.com/globe-investor/retirement/the-boomer-shift-how-canadas-economy-is-headed-for-majorchange/article27159892/.

Canada, Statistics Canada Government of (2017). *Population estimates on July 1st, by age and sex*. Last Modified: 2019-09-30 Library Catalog: www150.statcan.gc.ca. URL: https://www150.statcan.gc.ca/t1/tbl1/en/tv.action?pid=1710000501.

Chang, Chih-Chung and Chih-Jen Lin (2019). “LIBSVM: A library for support vector machines”. In: *ACM Trans. Intell. Syst. Technol.* 2.3, p. 39. ISSN: 2157-6904, 2157-6912. URL: https://dl.acm.org/doi/10.1145/1961189.1961199.

Emmert-Streib, Frank, Olli Yli-Harja, and Matthias Dehmer (2020). "Explainable Artificial Intelligence and Machine Learning: A reality rooted perspective". In: *eprint arXiv:2001.09464*, p. 13.

Gunning, David (2016). *Explainable Artificial Intelligence (XAI)*. Defense Advanced Research Projects Agency (DARPA). URL: https://www.darpa.mil/attachments/DARPA-BAA-16-53.pdf.

Ho, Kam (1995). "Random Decision Forests". In: Proceedings of the 3rd International Conference on Document Analysis and Recognition. Montreal, QC, pp. 278–282.

Kadri, E. H. et al. (2012). "The Compressive Strength of High-Performance Concrete and Ultrahigh-Performance". In: *Advances in Materials Science and Engineering* 2012. ISSN: 1687-8434, 1687-8442. URL: http://www.hindawi.com/journals/amse/2012/361857/.

Kantardzic, Mehmed (2011). *DATA MINING Concepts, Models, Methods, and Algorithms*. Second edition. Hoboken, New Jersey: John Wiley & Sons, Inc. ISBN: 978-0-470-89045-5.

Kohavi, R. and G. H. John (1997). "Wrappers for feature selection". In: *Artificial Intelligence* 97.1, pp. 273–324.

Kohavi, Ron (1995). "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection". In: International Joint Conference on Artificial Intelligence. Montreal, QB, p. 7.

Kulkarni, Preeti, S. Londhe, and M. Doe (2017). "Artificial Neural Networks for Construction Management: A Review". In: *Journal of Soft Computing in Civil Engineering*, p. 19. ISSN: 2588-2872.

Liddy, William and John Cross (2002). "Conceptual Estimating, and the Steel Fabrication.pdf". In: *Modern Steel Construction*.

Lu, Ming, S. M. AbouRizk, and U. H. Hermann (2001). "Sensitivity Analysis of Neural Networks in Spool Fabrication Productivity Studies". In: *Journal of Computing in Civil Engineering* 15.4, pp. 299–308. ISSN: 0887-3801, 1943-5487. DOI: 10.1061/(ASCE)0887-3801(2001)15:4(299).

Quinlan, J R (1992). "Learning with Continuous Classes". In: 5th Australian Joint Conference on Artificial Intelligence, p. 6.

Rosenblatt, Frank (1961). *Principles of Neurodynamics. Perceptrons and the Theory of Brain Mechanisms*. Buffalo, NY: Cornell Aeronautical Lab Inc. 626 pp.

Rudin, Cynthia (2019). "Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead". In: *arXiv:1811.10154 [cs, stat]*, p. 20. URL: http://arxiv.org/abs/1811.10154.

Ruping, Stefan (2006). "Learning Interpretable Models". PhD thesis. Dortmund: der Universitat Dortmund am Fachbereich Informatik von.

Tibshirani, Sami and Harry Friedman (2008). *The Elements of Statistical Learning*. Second edition. Stanford, California: Springer.

Vapnik, V. and S. Kotz (2006). *Estimation of Dependences Based on Empirical Data: Empirical Inference Science*. New York, NY: Springer. ISBN: 978-0-387-34239-9.

Vapnik, Vladimir and Corinna Cortes (1995). "Support-vector networks". In: *Machine Learning* 20.3, pp. 273–297. ISSN: 0885-6125, 1573-0565. URL: http://link.springer.com/10.1007/BF00994018.

Wang, Yong and Ian H Witten (1997). "Inducing Model Trees for Continuous Classes". In: *9th European Conference on Machine Learning Poster Papers*, p. 10.

Witte, Robert S. and John S. Witte (2017). *Statistics*. Eleventh edition. Hoboken, NJ: Wiley. 480 pp. ISBN: 978-1-119-25451-5.

Witten, Ian and Eibe Frank (2011). *Data Mining: Practical Machine Learning Tools and Techniques*. Second edition. San Francisco: Elsevier. 525 pp. ISBN: 978-0-12-374856-0.

Witten, Ian and Eibe Frank (2016). *The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques"*. URL: https://www.cs.waikato.ac.nz/ml/weka/Witten_et_al_2016_appendix.pdf.

Yeh, I.-C. (1998a). *Concrete Compressive Strength Dataset*. URL: https://archive.ics.uci.edu/ml/datasets/Concrete+Compressive+Strength.

— (1998b). "Modeling of strength of high-performance concrete using artificial neural networks". In: *Cement and Concrete Research* 28.12, pp. 1797–1808. ISSN: 00088846. URL: https://linkinghub.elsevier.com/retrieve/pii/S0008884698001653.

# Appendix A

# Regressions for Concrete test case

## A.1 Linear regression 1

$Age(days) <= 21$

$|\ \ Cement(kg/m^3) <= 354.5$ :

$$\begin{aligned}
Compressive\,strength\ (MPa) =\ & 0.075 * Cement(kg/m^3) + \\
& 0.0354 * Blast\ Furnace\ Slag(kg/m^3) + \\
& 0.0269 * Fly\ Ash(kg/m^3) + \\
& -0.0865 * Water(kg/m^3) + \\
& 0.6381 * Superplasticizer(kg/m^3) + \\
& -0.0112 * Coarse\ Aggregate(kg/m^3) + \\
& -0.0094 * Fine\ Aggregate(kg/m^3) + \\
& 1.144 * Age(days) + \\
& 20.764
\end{aligned}$$

## A.2 Linear regression 2

$Age(days) <= 21$

$| \; Cement(kg/m^3) > 354.5$ :

$$
\begin{aligned}
Compressive\ strength\ (MPa) = & 0.096 * Cement(kg/m^3) + \\
& 0.0898 * Blast\ Furnace\ Slag(kg/m^3) + \\
& 0.0134 * Fly\ Ash(kg/m^3) + \\
& -0.3097 * Water(kg/m^3) + \\
& -0.1961 * Superplasticizer(kg/m^3) + \\
& -0.0362 * Coarse\ Aggregate(kg/m^3) + \\
& -0.0348 * Fine\ Aggregate(kg/m^3) + \\
& 2.2142 * Age(days) + \\
& 92.8024
\end{aligned}
$$

## A.3 Linear regression 3

$Age(days) > 21$

$|\ Cement(kg/m^3) <= 355.95$

$|\quad |\ Cement(kg/m^3) <= 164.8$ :

$$
\begin{aligned}
Compressive\, strength\ (MPa) = & -0.0148 * Cement(kg/m^3) + \\
& \ 0.0983 * Blast\ Furnace\ Slag(kg/m^3) + \\
& \ 0.018 * Fly\ Ash(kg/m^3) + \\
& -0.1414 * Water(kg/m^3) + \\
& \ 0.0749 * Superplasticizer(kg/m^3) + \\
& -0.0227 * Coarse\ Aggregate(kg/m^3) + \\
& -0.012 * Fine\ Aggregate(kg/m^3) + \\
& \ 0.0793 * Age(days) + \\
& \ 66.538
\end{aligned}
$$

## A.4 Linear regression 4

$Age(days) > 21$

$|\ Cement(kg/m^3) <= 355.95$

$|\quad |\ Cement(kg/m^3) > 164.8$

$|\quad |\quad |\ Water(kg/m^3) <= 183.9$ :

$$
\begin{aligned}
Compressive\,strength\ (MPa) =& 0.2097 * Cement(kg/m^3) + \\
& 0.2104 * Blast\ Furnace\ Slag(kg/m^3) + \\
& 0.1838 * Fly\ Ash(kg/m^3) + \\
& 0.0627 * Water(kg/m^3) + \\
& 0.2028 * Superplasticizer(kg/m^3) + \\
& 0.1051 * Coarse\ Aggregate(kg/m^3) + \\
& 0.1063 * Fine\ Aggregate(kg/m^3) + \\
& 0.1621 * Age(days) + \\
& -245.7724
\end{aligned}
$$

## A.5 Linear regression 5

$Age(days) > 21$

$| \ Cement(kg/m^3) <= 355.95$

$| \quad | \ Cement(kg/m^3) > 164.8$

$| \quad | \quad | \ Water(kg/m^3) > 183.9$

$| \quad | \quad | \quad | \ Blast\ Furnace\ Slag(kg/m^3) <= 14.3$

$| \quad | \quad | \quad | \quad | \ Age(days) <= 42 :$

$$
\begin{aligned}
Compressive\ strength\ (MPa) =& 0.1303 * Cement(kg/m^3) + \\
& -0.3452 * Blast\ Furnace\ Slag(kg/m^3) + \\
& \ 0.1132 * Fly\ Ash(kg/m^3) + \\
& -0.2014 * Water(kg/m^3) + \\
& -0.4195 * Superplasticizer(kg/m^3) + \\
& -0.0057 * Coarse\ Aggregate(kg/m^3) + \\
& \ 0.0092 * Fine\ Aggregate(kg/m^3) + \\
& \ 23.6472
\end{aligned}
$$

## A.6 Linear regression 6

$Age(days) > 21$

$|\ Cement(kg/m^3) <= 355.95$

$|\ \ |\ Cement(kg/m^3) > 164.8$

$|\ \ |\ \ |\ Water(kg/m^3) > 183.9$

$|\ \ |\ \ |\ \ |\ Blast\ Furnace\ Slag(kg/m^3) <= 14.3$

$|\ \ |\ \ |\ \ |\ \ |\ Age(days) > 42 :$

$$\begin{aligned} Compressive\ strength\ (MPa) = & 0.1475 * Cement(kg/m^3) + \\ & 0.2673 * Blast\ Furnace\ Slag(kg/m^3) + \\ & 0.0642 * Fly\ Ash(kg/m^3) + \\ & -0.5462 * Water(kg/m^3) + \\ & 1.3101 * Superplasticizer(kg/m^3) + \\ & 0.0104 * Coarse\ Aggregate(kg/m^3) + \\ & 0.0215 * Fine\ Aggregate(kg/m^3) + \\ & 0.0157 * Age(days) + \\ & 63.8728 \end{aligned}$$

## A.7 Linear regression 7

$Age(days) > 21$

$|\ Cement(kg/m^3) <= 355.95$

$|\ \ |\ Cement(kg/m^3) > 164.8$

$|\ \ |\ \ |\ Water(kg/m^3) > 183.9$

$|\ \ |\ \ |\ \ |\ Blast\ Furnace\ Slag(kg/m^3) > 14.3$

$|\ \ |\ \ |\ \ |\ \ |\ Cement(kg/m^3) <= 263.25$

$|\ \ |\ \ |\ \ |\ \ |\ \ |\ Age(days) <= 59:$

$$\begin{aligned}
Compressive\ strength\ (MPa) = & 0.0448 * Cement(kg/m^3)+ \\
& 0.0312 * Blast\ Furnace\ Slag(kg/m^3)+ \\
& -0.0868 * Fly\ Ash(kg/m^3)+ \\
& -0.0371 * Water(kg/m^3)+ \\
& 1.2251 * Superplasticizer(kg/m^3)+ \\
& -0.0076 * Coarse\ Aggregate(kg/m^3)+ \\
& -0.0385 * Fine\ Aggregate(kg/m^3)+ \\
& 59.2209
\end{aligned}$$

## A.8 Linear regression 8

$Age(days) > 21$

$|\ Cement(kg/m^3) <= 355.95$

$|\quad |\ Cement(kg/m^3) > 164.8$

$|\quad |\quad |\ Water(kg/m^3) > 183.9$

$|\quad |\quad |\quad |\ Blast\ Furnace\ Slag(kg/m^3) > 14.3$

$|\quad |\quad |\quad |\quad |\ Cement(kg/m^3) <= 263.25$

$|\quad |\quad |\quad |\quad |\quad |\ Age(days) > 59:$

$$
\begin{aligned}
Compressive\ strength\ (MPa) = &-0.0536 * Cement(kg/m^3)+\\
&\ 0.0591 * Blast\ Furnace\ Slag(kg/m^3)+\\
&-0.0489 * Water(kg/m^3)+\\
&\ 0.0832 * Coarse\ \ Aggregate(kg/m^3)+\\
&\ 0.022 * Age(days)+\\
&-30.0589
\end{aligned}
$$

## A.9 Linear regression 9

$Age(days) > 21$

$| \ Cement(kg/m^3) <= 355.95$

$| \quad | \ Cement(kg/m^3) > 164.8$

$| \quad | \quad | \ Water(kg/m^3) > 183.9$

$| \quad | \quad | \quad | \ Blast\ Furnace\ Slag(kg/m^3) > 14.3$

$| \quad | \quad | \quad | \quad | \ Cement(kg/m^3) > 263.25 :$

$$
\begin{aligned}
Compressive\ strength\ (MPa) =& 0.0737 * Cement(kg/m^3) + \\
& -0.0213 * Blast\ Furnace\ Slag(kg/m^3) + \\
& \ 0.007 * Fly\ Ash(kg/m^3) + \\
& -0.1053 * Superplasticizer(kg/m^3) + \\
& -0.0267 * Coarse\ Aggregate(kg/m^3) + \\
& \ 0.0533 * Fine\ Aggregate(kg/m^3) + \\
& \ 0.0265 * Age(days) + \\
& \ 12.1755
\end{aligned}
$$

## A.10 Linear regression 10

$Age(days) > 21$

$| \ Cement(kg/m^3) > 355.95$ :

$$
\begin{aligned}
Compressive\ strength\ (MPa) = & 0.1071 * Cement(kg/m^3) + \\
& 0.1607 * Blast\ Furnace\ Slag(kg/m^3) + \\
& 0.0958 * Fly\ Ash(kg/m^3) + \\
& -0.3161 * Water(kg/m^3) + \\
& -0.5749 * Superplasticizer(kg/m^3) + \\
& 0.0023 * Coarse\ Aggregate(kg/m^3) + \\
& 0.0095 * Fine\ Aggregate(kg/m^3) + \\
& 0.0446 * Age(days) + \\
& 48.0601
\end{aligned}
$$

# Appendix B

# Steel fabrication dataset sample

This appendix contains the steel fabrication dataset sample and its characteristics. The are seven instances provided in Tables B.2 and B.3, one for each Leaf.

TABLE B.1: Steel dataset characteristics

| | |
|---|---|
| No. of attributes | 38 |
| No. of instances initial | 935 |
| No. of missing values initial | 22435 |
| No. of instances after cleansing | 218 |
| No. of missing values after cleansing | 5 |
| No. of zeros(footnote 2, page 15) after cleansing | 3505 |

TABLE B.2: Steel dataset sample(weight in kilograms, length in meters

| Regression No. | LR 1 | LR 2 | LR 3 | LR 4 |
|---|---|---|---|---|
| Instance No. | 149 | 65 | 47 | 27 |
| Scope | Supply & erect. | Supply | Supply & erect. | Supply |
| Sector | Industrial | Industrial | Infrastructure | Industrial |
| Location | Edmonton | Saskatchewan | Edmonton | Winnipeg |
| Complexity | Medium | Medium | Very heavy | Medium |
| Hollow Steel weight | 0 | 0 | 0 | 0 |
| Hollow Steel quantity | 0 | 0 | 0 | 0 |
| Hollow Steel length | 0 | 0 | 0 | 0 |
| Wide flange weight | 6538 | 0 | 4661 | 1346 |
| Wide flange quantity | 42 | 0 | 27 | 9 |
| Wide flange length | 678 | 0 | 58 | 146 |
| C-shape weight | 258 | 0 | 0 | 95 |
| C-shape quantity | 9 | 0 | 0 | 1 |
| C-shape length | 14 | 3 0 | 0 | 66 |
| L-shape weight | 2038 | 0 | 2624 | 270 |
| L-shape quantity | 44 | 0 | 40 | 5 |
| L-shape length | 258 | 0 | 97 | 94 |
| Plate weight | 1496 | 15192 | 6289 | 84114 |
| Plate quantity | 177 | 1776 | 410 | 239 |
| Plate length | 245 | 84 | 103 | 1600 |
| Round bar weight | 0 | 0 | 0 | 0 |
| Round bar quantity | 0 | 0 | 0 | 0 |
| Round bar length | 0 | 0 | 0 | 0 |
| Miscellaneous weight | 0 | 0 | 0 | 0 |
| Miscellaneous quantity | 0 | 0 | 0 | 0 |
| Miscellaneous length | 0 | 0 | 0 | 0 |
| S-shape weight | 0 | 0 | 0 | 0 |
| S-shape quantity | 0 | 0 | 0 | 0 |
| S-shape length | 0 | 0 | 0 | 0 |
| T-shape weight | 0 | 0 | 0 | 43 |
| T-shape quantity | 0 | 0 | 0 | 2 |
| T-shape length | 0 | 0 | 0 | 14 |
| Pipe weight | 0 | 19693 | 0 | 0 |
| Pipe quantity | 0 | 3108 | 0 | 0 |
| Pipe length | 0 | 118 | 0 | 0 |
| Total weight | 10330 | 34885 | 13575 | 85869 |
| Total quantity | 272 | 4884 | 477 | 256 |
| Total length | 1325 | 201 | 258 | 1920 |
| Labor-hours | 323 | 1332 | 3022 | 2490 |
| Hours, predicted | 213 | 1175 | 1904 | 2074 |

TABLE B.3: Steel dataset sample continuation

| Regression No. | LR 5 | LR 6 | LR 7 |
|---|---|---|---|
| Instance No. | 79 | 132 | 118 |
| Scope | Supply & erect. | Supply | Supply |
| Sector | Commercial | Industrial | Infrastructure |
| Location | Saskatchewan | Saskatchewan | Vancouver |
| Complexity | Light | Heavy | Medium |
| Hollow Steel weight | 65709 | 103477 | 477 |
| Hollow Steel quantity | 302 | 490 | 66 |
| Hollow Steel length | 4066 | 7325 | 90 |
| Wide flange weight | 267183 | 11768 | 81463 |
| Wide flange quantity | 610 | 77 | 102 |
| Wide flange length | 16921 | 396 | 98 |
| C-shape weight | 14953 | 7550 | 6422 |
| C-shape quantity | 116 | 46 | 160 |
| C-shape length | 726 | 194 | 210 |
| L-shape weight | 34911 | 2907 | 47 |
| L-shape quantity | 1577 | 134 | 16 |
| L-shape length | 4612 | 74 | 4 |
| Plate weight | 33120 | 47044 | 575571 |
| Plate quantity | 2703 | 577 | 2508 |
| Plate length | 2470 | 2193 | 2552 |
| Round bar weight | 969 | 0 | 0 |
| Round bar quantity | 2129 | 0 | 0 |
| Round bar length | 322 | 0 | 0 |
| Miscellaneous weight | 0 | 0 | 0 |
| Miscellaneous quantity | 0 | 0 | 0 |
| Miscellaneous length | 0 | 0 | 0 |
| S-shape weight | 0 | 0 | 1299 |
| S-shape quantity | 0 | 0 | 57 |
| S-shape length | 0 | 0 | 155 |
| T-shape weight | 136 | 14243 | 10106 |
| T-shape quantity | 1 | 18 | 80 |
| T-shape length | 66 | 1788 | 143 |
| Pipe weight | 3558 | 57 | 0 |
| Pipe quantity | 127 | 12 | 0 |
| Pipe length | 556 | 16 | 0 |
| Total weight | 420539 | 187046 | 675385 |
| Total quantity | 7565 | 1354 | 2989 |
| Total length | 29740 | 11986 | 3252 |
| Labor-hours | 5678 | 9269 | 13372 |
| Hours, predicted | 4452 | 11312 | 13244 |

# Appendix C

# WEKA Concrete example setup manual

This appendix is provided for the teaching purposes. It explains how to setup M5P model for the Concrete example test case using WEKA software.

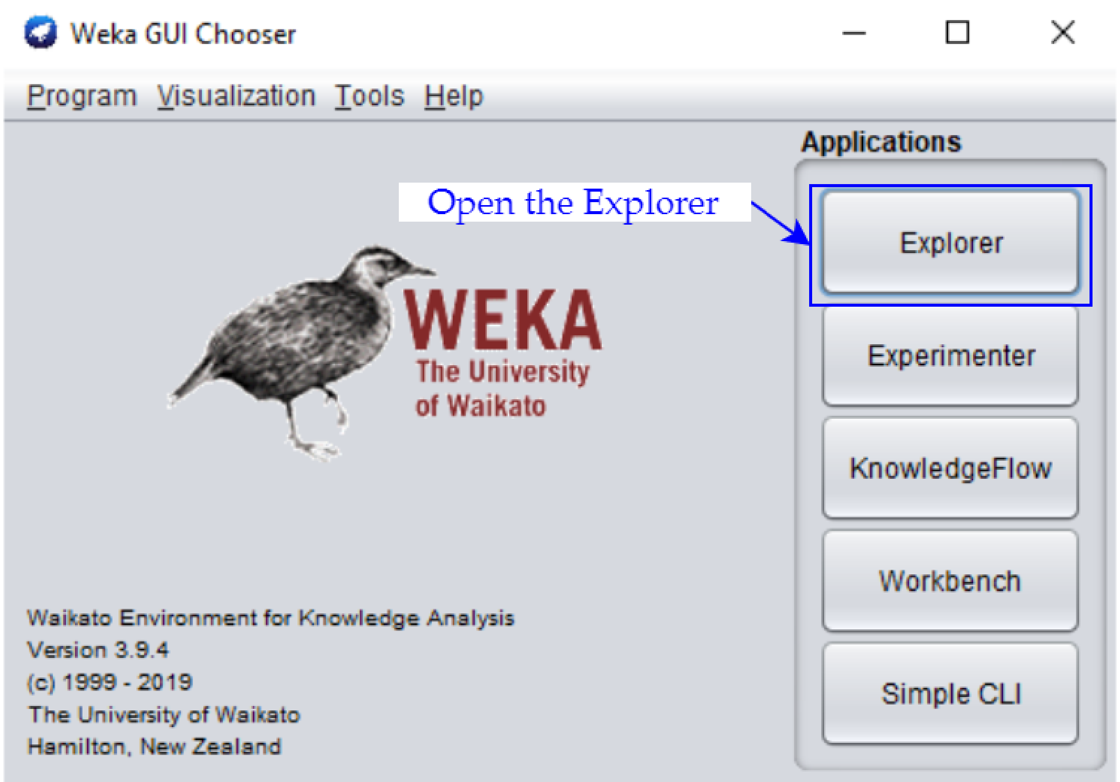


FIGURE C.1: Applications window

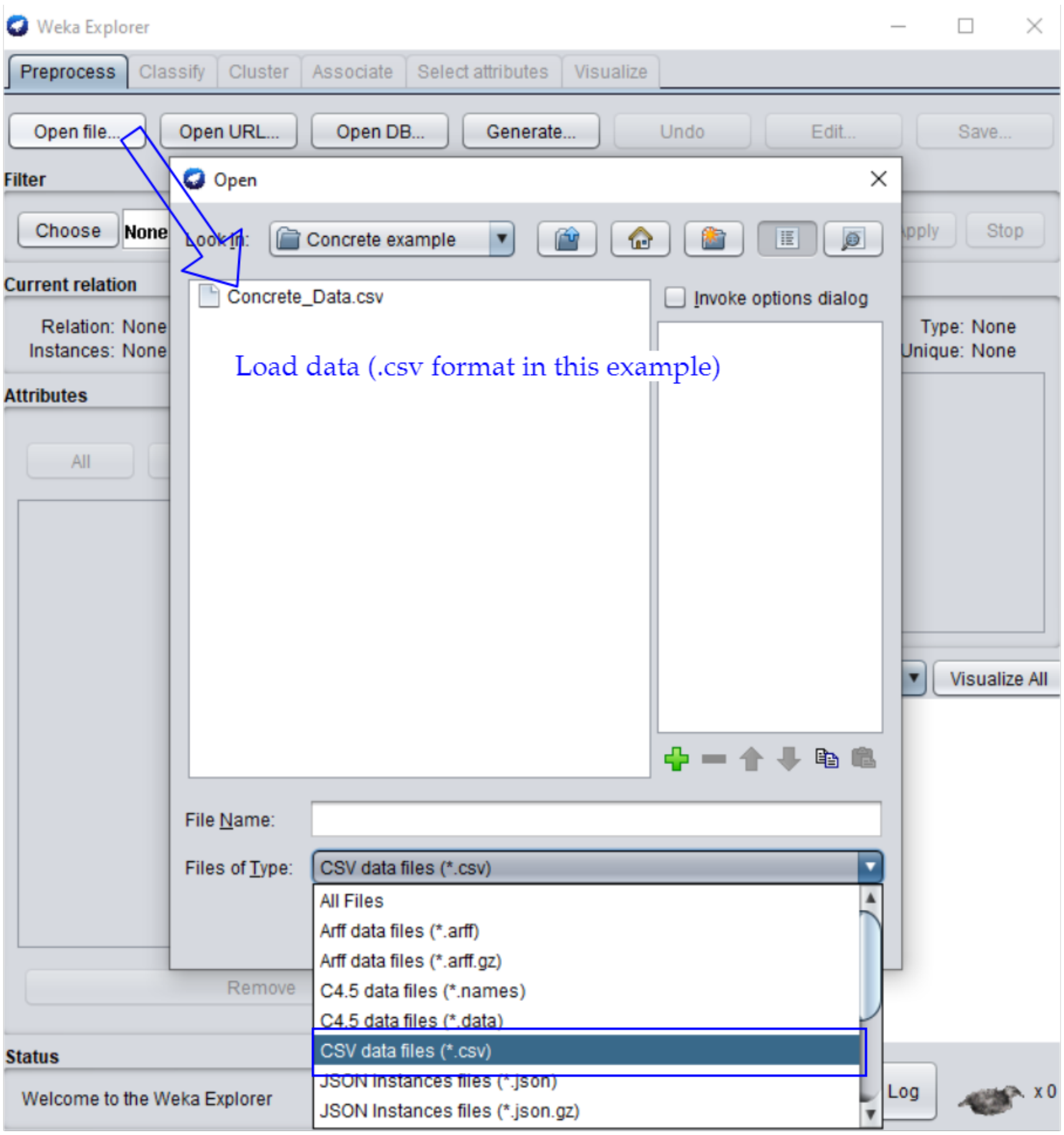


FIGURE C.2: Loading a dataset

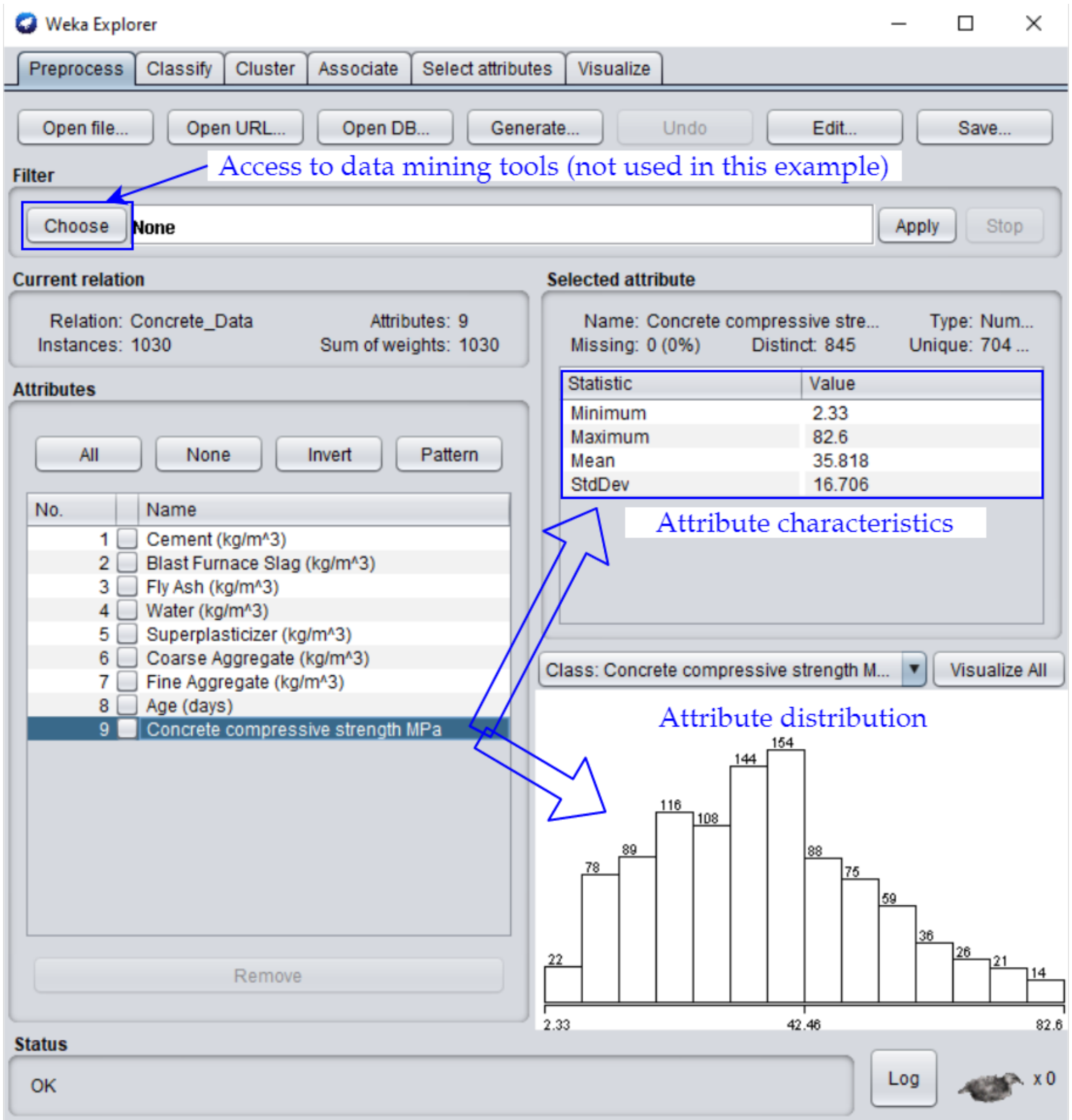


FIGURE C.3: Attribute features

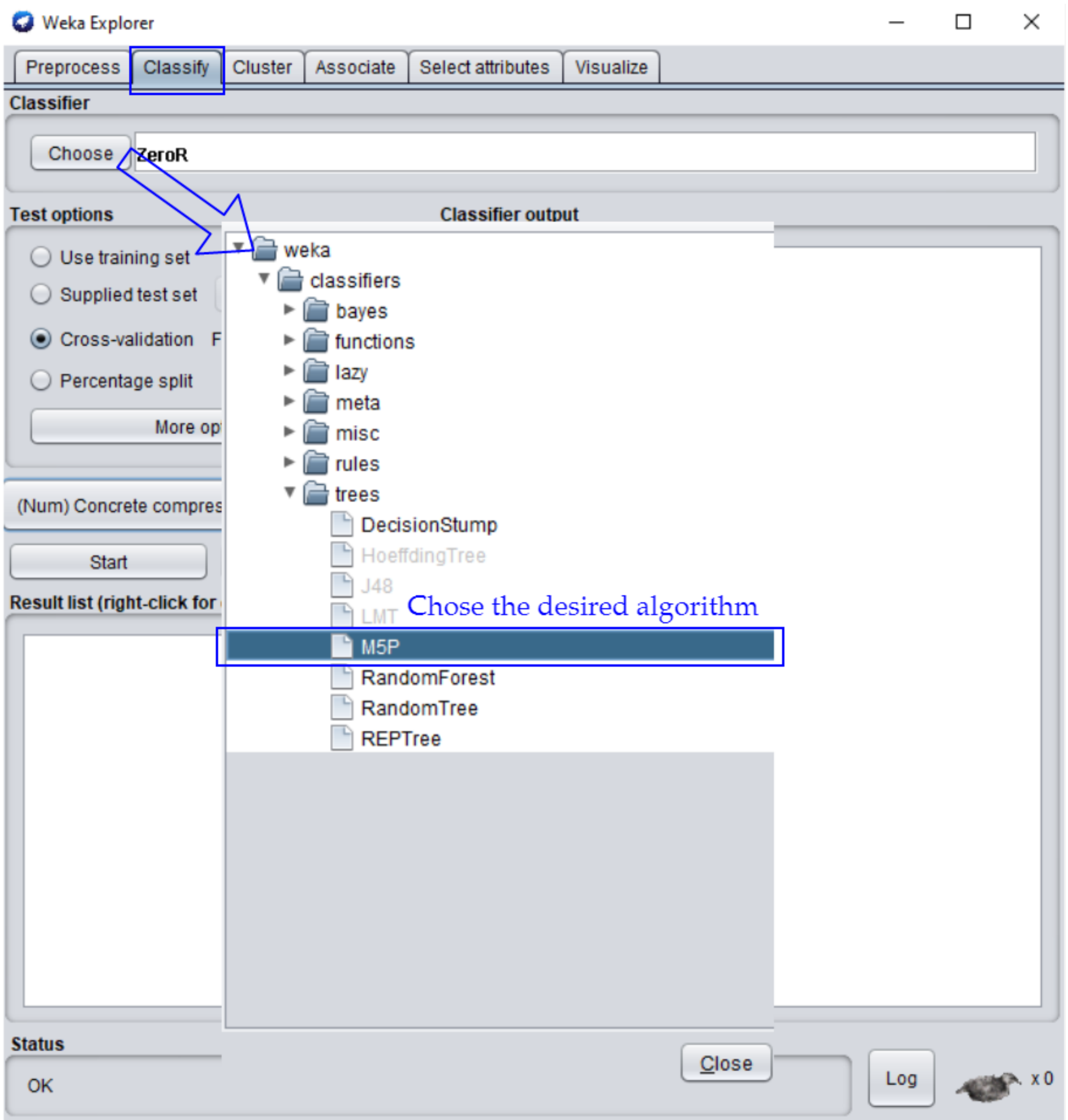


FIGURE C.4: Loading an algorithm

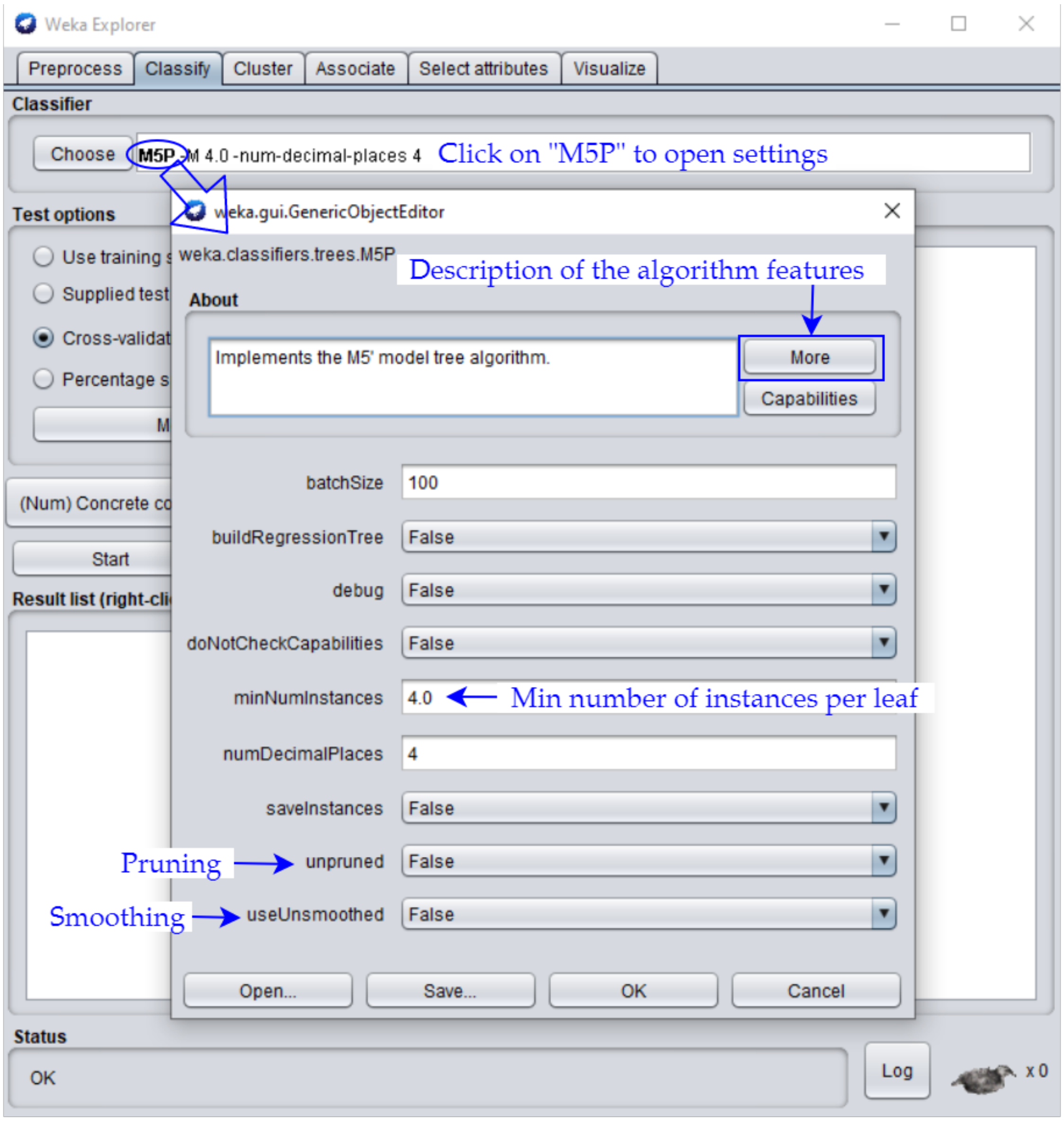


FIGURE C.5: Setting the algorithm features

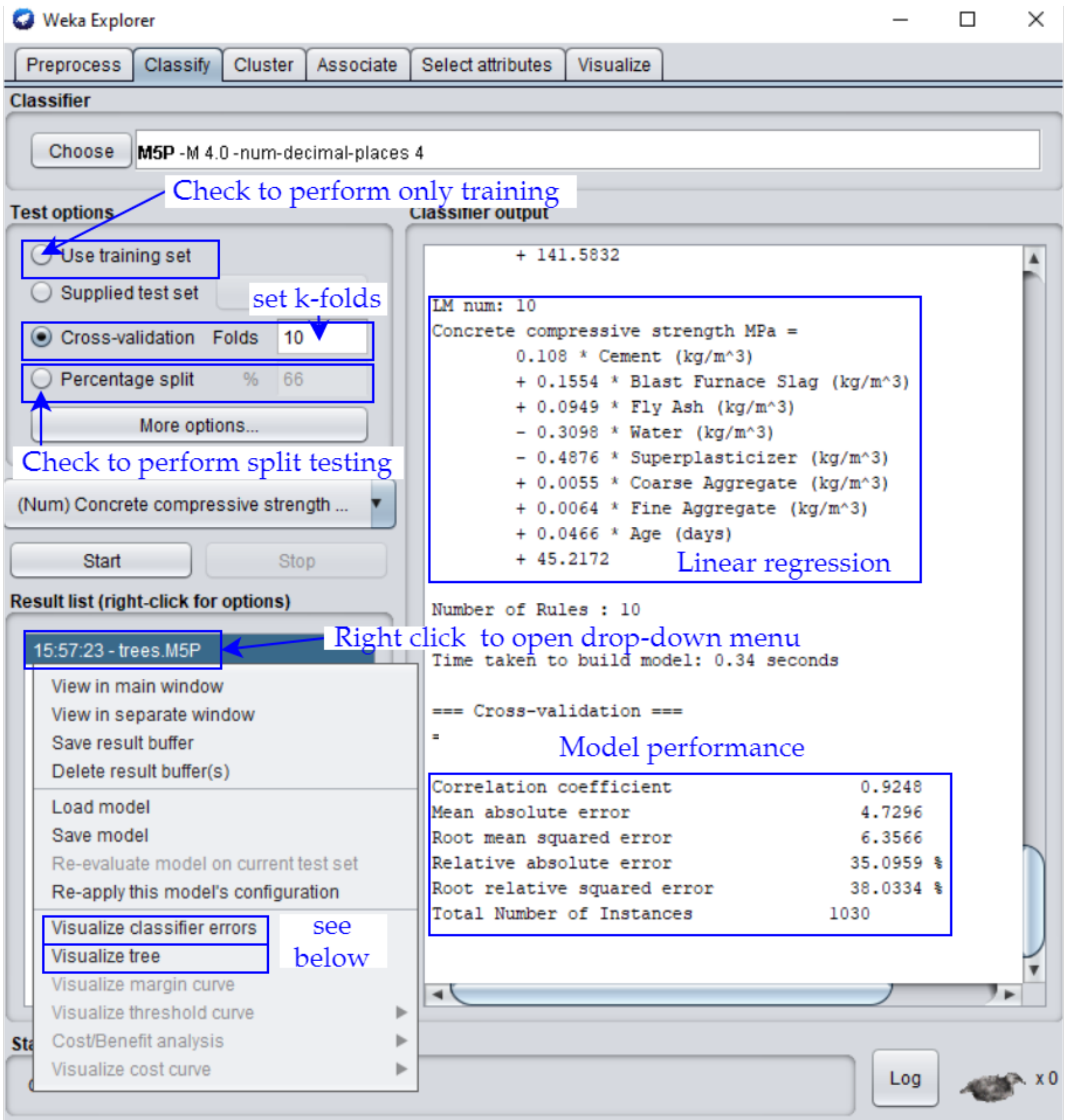


FIGURE C.6: Training a model

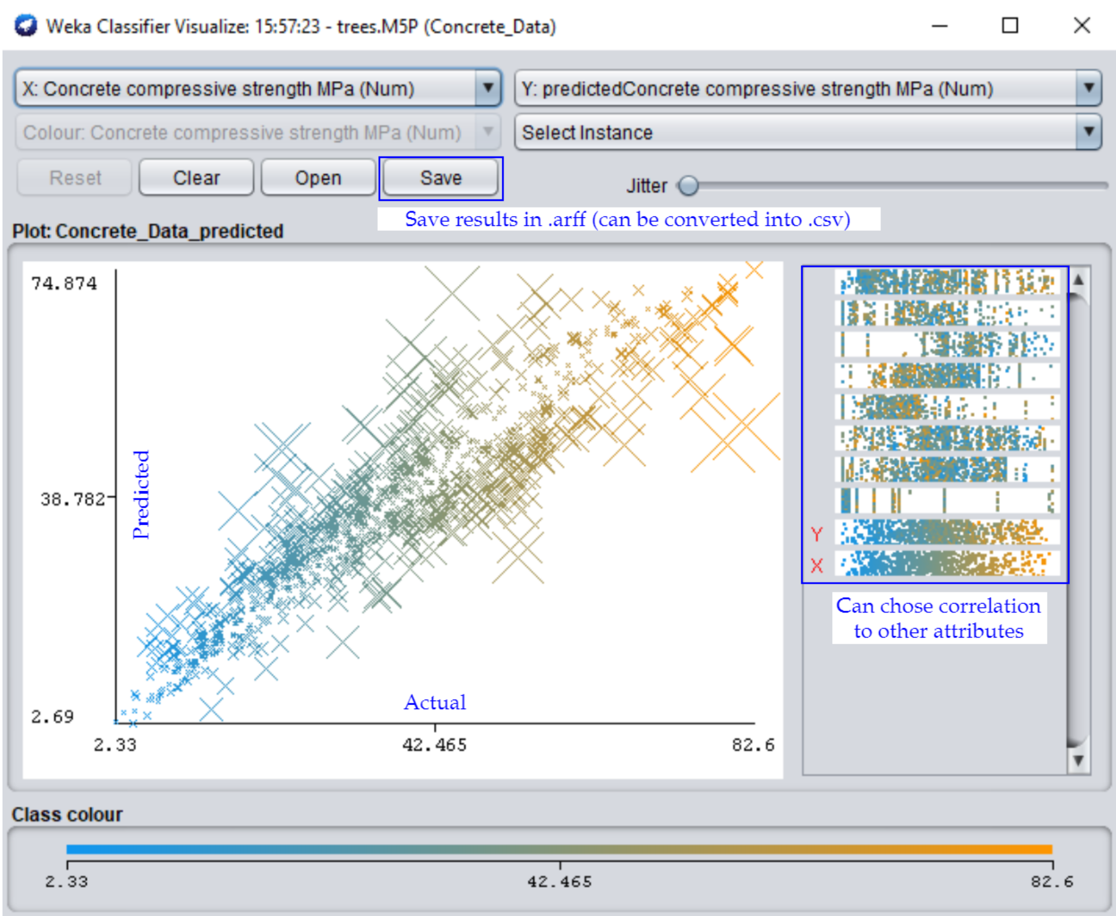


FIGURE C.7: Classifier error visualization

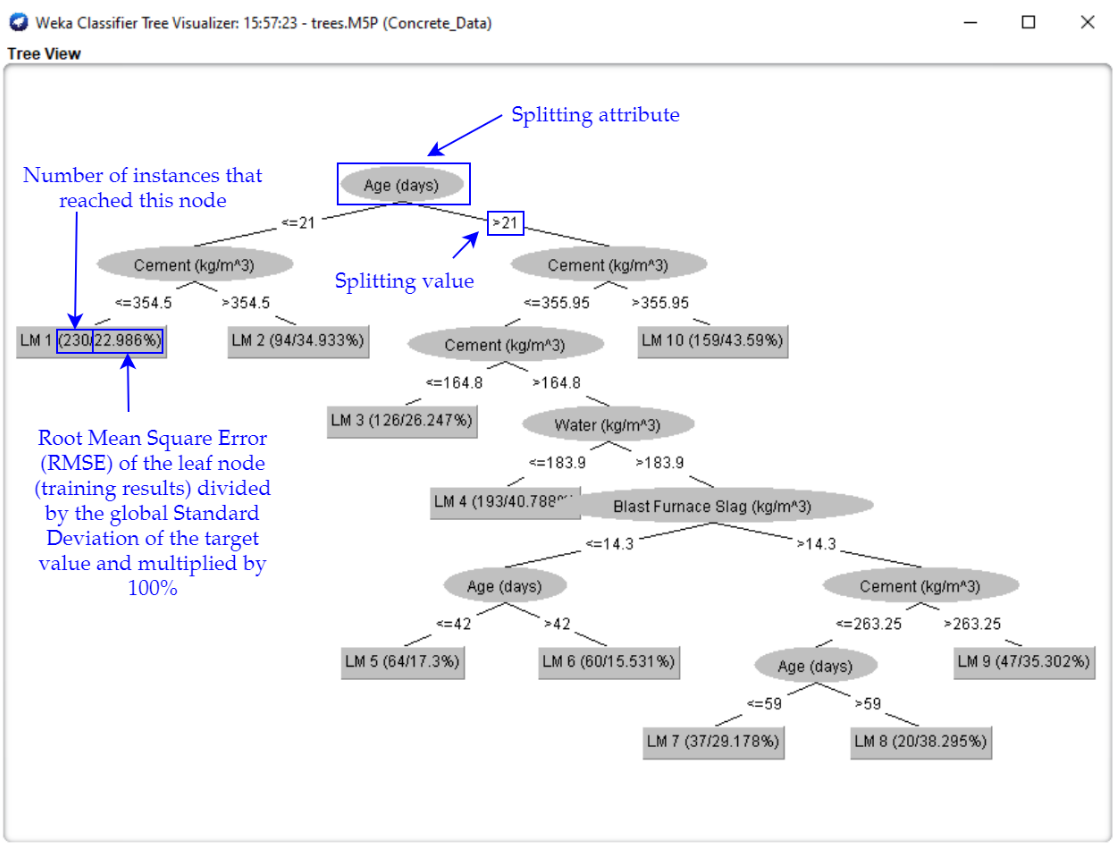


FIGURE C.8: Tree visualisation

# Appendix D

# WEKA model setups from Chapter 5

## D.1 ANN setup

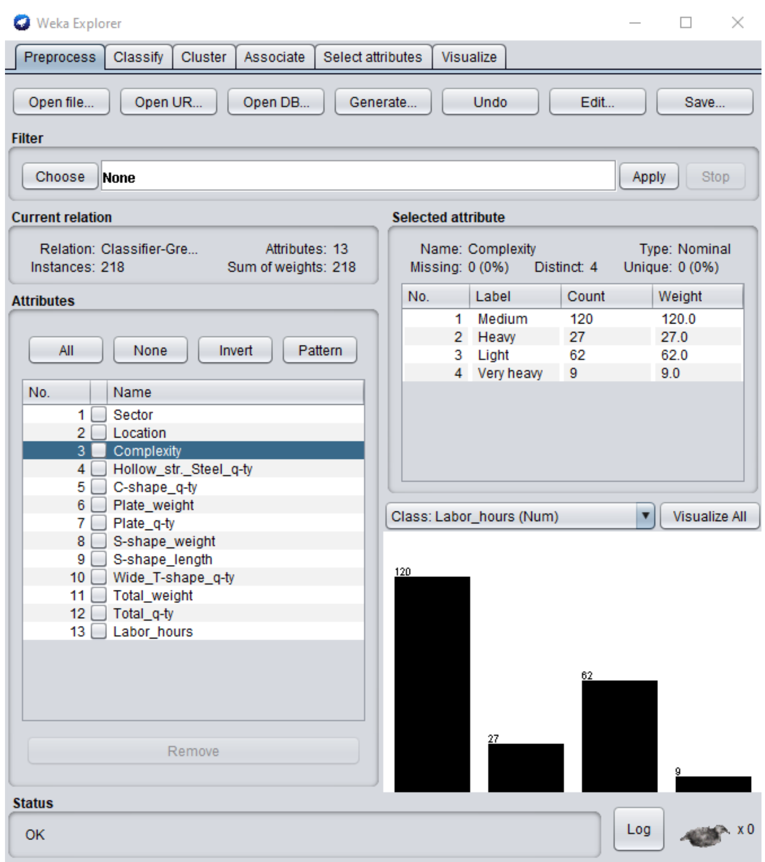


FIGURE D.1: WEKA ANN preprocess

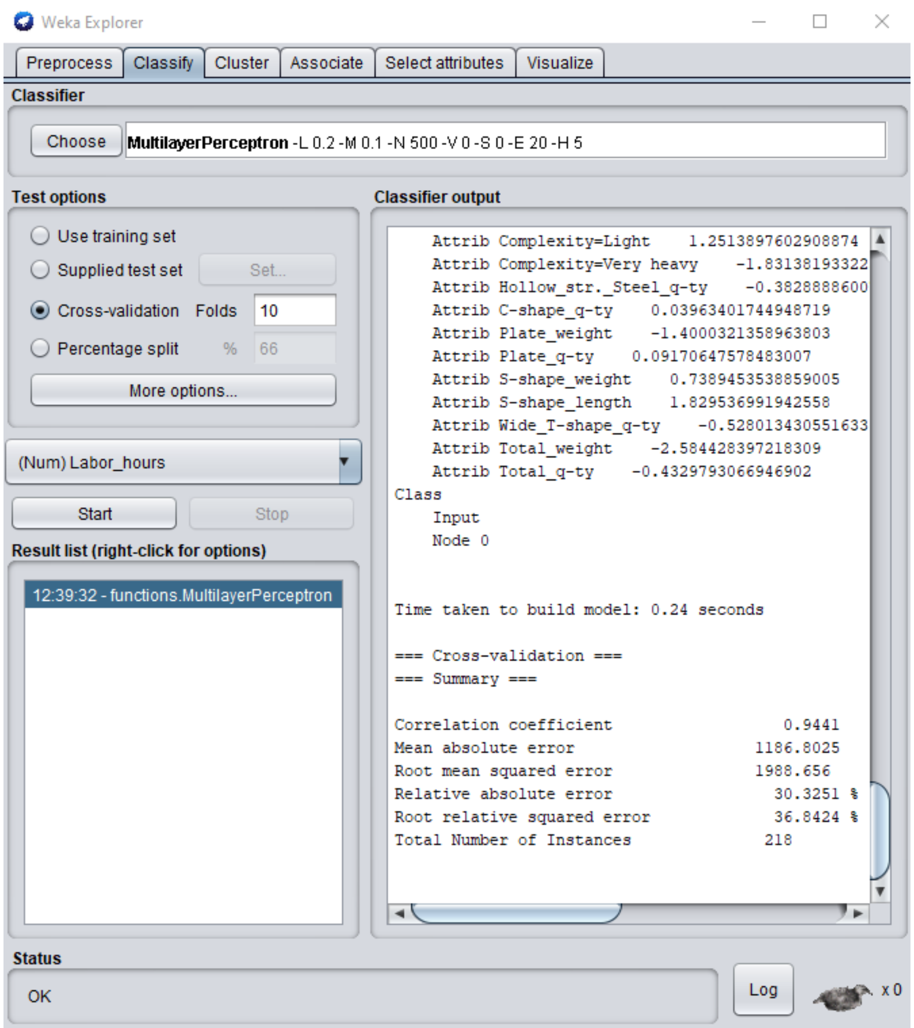


FIGURE D.2: WEKA ANN performance

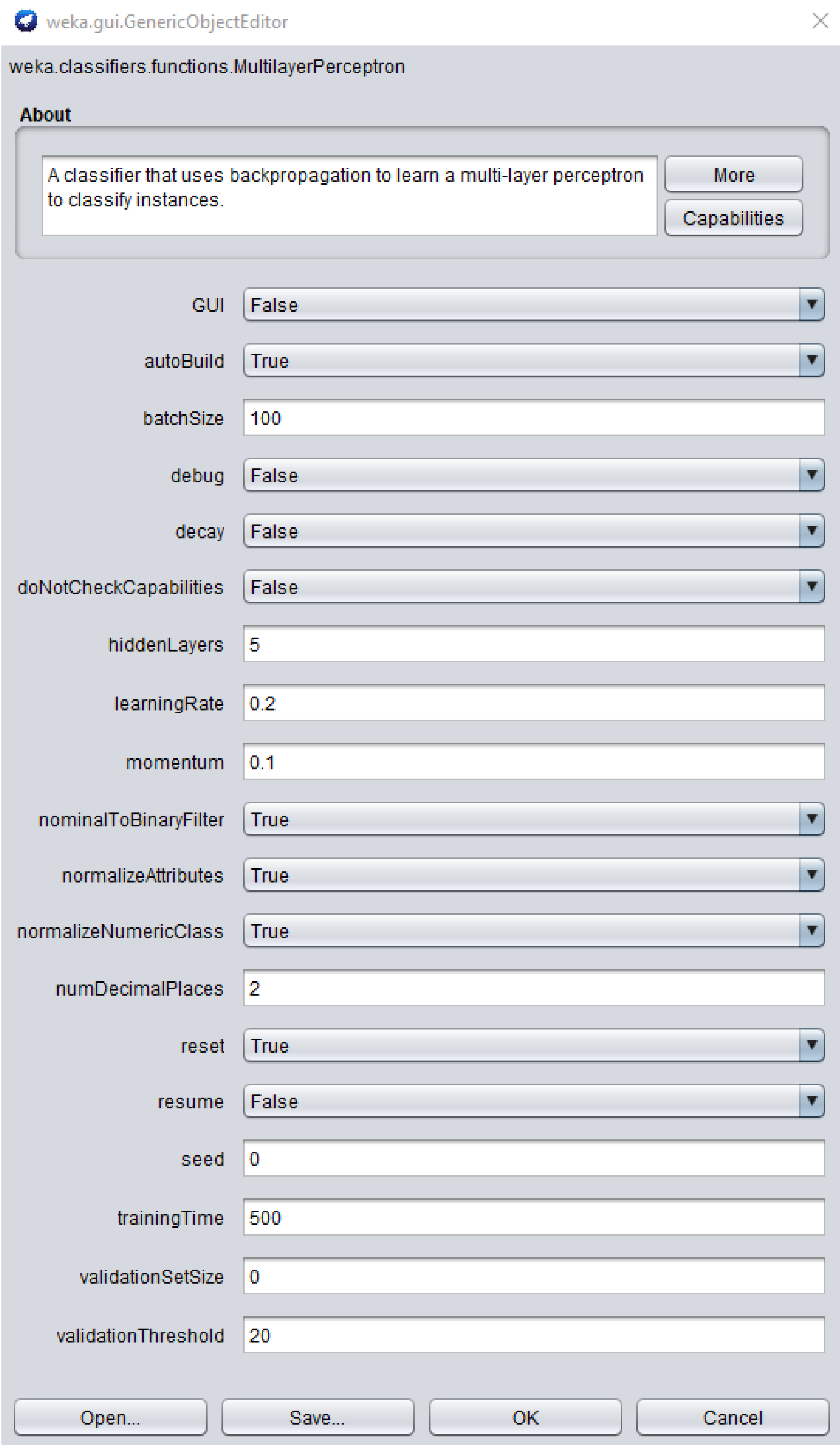


FIGURE D.3: WEKA ANN setup

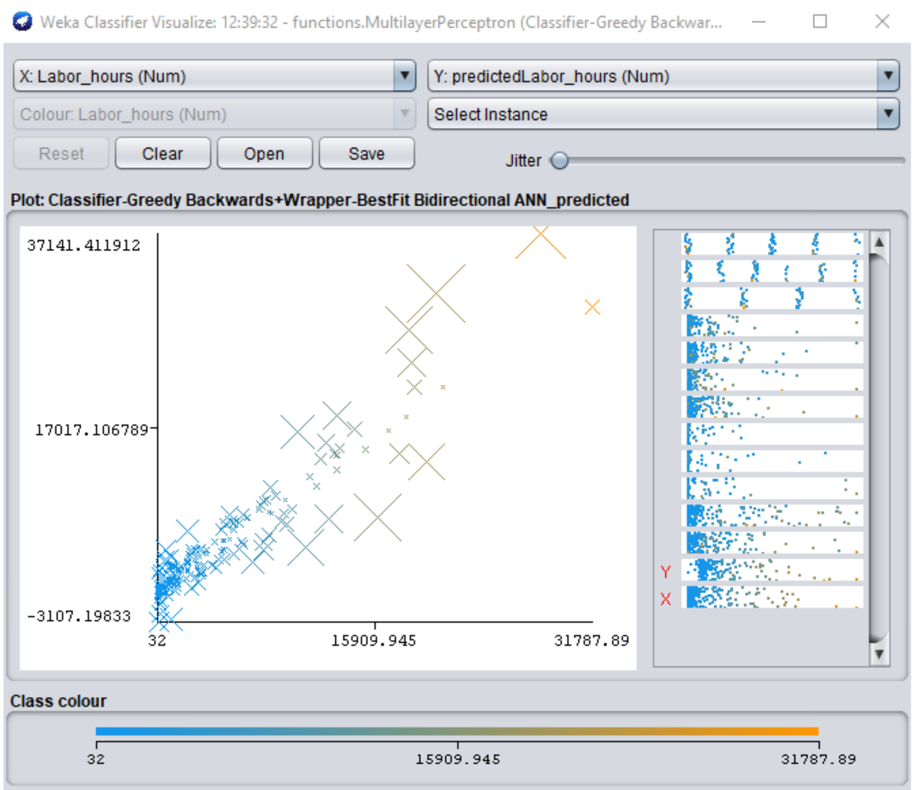


FIGURE D.4: WEKA ANN regression error

## D.2 SVM setup

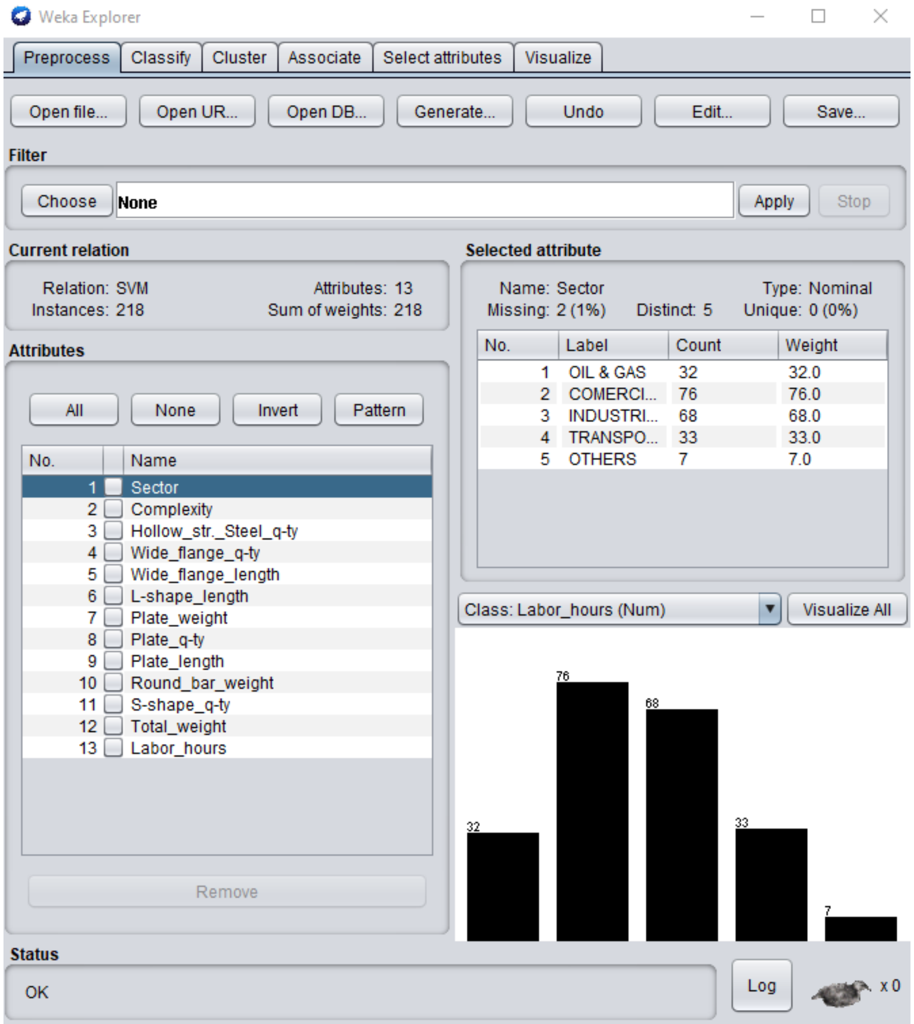


FIGURE D.5: WEKA SVM preprocess

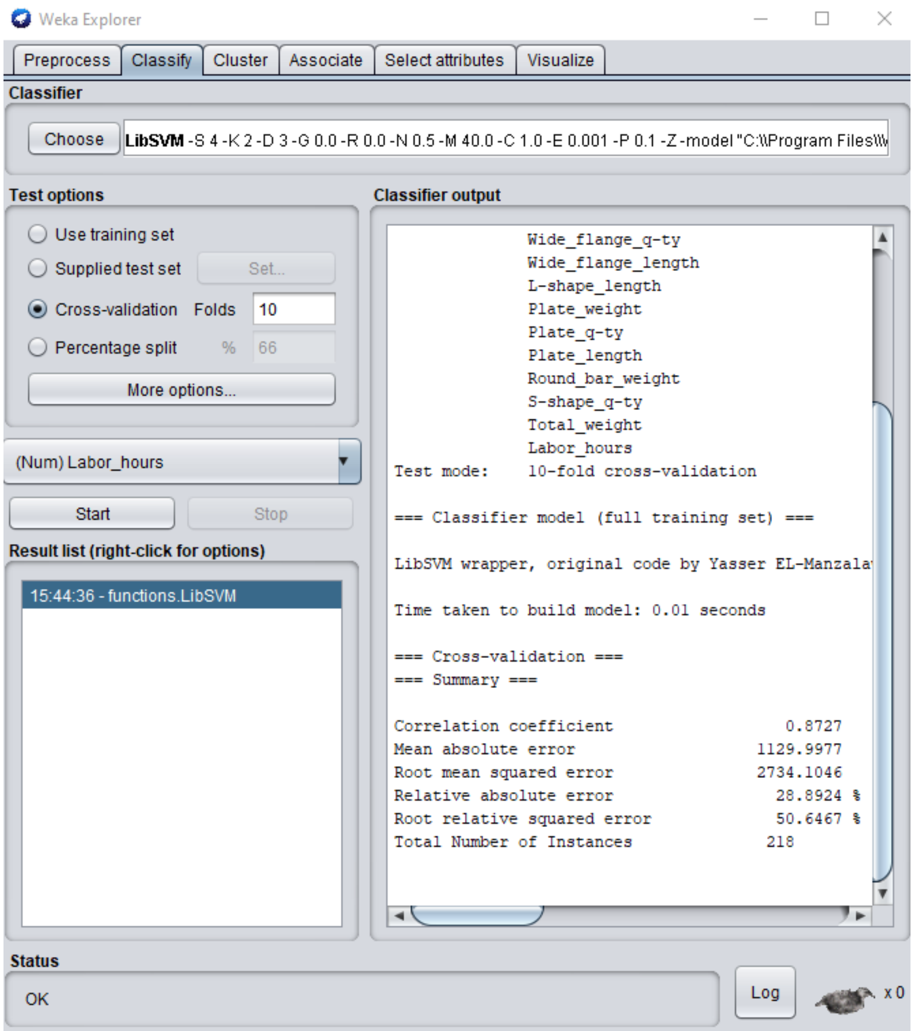


FIGURE D.6: WEKA SVM performance

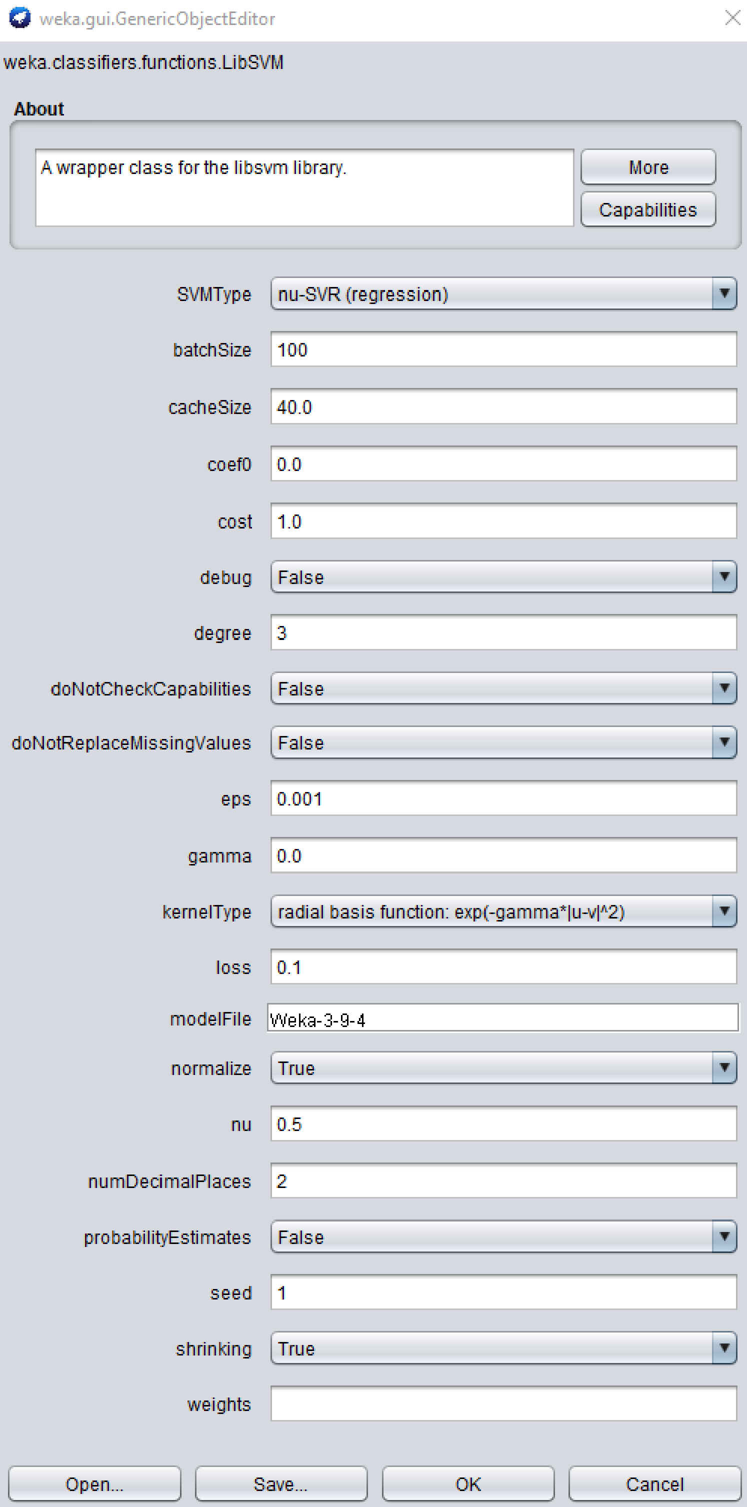


FIGURE D.7: WEKA SVM setup

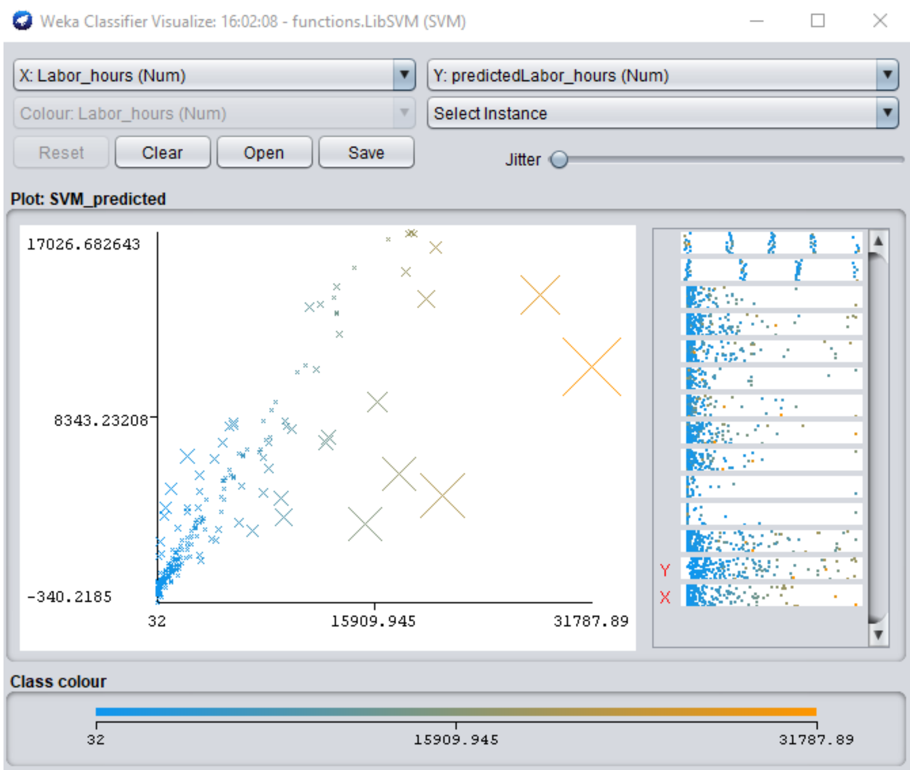


FIGURE D.8: WEKA SVM regression error

## D.3 RF setup

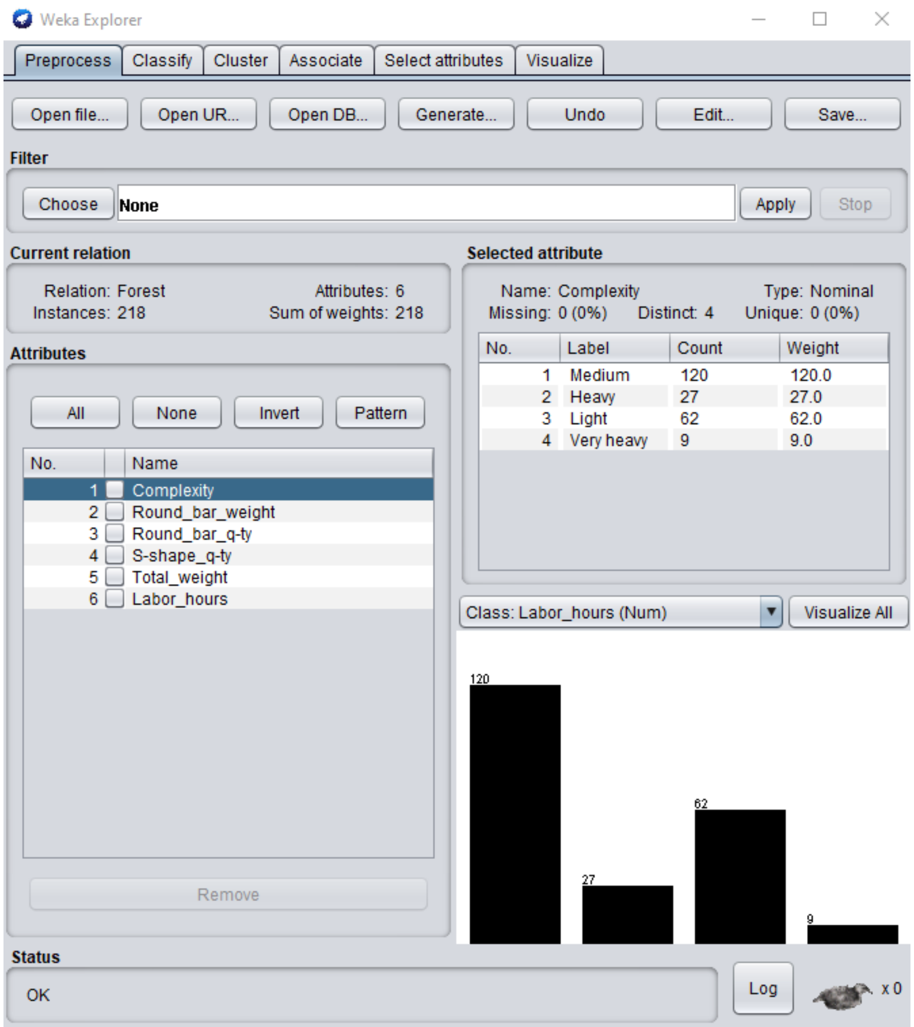


FIGURE D.9: WEKA RF preprocess

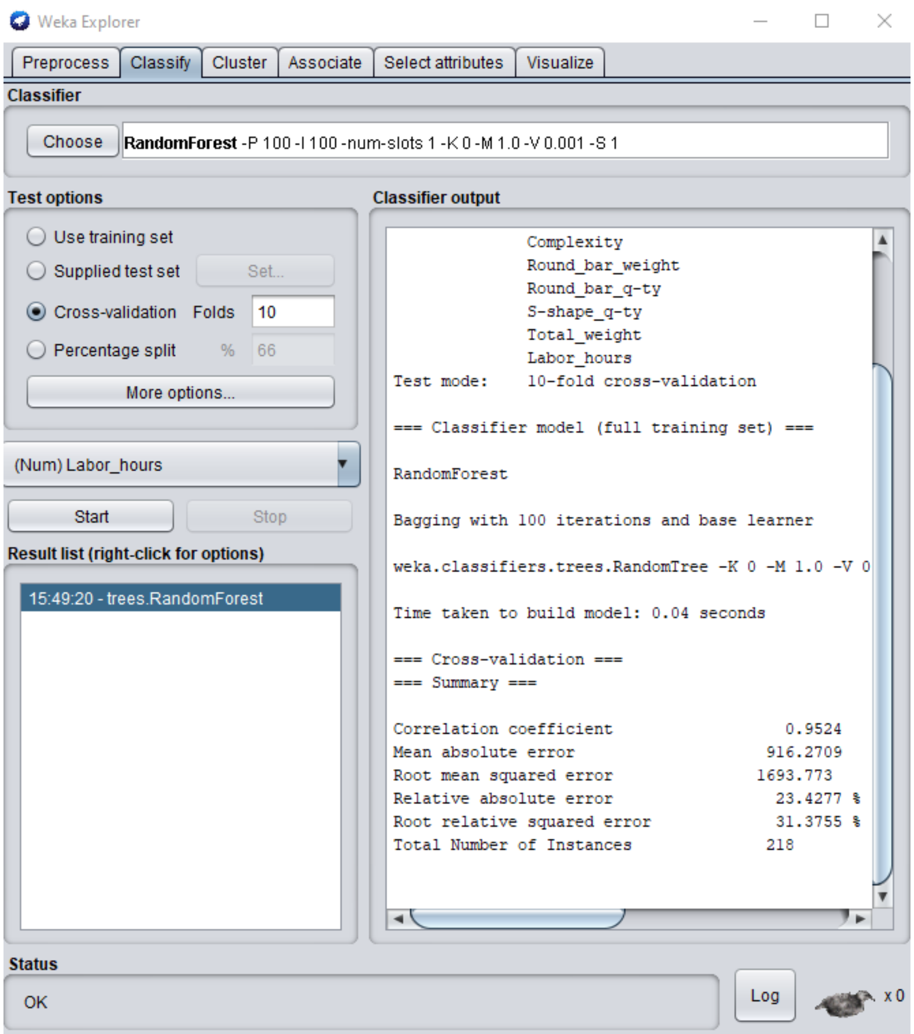


FIGURE D.10: WEKA RF performance

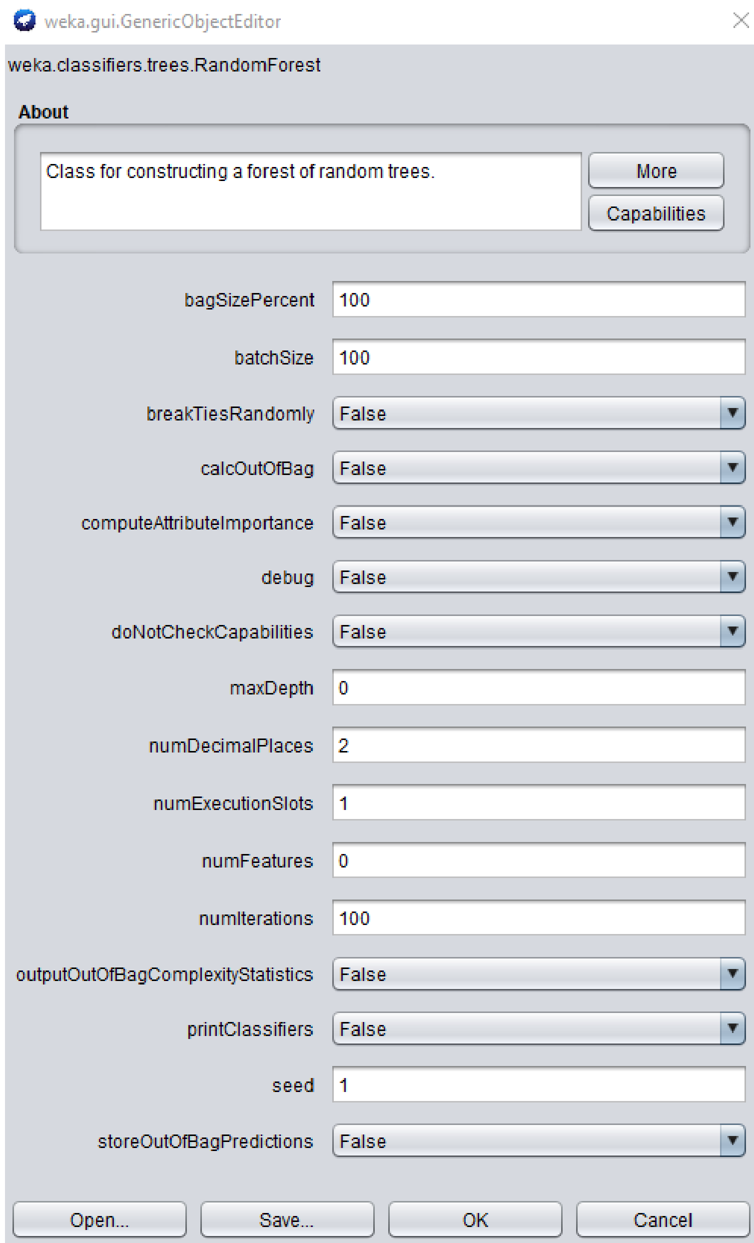


FIGURE D.11: WEKA RF setup

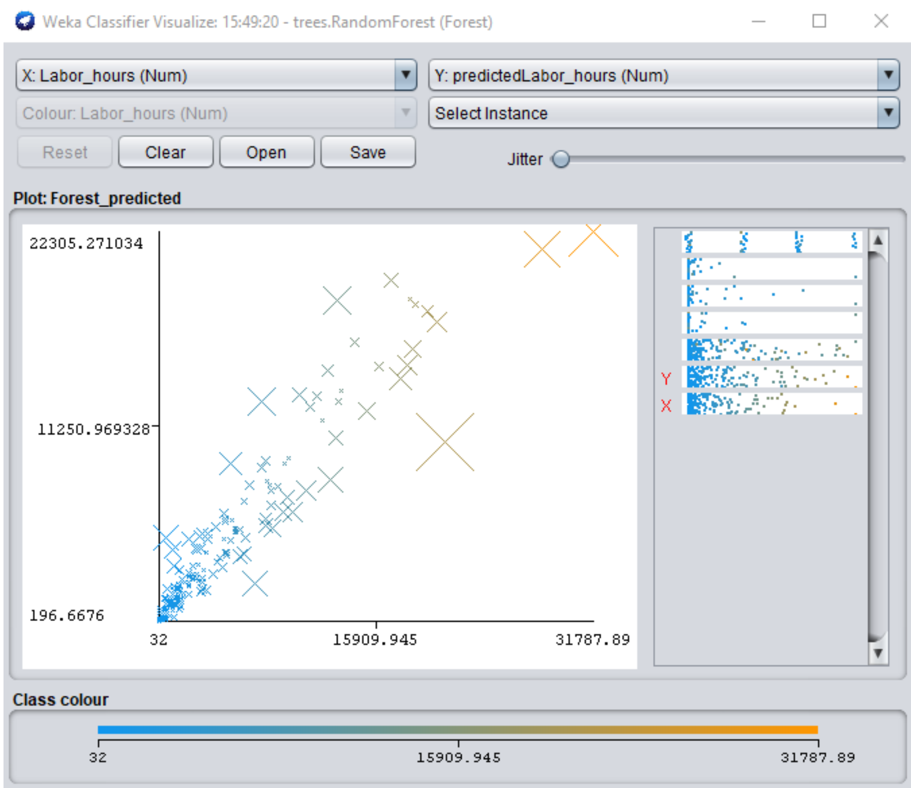


FIGURE D.12: WEKA RF regression error

## D.4 M5P setup

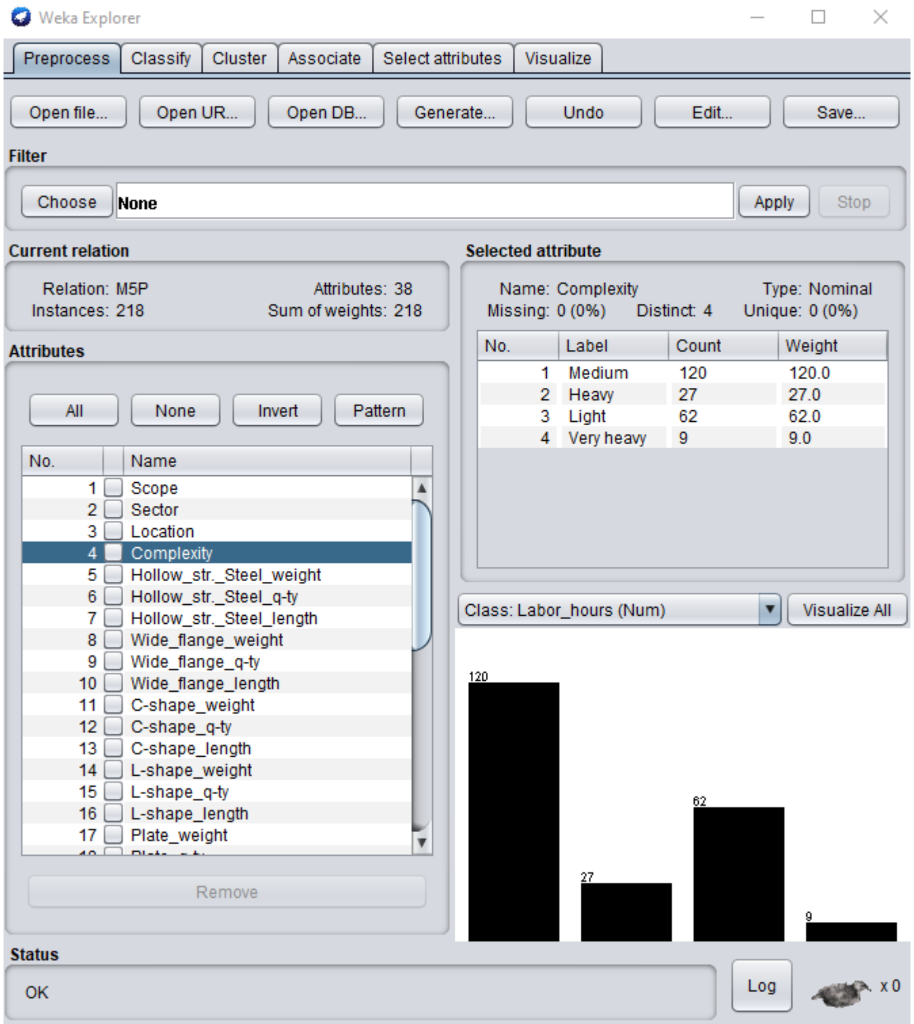


FIGURE D.13: WEKA M5P preprocess

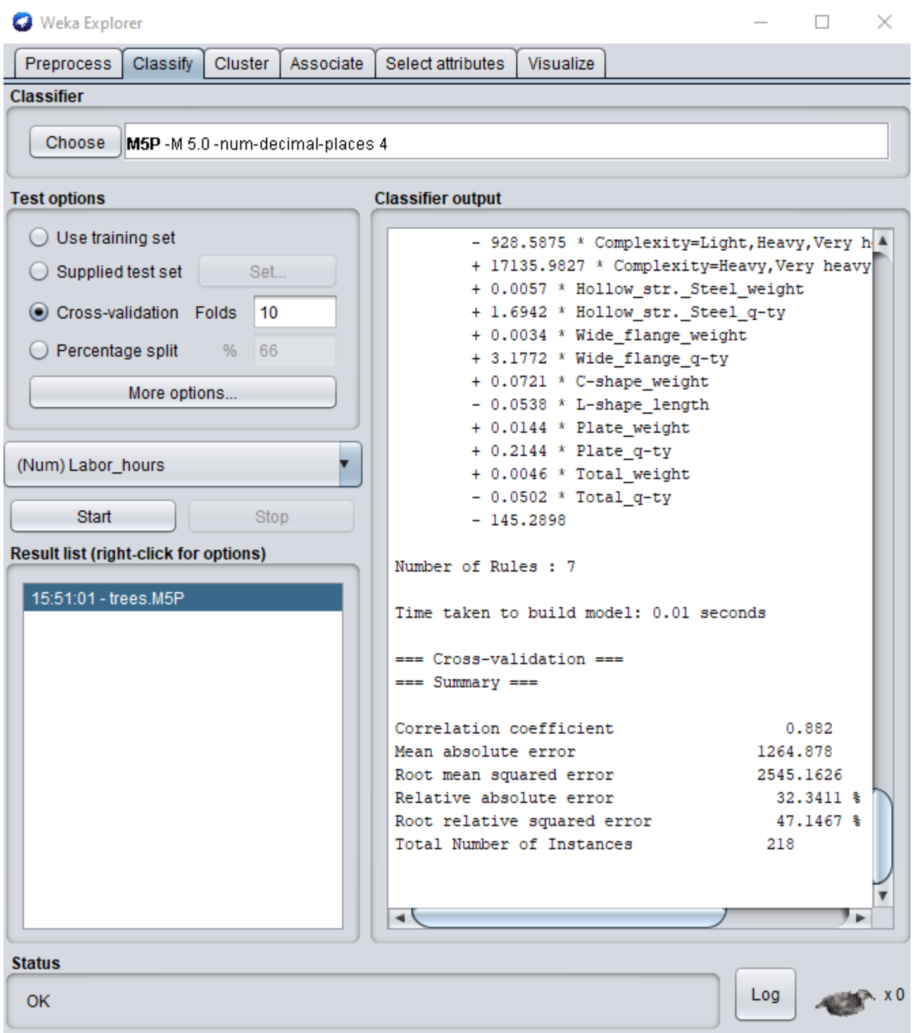


FIGURE D.14: WEKA M5P performance

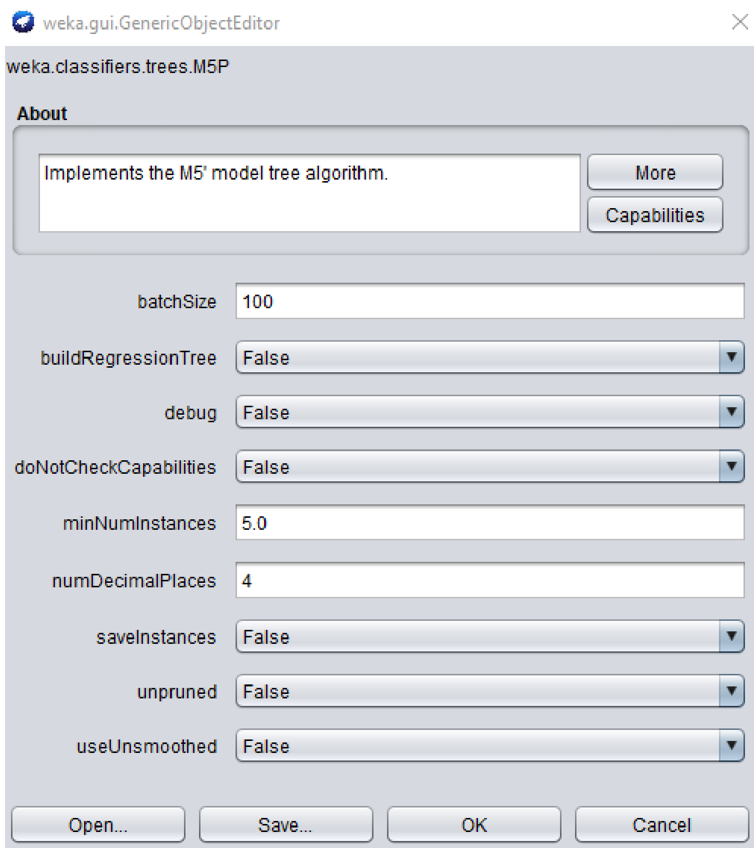


FIGURE D.15: WEKA M5P setup

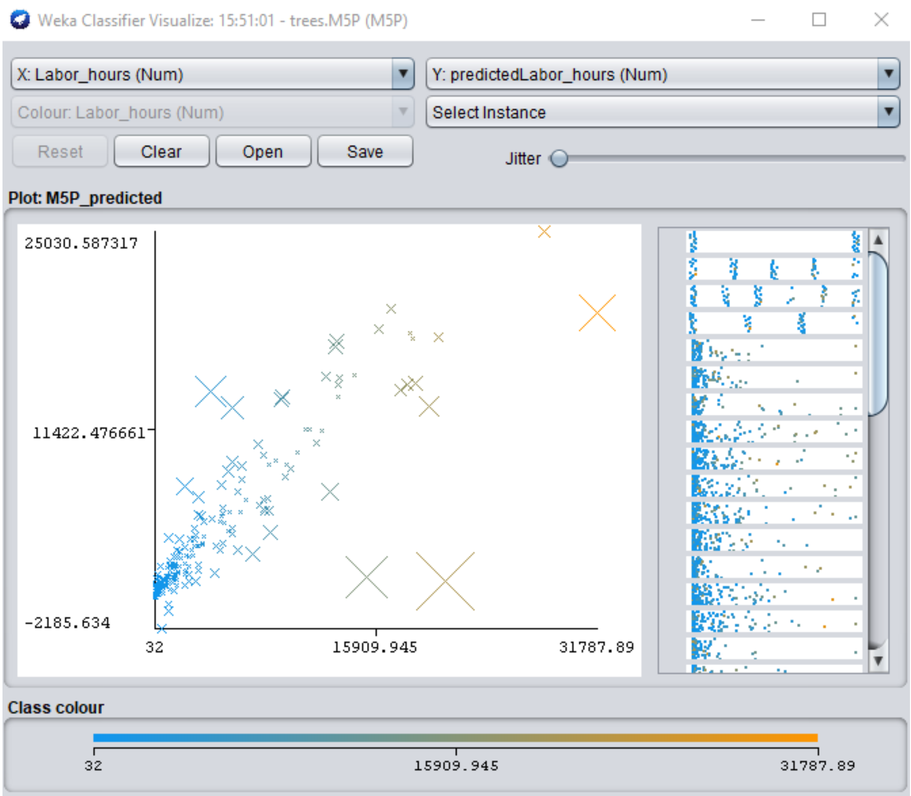


FIGURE D.16: WEKA M5P regression error