UQAC
Université du Québec
à Chicoutimi

Data Pre-Processing: Assistance to Non-Expert Users

par Hans Darmstadt-Bélanger

Mémoire présenté à l'Université du Québec à Chicoutimi en vue de l'obtention du grade de Maître ès sciences (M. Sc.) en Informatique

Québec, Canada

RÉSUMÉ

Dans un projet typique d'apprentissage machine, l'étape de prétraitement des données est la plus longue et l'une des plus importantes. La qualité du prétraitement peut avoir un impact important sur le résultat de l'analyse. L'importance de la réussite de cette étape et la difficulté que les utilisateurs non experts peuvent avoir à effectuer des tâches de prétraitement forment une barrière à l'entrée pour les nouveaux utilisateurs de technologies émergentes dans le domaine de l'apprentissage machine. Le prétraitement est généralement effectué en écrivant un script dans un langage de programmation tel que Python ou R. Toutes les personnes qui ont besoin de prétraiter des données ne sont pas des programmeurs informatiques chevronnés. Dans le secteur de l'assurance, les professionnels de l'actuariat créent et entrainent souvent des modèles d'apprentissage automatique tels que les modèles linéaires généralisés. Ils sont qualifiés en raison de leurs connaissances avancées des modèles mathématiques. Cependant, leurs capacités de programmation sont souvent plus limitées. Ces professionnels de l'actuariat et d'autres utilisateurs non experts ont tendance à consacrer un temps disproportionné à l'écriture des scripts de prétraitement par rapport à leurs pairs ayant une formation dans l'apprentissage automatique ou les technologies de l'information. Nous avons entrepris de développer un programme simple mais puissant pour tirer parti de la compréhension des concepts de pré-traitement des données par les non-experts sans qu'ils aient besoin de savoir comment écrire les scripts pour effectuer les transformations de prétraitement souhaitées. L'outil prend la forme d'un programme de type ETL (extraction, transformation et chargement). Bien que d'autres programmes ETL soient déjà disponibles au public, leur principal objectif est d'homogénéiser les données entrantes avant qu'elles ne soient stockées dans l'entrepôt de données d'une entreprise. La différence d'objectif entre notre outil et les alternatives existantes se traduit par un ensemble de fonctionnalités différentes pour l'utilisateur final. Nous examinons le programme qui en résulte, son architecture et le temps qu'il permet de gagner, qui tourne autour d'une réduction de 50 % dans des conditions idéales.

ABSTRACT

In a typical machine-learning project, the data pre-processing step is the most time consuming and one of the most relevant steps. The quality of the pre-processing can have important impacts on the result of the analysis. The importance to get this step right is contrasted with the difficulty non-expert users may have to perform pre-processing tasks. Pre-processing is typically performed by writing a script in a programming language such as Python or R. Not every person that needs to pre-process data is a well-versed computer programmer. In the Insurance industry, actuarial professionals often build and train machine-learning models such as generalized linear models. They are qualified due to their advanced knowledge of mathematical models. However, their programming abilities are often more limited. These actuarial professionals and other non-expert users tend to spend a disproportionate amount of time to write the pre-processing scripts as compared to their peers with a career machine learning or information technologies background. We set out to develop a simple, yet powerful program to leverage the non-experts understanding of data manipulation concepts without needing them to know how to write the scripts to perform the desired pre-processing transformations. The tool takes the form of an ETL (extract, transform, and load) type of program. While other ETL programs are publicly available already, their main objective is to homogenise incoming and new data before it gets stored in the client company's data warehouse. The difference in focus between our tool and the existing alternatives translates into a different feature set for the end user. The resulting program, its architecture, and the amount of time it saves, which hovers around a 50% reduction under ideal conditions, are examined.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

The abbreviations listed below are used in the present Master thesis.

AOI	Attribute of interest
API	Application Programming Interface
CGIC	Co-Operators General Insurance Company
CRM	Customer relationship management
CRISP-DM	Cross-Industry Standard Practice for Data Mining
DSM	Data Science Methodology
ETL	Extract, transform, and load
FN	False Negative
FP	False Positive
ID3	Iterative Dichotomiser 3
IDE	Integrated Development Environment
IoT	Internet of things
IQR	interquartile range
ISBSG	International Software Benchmarking Standards Group
KNN	K-nearest neighbors
MICE	Multiple Imputation by Chained Equations
ML	Machine learning
MLG	Multinomial logistic regression
OOP	Object-oriented Programming

PCA	Principal Component Analysis
RFE	Recursive Feature Elimination
SQL	Structured Query Language
UI	User interface
TN	True Negative
TP	True Positive

DEDICATION

Dedicated to my parents and family,

Your support and encouragement motivated me to pursue higher education goals and longer projects. You helped to keep me going when motivation was getting hard to find, particularly during the isolated months of the Covid-19 lockdowns. Without you I would most likely not have finished this project and master's thesis. I am grateful to you.

ACKNOWLEDGEMENTS

I want to thank my professors Bruno Bouchard, Julien Maitre, and Bob-Antoine Jerry Menelas who supported me during the project. The proactive approach of checking-in frequently even when I was in constant contact with Co-Operator employees was really appreciated.

I also want to thank the great Co-Operator employees I had the chance to work with. The way they integrated me in the company and in the research group made me feel like I belonged to the team rather than being an external asset to the company.

Finally, I want to thank Charles Mecheriki, the other student who interned with me on this project. Without his contributions, the result would undoubtedly have been of a lower quality.

CHAPTER 1

INTRODUCTION

1.1. DATA PROCESSING IN MACHINE LEARNING

Machine-learning (ML) algorithms have existed since the 1950s [1], at least in theory. Their potential benefits were understood early. Nevertheless, several decades passed until these benefits could be realized. The initial impact of ML was significantly limited by the availability of data and computing power. Both resources are becoming increasingly abundant, and breakthroughs are becoming more frequent in various areas.

With increased success, mainstream adoption followed. This was accompanied by a shifting ML user profile. Initially, the users were computer specialists. However, with increased democratisation, many non-expert users started to deal with ML technologies. The data analytics process is comprised of multiple steps as shown in Figure 1, which these new users may find challenging and by extension, time consuming. Of these steps, data preparation or pre-processing is usually the most time consuming. More precisely, according to [2], the time spent on data pre-processing represents some 50% to 80% of the total time of a ML project.

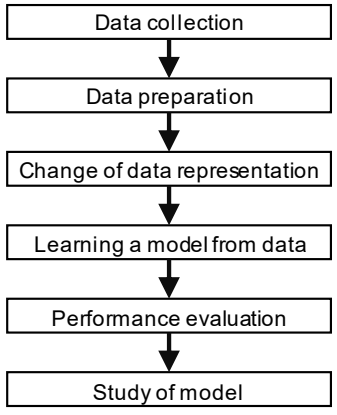

Figure 1: Steps of the data analytics process [2] © Hans Darmstadt-Bélanger

1.1.1. DATA PRE-PROCESSING

Pre-processing the raw data consists of several steps [3-5]. First, the data must be cleaned. Reasons for this include:

- the dataset has missing data,
  - for example, due to an offline sensor, or survey responders that skipped a question,
- the dataset contains impossible values,
  - for example, a negative weight, caused by a faulty sensor, or a manipulation error,
- the dataset contains data of different types,
  - some ML algorithms only work with certain types of data, mostly numbers. Encoding text attributes to numerical values is needed,
- various datasets will be used in conjunction, but they have different formats.

The last example is more common than an outsider may intuitively believe. For example, in the insurance industry it is a common practice to procure data from a data broker [6]. Usually, the formats of the procured and internal data differ. Thus, some treatment of the procured data is required before it can be used by the insurance company.

Once the data has been cleaned, it must be optimised for ML. After cleaning, the ML algorithm might process the data without returning an error. However, the computing time might be unacceptably long. Therefore, another sub-step of processing consists in changing the data representation [7]. This step consists of identifying the optimal set of transformations for the dataset and applying it. Normalising, standardising, encoding the data, and dimensionality reduction transformations are examples of transformations without which the ML algorithm's performance might be unsatisfactory.

One reason why data pre-processing is so time consuming is that real world data usually differ widely. Faults in the data do not always have the same shape. Thus, automating the dataset's pre-processing can be a challenging task. Therefore, usually the pre-processing step

in a ML project is performed manually. Even if time consuming, it is a mandatory step as putting the raw data with no pre-processing applied onto it into a ML algorithm usually yields poor results [8, 9].

1.1.2. PERSONS PERFORMING DATA PRE-PROCESSING

Companies would gain from making pre-processing tasks accessible to non-expert users to free the expert users for other specialised tasks. Various technologies have been developed to assist users in the pre-processing tasks of a ML project. Some of them are aimed at non-expert users. Nevertheless, they have had limited success in speeding up the pre-processing tasks. This may be surprising given the existing frameworks (*e.g.* Weka, RapidMiner, Knime, SAS, and scikit-learn) to facilitate this step [10-14]. Even with an intuitive user interface (UI), the usual functionalities are based on mathematical concepts. These concepts require a certain level of expertise for the user to truly understand what the program is doing behind the scenes. For example, it can be challenging to explain to a non-expert user what a random forest [15] is and how it works, regardless of how easy it is to select the random forest in the program's UI. Furthermore, there are various data pre-processing operations (*e.g.,* discretization, normalization, and standardization). Usually, only experts know which operations are appropriate for a given task. Not to mention the different algorithms allowing to perform the same operation, or the uncertainty regarding if the selected transformer should be applied locally on one or some specific columns of the dataset or globally on every column of the dataset.

Further compounding the situation is the fact that non-expert users may have little to no coding experience. While some of the frameworks mentioned above can be used without having to know how to code, not all of them have graphical UIs. Those that do often have complex menus and controls to manage their impressive array of functionalities. This makes the interface seem convoluted and daunting to new users.

Usually, considerable time is required to identify the optimal, or at least close to optimal, combination of pre-processing operations. For a given dataset and ML-based task, different of data pre-processing techniques and alternative data curation strategies may lead to drastically different results with varying quality performance [16]. The very nature of big data renders it very challenging for humans to identify the best approach to select the optimal transformations. Thus, many rely on experience or exploration akin to trial and error [17].

1.1.3. EFFORT REQUIRED FOR DATA PRE-PROCESSING

As outlined in the previous sections, data pre-processing has significant importance and is time consuming. Thus, unsurprisingly, it is attracting a lot of attention and has become a major subject of research in the past years. As outlined in [2]'s figure 3, preparing the data is perceived as the most critically important step of a modelling project. Nevertheless, it is also perceived as receiving the least amount of energy from the research community [2]. For example, the modelling step is perceived as receiving over five times more energy than the pre-processing step as can be seen in [2]'s Figure 4. That more energy needs to be spent on pre-processing could be considered as an understatement.

It can be summarised that the existing solutions for non-expert users have considerable shortcomings. The objective of the present Master thesis is therefore to develop an interfacing tool for data pre-processing, easy to use by non-experts.

1.2. BIG DATA

Big data refers to datasets of an enormous size when compared to regular databases. Use of big data depends on the ease of data generation and on costs of data processing.

1.2.1. DATA GENERATION

There is more and more data available as its generation becomes less expensive. With the decreasing prices of electronic devices, they are becoming increasingly present in our everyday lives. Increased use of electronics is associated with increased data generation. One prominent example is the data generated by the numerous applications on our mobile phones. Furthermore, numerous household items now are connected to the Internet in one way or another to enable new ways of interacting with them. Smart lightbulbs that can be managed using a mobile phone application are one example. Embedding an internet-connected controller in a lightbulb would have been considered a ridiculous concept two decades ago. With increasing availability, this is rapidly becoming an unremarkable feature for the lightbulb. Everyday mundane items connected to the internet is known as the internet of things (IoT) [6]. This is a relatively recent phenomenon. Just some decades ago, it would not have been cost effective to embed microcomputers in everyday objects.

1.2.2. COSTS OF DATA PROCESSING

Some types of big data have been available before the widespread use of electronic devices discussed above. This includes government registration data, client lists of utility services, and online browsing data from desktop computers. Here, high costs for data processing limited their use. Thus, only if the high costs were judged acceptable, would this data be used in big data applications. One example is the dragnet search used in 1979 in Germany (*Rasterfahndung* in German). Here, filtering of client lists of utility services by using registration data was successfully used to identify terrorist safe houses [18].

Data processing became less expensive with increasing performance of mainframe computers. Afterwards, applications such as the generation of targeted advertisement, based on online browsing data, became commercially feasible.

It can be summarised that large data quantities are now generated and can be processed at reasonable costs. This makes many big data applications feasible.

1.2.3. BIG DATA IN THE INSURANCE INDUSTRY

Commercial applications of big data projects were viable before commercial applications of projects based on data gathered by IoT devices. The insurance industry has seen a shift in where its data comes from and how it gets used. First, the operation about one decade ago is discussed.

1.2.3.1. THE INSURANCE INDUSTRY A DECADE AGO

Today's insurance companies claim archives are a proving to be invaluable. Indeed, decades of past insurance claims give an insight into what characteristics have the greatest correlation between them and the likelihood that a motorist is involved in an accident. For example, on average, male motorists cause more accidents than female motorists [19] [20]. That correlation thus motivates higher insurance premiums for male drivers than for their female counterparts. Nevertheless, there are important limitations. The data points are relatively generic. Using characteristics such as gender, age, years licensed, and location only goes so far as there can be two motorists with similar profiles while having different driving behaviours. Even if on average, male motorists cause more accidents than their female counterparts do, there are reckless female motorists and prudent male motorists. Using general correlations, the risk assessment described above was inappropriate for reckless female motorists and prudent male motorists. The insurance industry realized that individualised risk profiles for each customer enables insurers to provide appropriate premiums for each customer.

1.2.3.2. THE INSURANCE INDUSTRY TODAY

Considering the topic of the present Master thesis, in-vehicle data recorders are discussed below as a relevant example. In-vehicle data recorders track standard telematics variables, such as speed, acceleration, and braking. This data allows the insurer to improve the risk assessment of customers. Thus, the insurers can better tailor their products to the customers' risk profile [21]. This is illustrated by the following example. If a motorist breaks abruptly, it is assumed that the need to break was realized somewhat too late as most obstacles can be anticipated well in advance. Thus, sudden breaking is used to gage how attentive the motorist is to the situation around his or her vehicle. If a motorist frequently breaks abruptly, the probability of causing a road accident is higher as compared to another motorist that mostly breaks softly [22]. Aggressive driving could be described as a combination of the motorist's aggressive accelerating, breaking, and respect of speed limits. All these pieces of information can be gathered by embedded sensors that would fall under the umbrella of IoT devices. The generated data can be classified as big data. Both references [21] and [23] state that about three months of driving data from a motorist is sufficient to access the motorist's accident risk. This relatively short period makes it attractive for insurance companies to launch a project on telematic data.

This is where the individualised risk assessment powered by telematics comes into play. Both approaches are only possible with big data. Some types of data are easier to process and use than others, however.

1.2.4. DATA STRUCTURE

Various data collection points generate data that can be classify as unstructured, semi structured, or structured. This refers to how easy it is to extract information from the data.

A user filling in a form is producing structured data, as the system knows exactly where the information is. This kind of data will usually be stored in a relational database. Accessing the information is usually straightforward.

On the opposite end of the spectrum, unstructured data is for example a car's dash camera video. Before any useful information could be extracted from the data, an image recognition algorithm would presumably need to process the video file. This could classify what was captured by the camera in a way that is readable by the downstream processes. Due to its large size, unstructured data is costly to store. Additional costs arise during treatment required for their use.

Semi structured data sits in the middle. It is not fit for being stored directly into a database but has some structure to its data. XML files are an example of semi-structured data as they have tags allowing knowing exactly what data comes next, but these tags are not too easy to be quickly found by searching functions.

1.2.5. THE 4 VS OF BIG DATA

Another set of characteristics used to quantify big data are what is known as the 4Vs: volume, velocity, variety, and veracity [24].

Volume is self-explanatory as it refers to the amount of information that represents a dataset. The volume of the data affects the time, complexity, and costs associated with processing the data.

Velocity characterises the speed at which the data is generated. Of course, this data needs processing and storing. Many devices collect data in real time and stream it to a server. Though not directly connected to the field of insurance, medical monitoring devices can upload data multiple times a second. These devices can alert a barely autonomous person's entourage when they need assistance or if they show sign of an impending health crisis. With the data being always generated, it must also be processed in real time to prevent losses of data. If the

requirements in network bandwidth, hard drive space, or processing power are not being met, data is lost.

Variety represents the quality and the type of the information being produced. A piece of data can be of any format and has value so long as it is possible to derive information from it. As such, the data could be in text, audio, video, image, or any other format.

Veracity quantifies the quality and reliability of the data. Should the data be corrupted or missing, deriving accurate results from it could hardly be expected.

Tying the big data concept to the research context of the present Master thesis, real world data is messy. As outlined in the previous section, raw data in and of itself cannot be used to derive a meaningful conclusion. It is therefore mandatory to pre-process the data adequately if useful results should be obtained.

1.3. PROBLEM STATEMENT

This project was carried out in partnership with Co-Operators General Insurance Company (CGIC), a Canadian insurance company. After reviewing its needs, CGIC concluded that a typical use for such a tool would be to assist actuarial professionals working with ML models already in place. Thus, there is little need to implement a new ML model. Instead, the actuarial professionals need to fine-tune the existing models for use with new data.

For similar reasons, the present project does not aim to implement an automated way of identifying the optimal pre-processing transformations for a given ML algorithm, as the actuaries typically know what they want. They found themselves lacking in coding proficiency, which causes the actuaries to spend a disproportional amount of time to code a script that would seem trivial to a seasoned programmer.

Therefore, the objective of the present Master thesis is to minimize the time spent on data pre-processing by the actuaries. This is to be achieved by developing an interfacing tool, easy

to use, and educational for non-experts. The tool should allow them to perform the desired pre-processing tasks without the need to learn or apply any code.

CHAPTER 2

LITERATURE REVIEW

Pre-processing is first and foremost a step in the ML methodology. Its purpose is to prepare the data for the next stages of the ML process. To better understand the place pre-processing occupies, an overview of the ML process, based on the Cross-Industry Standard Practice for Data Mining (CRISP-DM) [25] is outlined next and illustrated in Figure 2.

2.1. CRISP-DM

Another name that generally refers to the same concept would be the Data Science Methodology (DSM). The following steps help the analyst to keep track of the phase of the analysis he or she is performing.

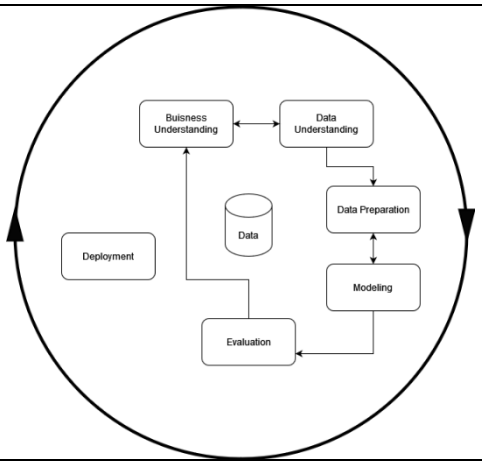

Figure 2 : Cross-Industry Standard Practice for Data Mining. © Hans Darmstadt-Bélanger

2.1.1. BUSINESS UNDERSTANDING

The first step of a data science project is to understand the business objective. The analyst should understand, from a business perspective, what the customer aims to accomplish. Based on these needs, the situation is assessed. This includes determining the available data to be used, the project requirements, and conducting a cost-benefit analysis. The data mining goals then get determined. The main question that gets asked here is “from a technical perspective,

what does success look like?". Based on this stated goal, the project plan is produced. The technologies and tools are also chosen here.

2.1.2. DATA UNDERSTANDING

The Data Understanding step consists of collecting the data and assessing it on a high level. The analyst examines and summarily documents the data by noting properties like data format, number of records, or field identities. Once the analyst has a surface level understanding of the data being explored, he or she investigates the technical quality of the data. Any data quality issue is documented. If too many quality issues are found, the analyst should not be shy about securing another dataset before moving into the next steps.

2.1.3. DATA PREPARATION

The data preparation step includes data pre-processing as discussed in the introduction. The pre-processing step is the most time-consuming step of the entire process. The analyst determines which datasets are going to be used. The reasons for using or excluding various datasets are documented. Then, data pre-processing takes place. The data pre-processing process is described in detail in paragraph 2.5. In short, the data is cleaned, new attributes are derived from the data, if applicable the various datasets are combined, and the data gets re-formatted.

2.1.4. MODELLING AND EVALUATION

The modelling step consists of building the predictive ML model itself. This is done iteratively based on the results the evaluation steps produce. These two steps are therefore grouped under the same title here.

First, the analyst must determine which algorithms to use (*e.g.* random forest [15], c4.5 [26]). Before training the algorithm, the data is split into 3 subsets called the training set, the test set and the validation set. The reasons for doing so are explained next.

One important task in the evaluation step is to quantify the performance of a machine learning algorithm. The performance of an algorithm can be equated to the model's accuracy in the case of classification tasks. The accuracy is the proportion of examples where the model outputs the correct value. In the case of regression tasks, such as density estimation for example, the performance must be calculated using a different metric. The average log-probability the models assigns to some examples will be the performance metric used in these cases [27].

We are usually interested in how well the machine learning algorithm performs when faced with data it has not seen before, as this determines how well the algorithm will perform in the real world. This is why the performance measures need to be gathered using the test set, as its data is not present in the training dataset. The ability to perform on previously unseen inputs is called generalization. In a typical case when training a ML algorithm we can compute some error measure on the training set, which we call the training error, which we aim to minimize. So fare, this description makes the task look like an optimization problem. The key difference that distinguishes the ML field from optimization is that we are also interested in the test error to be low as well.

We define the generalization error as the expected value of the error on a new input [27]. We take the expectation across different possible inputs from the distribution of inputs that we expect the system will encounter in practice. We estimate the generalization error of a model by measuring its performance on a test set of examples separate from the training set.

It is important to specify that the test, training, and validation sets can be separated only as long as each set is independent from the others, that the sets are identically distributed and drawn from the same probability distribution as each other. Only if these conditions are met

can we describe the data-generating process with a probability distribution over a single example.

During the training process, the training set is used to select the value of the ML algorithm parameters to reduce the training set error. Then the test set error is calculated. The expected test set error is greater than or equal to the training set error. The factors determining how well a ML algorithm will perform are the ability of the ML algorithm to make the training error small and to make the gap between the training and test error small. These challenges correspond to the underfitting and overfitting factors discussed in section 2.2.2.

A Typical ML algorithm has several settings that serve to control its behaviour. These settings are called hyperparameters. These hyperparameters are typically not adapted by the learning algorithm itself, even if it certainly would be possible to design a nested learning procedure that finds the best hyperparameters. Taking a polynomial regression as an example, one hyperparameter could be the degree of polynomial. A setting can be chosen to be a hyperparameter, and thus have the learning algorithm not learn from it, because it is difficult to optimize. More often, a hyperparameter is designated as such because it would not be appropriate for the ML algorithm to learn that hyperparameter from the training set. This is relevant for all hyperparameters that dictate the capacity of the model. When trained on the training set, these hyperparameters would invariably opt for the highest model capacity, leading to overfitting.

This is where the validation set becomes relevant. The ML algorithm does not observe this set. Therefore, it can be used to estimate the generalization error during or after the training. Using this data, the hyperparameters will be updated accordingly. As the validation set is used to optimize the hyperparameters, the validation set will underestimate the generalization error, by a typically smaller amount than the training set error.

2.1.4.1. CROSS-VALIDATION

Using a fixed division of the dataset into training and test sets can pose issues, especially if the test set ends up being small. A limited test set can lead to statistical uncertainly in the average test error estimation, making it more difficult to confidently assert that algorithm A outperforms algorithm B for a specific task. This isn't a significant concern for large datasets. For smaller datasets, there are alternative methods that allow for the utilization of all examples to estimate the average test error, though they come with a higher computational cost. These methods revolve around repeatedly training and testing on various subsets or divisions of the original dataset. The most prevalent method is the k-fold cross-validation. The dataset is divided into k distinct subsets. The test error is then approximated by averaging the test errors over the k trials. In each trial i, the i-th data subset is used for testing, while the remaining data serves as the training set. A challenge with this approach is the lack of unbiased estimators for the variance of such average error estimates [28], but approximations are generally employed.

2.1.5. DEPLOYMENT

The model and its results need to be made accessible to the customer. Otherwise, it has no use. This step can vary in complexity from one project to another and should as such not be underestimated. The deployment is documented and planned, so does the monitoring and the maintenance. Depending on the business context, the data could change over time. In this case, the existing model needs to be re-trained using updated data. Therefore, the model must be monitored at recurring intervals.

The project must also be documented. This can include a final representation of the data mining results. Finally, the project is reviewed. This includes notably a list of what went well and what could have been improved. A plan on how to improve in the future is outlined here.

2.2. THE IMPORTANCE OF DATA PRE-PROCESSING

Data pre-processing represents the most important step according to the DSM [25]. Indeed, pre-processing can be used to transform data to be compatible with the requirements of machine learning algorithms on their inputs (*e.g.*, ID3 accepts only numerical inputs) and, sometimes, to simply transform one representation into another (*e.g.*, an image into a feature matrix) [29]. Finally, and probably most importantly, data pre-processing has a significant impact on improving the performance (also generalization) of machine learning models [30]. It is the latter that interests us and that was topic of the pilot project with CGIC.

The time spent pre-processing data constitutes about 50% to 80% of the total time of a machine learning project [2]. This may be surprising given the existing frameworks (*e.g.*, Weka, RapidMiner, Knime, SAS, and scikit-learn) to facilitate this step [10-14]. Nevertheless, the number of data pre-processing operations is substantial (*e.g*. discretization, normalization, and standardization) and only the experts know exactly which operations to apply. Not to mention the different algorithms allowing to perform the same or similar operation. On the other hand, experts spend a lot of time to find one of the optimal combinations of pre-processing operations to obtain satisfactory performance and then send the trained machine learning model to production. Therefore, it is essential to focus on this step, which is key to the success of a data science project.

The characteristics of a dataset can have an impact on how the learning algorithm is going to interpret the training data it would learn from. Data pre-processing can have some influence on under-fitting or over-fitting the model to the data.

2.2.1. UNDER-FITTING

Under-fitting describes a situation where the learning model is unable to properly recognise the underlying logic of the data. It is therefore unable to provide correct answers.

Figure 3 illustrates the difference between the true function and the model function predicted using the observations when under-fitting takes place.

2.2.2. OVER-FITTING

Over-fitting is what happens when a model interprets each data point to model itself around the points completely. At first glance, it may seem like a desirable outcome, but the model fits itself too closely to random noise, which would not be the case with a properly trained model. Figure 4**Erreur ! Source du renvoi introuvable.** illustrates the difference between the true function and the model function predicted using the observations when over-fitting takes place.

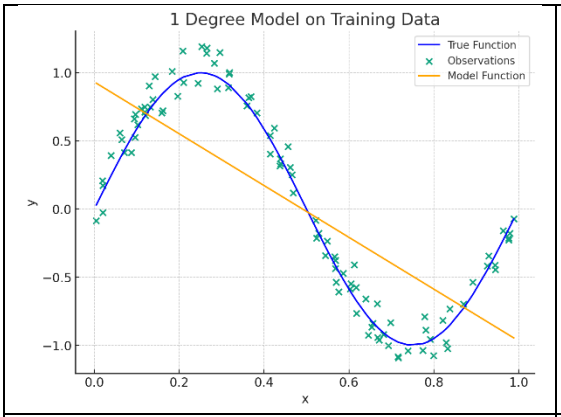

Figure 3 : Under-fitted model [31] © Hans Darmstadt-Bélanger

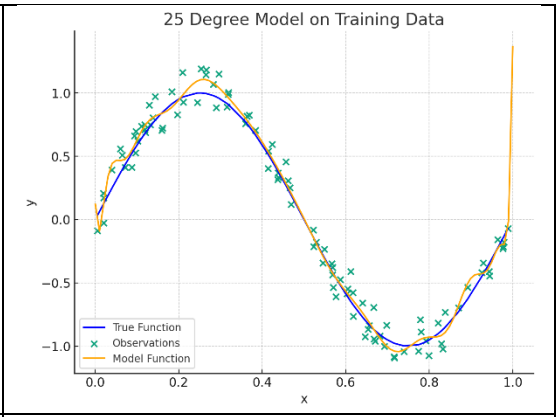

Figure 4 : Over-fitted model [31] © Hans Darmstadt-Bélanger

2.3. CONFUSION MATRIX

The predictions being generated by the model can be compiled using a Confusion Matrix. This helps to compile the types of responses being given and could be used to better tweak the model after a round of testing.

A confusion matrix is an n x n table, where 'n' represents the number of distinct classes. It details the number of correct and incorrect predictions for each class by comparing the predicted class labels with the true class labels. A positive response represents an observation

belonging to the class one is trying to predict whereas a negative response is an observation not belonging to the target class. An illustration of a confusion matrix is presented in Table 1.

Table 1: Confusion Matrix

Total Population		True Condition	
		Positive	Negative
Predicted Condition	Positive	True Positive (tp)	False Positive (fp)
	Negative	False Negative (fn)	True Negative (tn)

The confusion matrix thus consists of four values. The true positives, the false positives, the false negatives, and finally the true negatives:

- True positive: a correctly predicted positive observation,
- False positive: an incorrectly predicted positive observation,
- False negative: an incorrectly predicted negative observation,
- True negative: a correctly predicted negative observation.

2.3.1. ACCURACY

One of the most intuitive metrics that can be derived from a confusion matrix is the accuracy of the model. We can determine the model's accuracy by calculating the ratio of correct predictions against the total sum of predictions (Eq. 1).

$$\text{accuracy} = \frac{tp+tn}{tp+fp+tn+fn} \quad (1)$$

A dataset is considered unbalanced when the distribution of the dataset is unequal among the various classes. The accuracy of the classification algorithm usually suffers as a result of the unbalanced characteristic of the dataset.

In this context, the precision and recall metrics can become useful parameters. Precision quantifies the likelihood of a positive test result (Eq. 2). Recall quantifies the number of true positives found in the model (Eq. 3).

$$\text{precision} = \frac{tp}{tp+fp} \tag{2}$$

$$\text{recall} = \frac{tp}{tp+fn} \tag{3}$$

2.4. QUANTIFYING DATA QUALITY

Data quality is a seemingly vague concept that pertains to every aspect and characteristic of data. It has been assessed multiple times throughout the present Master thesis that if the data is not properly transformed, a learning algorithm can have difficulties to derive any meaningful information out of it. The various causes for these difficulties potentially experienced by the learning algorithms are discussed next.

[32] categorises each discrete issue in three classes of issues: accuracy, relevance and provenance. This categorisation is depicted in Figure 5.

2.4.1. ACCURACY

The concept of accuracy as defined in the Merriam-Webster dictionary is "*freedom from mistake or error*" [33]. In the context of big data, the concept of accuracy refers to the correctness of the data and the absence of noise [32]. It is essential to have accurate data as practitioners rely on data to build regression and prediction models to improve the practice of software engineering. Should there be underlying quality problems with the data used to train a model, it cannot be expected to provide outcomes that meet the practitioners' expectations.

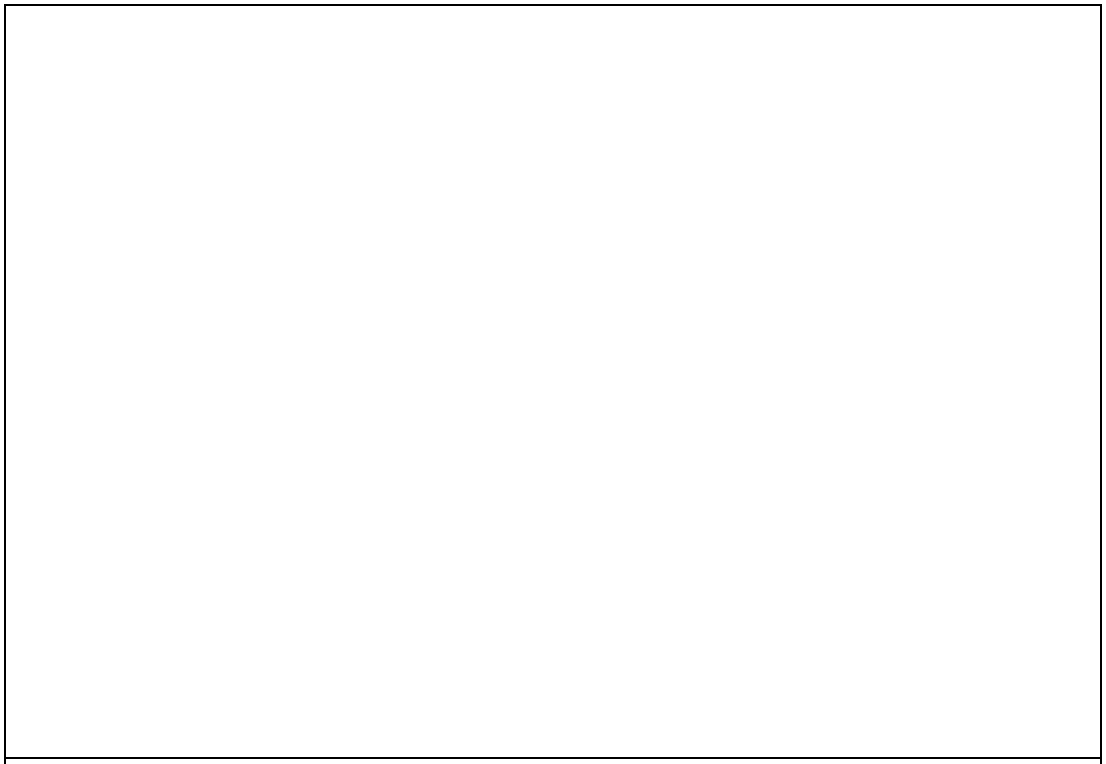

Figure 5 : Taxonomy of Data Quality. Adapted from [32] © Hans Darmstadt-Bélanger

2.4.1.1. OUTLIERS

Outliers can be defined as discordant observations for contaminants which are noted by the data analyst due to their difference to the rest of the dataset [34]. The presence of an outlier may be a symptom for some sort of problem. It is important to exercise caution when determining if an outlier is a problem or not. In a real estate dataset, the outliers are multi-million-dollar houses and these values are legitimate. On the other hand, if a person is supposedly 6m tall, the analyst can reasonably conclude that the respondent expressed the size in feet rather than meters when filling in a hypothetical survey. One possible way of identifying outliers would be to discover reliable and frequent patterns that reflect the typical characteristics of software project data. When the pattern does not match the data, an outlier value has been found. One drawback of using pattern-based outlier detection is that some heterogeneous datasets could mask the patterns relevant to subsets of the data. Furthermore, we do not currently have underlying theories that tell us what patterns to expect. Another common method of identifying outliers is the use of statistical analysis [35].

2.4.1.2. NOISE

Noise refers to erroneous data. Instances of noisy data should be removed as their removal has been shown to improve model's performances [36-38]. To remove the noisy instances, they must first be identified and to do so, various noise detection techniques are at the analyst's disposal such as Bayesian multiple imputation, Ensemble-Partition filter, and rule-based noise detection.

Bayesian multiple imputation can be summarily described as a clustering-based noise detection approach using the k-means algorithm.[32]

Ensemble-Partition filter refers to a technique to detect noise relative to an attribute of interest (AOI) [32].

The rule-based technique consists of expressing constraints that indicate an erroneous data if they are violated.[39] Easy examples of impossible data are negative values for a weight measurement, or an end date recorded has having happened prior to the start date of the same event.

2.4.1.3. INCONSISTENCY

Inconsistency can be defined as a lack of harmony between different parts or elements; instances that are self-contradictory or lacking in agreement when it is expected.

A lack of inconsistency or simply consistency in the data labels and in the recorded data is one of the essential factors in achieving good quality data. Analysts should make sure that the columns and the values in their datasets are properly explained and should put in place mechanisms to resolve problems associated with the recording and interpretation of data.

2.4.1.4. INCOMPLETENESS

Most real-world datasets will have some missing values to them. This is widely recognised by the data science community [40-43]. Data is considered as missing if it is not to be found either because a value is present but not where it is expected or not present or included when it is expected. Incompleteness as a notion is broader and can also refer to incomplete finished and imperfect data. Incompleteness can mean to lack a part or parts; it describes something that is not whole or not full. Incompleteness is of course relevant since smaller datasets could mean a model is not statistically significant.

Once it has been established that data is missing, the analyst must take steps to deal with the problem. These steps may include imputation, a procedure that "fills-in" the missing values with one or more plausible values. One of the simplest ways of imputing missing data is to use either the mean or median values of the column but more comprehensive imputing techniques also exist. Bayesian multiple imputation, k-nearest neighbour imputation and Multinomial logistic regression (MLR) are such examples. Khoshgoftaar and Van Hulse [44] have demonstrated that Bayesian multiple imputation is an effective imputation technique.

Other times, placeholder values like ‘Unknown’, ‘NA’ or 0 can be imputed in the missing fields. This should be done when the absence of values provides useful information. The analyst should always keep in mind that this kind of imputation is at a greater risk of skewing data distributions.

It is also important to note that data imputation carries its risks, such that it may be better to not impute at all, given the context. Since the values of the missing data is unknown, it is impossible to know its distribution. Imputing may therefore skew the data which may lead to misleading results. This is known as the “Missing Not At Random” problem [45].

Imputation is a useful tool in face of the problem of missing values when it is not rampant. The data analyst must however ensure that they are using the correct imputation technique to respond adequately to the kind of missing data they may encounter. Numerical or categorical

values, class or attribute missing values, single, or multiple missing attributes may require a different way of being imputed to optimal results.

2.4.1.5. REDUNDANCY

Redundant and duplicate data in datasets will over-represent certain classes which could confuse the learning algorithm into misleading results and can adversely impact the performance of classifiers. Redundant data also has the impact of slowing down the building of classification models due to the additional processing needed to parse and consider the redundant values.

Once the redundant observations have been identified, it is standard procedure to keep only one instance of the redundant observations.

2.4.2. RELEVANCE

The concept of relevance in the context of data quality, particularly in data science, is essential for ensuring the effectiveness and accuracy of models. As highlighted by [46], relevance is identified as one of the most significant data quality dimensions, encompassing the appropriateness and pertinence of data to the specific task or project at hand. For instance, in machine learning (ML) projects, the use of relevant data is crucial. A clear example of this is that data pertaining to farming crop yields would be irrelevant for predicting the output of a car factory. The relevance of data in such scenarios is determined by the strength of the correlation between the phenomena observed in the data and the phenomena being predicted.

One method to quantify this correlation is by calculating the regression coefficient [47]. Another is by utilizing the Pearson correlation coefficient, which measures the linear relationship between two variables [48]

2.4.2.1. AMOUNT OF DATA

Small datasets are a known issue in data science since they tend to not lend themselves to the generalization of results. Another constraint they bring is with the list of suitable analysis techniques. Some approaches are based on the assumption that a certain volume of data will be available[49]. Furthermore, the act of pre-processing, particularly during feature selection, may further reduce the data used for modeling compared to what was initially available. Although a dataset might start off sufficiently large, the application of feature set selection strategies could result in a subset too small for effective significance testing. This form of statistical testing, crucial for determining whether observed results are due to genuine effects or random chance, often relies on hypothesis testing and p-value calculations. However, small datasets may lack the statistical power needed for such tests, potentially leading to unreliable or inconclusive outcomes[50]. It's vital for analysts to be cognizant that pre-processing should not compromise the validity of generalizations drawn from the data, which can occur if the dataset becomes too small or if the modeling methods are not suited to the size of the dataset.

2.4.2.2. HETEROGENEITY

We can ask ourselves if datasets coming from a single organization, when put together, are superior to datasets coming from various organizations. Research indicates that utilizing datasets from a variety of organizations, as opposed to relying solely on data from a single source, can lead to improved outcomes. This is primarily due to enhanced data comparability and reduced heterogeneity achieved through innovative data models, which facilitate more effective aggregation and analysis of diverse datasets [51, 52]. The use of relevancy filtering in generating estimates based on data from another project was analysed in a study focused on the timing of utilizing data from different projects for effort estimation. The research suggested that employing cross-organization data, with the implementation of a relevancy filter,

could lead to estimation accuracies comparable to those achieved with data from the same organization [53].

It should be noted that the distinction between single-company and multi-organization could be an oversimplification as some single companies manage hugely diverse projects.

2.4.2.3. TIMELINESS

Timeliness refers to how current the data is. This property of a given observation is linked to the concept of relevance although the attention it has received in research literature is limited. Datasets and their characteristics should be reviewed constantly to check for any changes in context and operation that could reduce the relevance of the data to contemporary settings. Using an older dataset is not necessarily a mistake on the analyst's part but questions about the appropriateness of using this data in the context of present-day needs could be raised. It is therefore suggested to prioritize data that has been collected more recently to build models for current use, barring any other issues.

2.4.3. PROVENANCE

Provenance is defined in the Merriam-Webster Dictionary as "*the history of ownership of a valued object or work of art or literature"* or as *"origin, source*" [54]. Being able to trace back to the origin of a data point allows the analyst to audit the source to ensure its reliability. This would determine how much trust the analyst would place on the data and on the results yielded from it. Therefore, a scientific result would be considered to be as important as its provenance.

2.4.3.1. COMMERCIAL SENSITIVITY

The commercial sensitivity of the data is a common constraint on the provenance ML. When a company has data that gives it a competitive advantage, it is less likely this company releases the data to independent researchers to maintain its competitive advantage. The

company could also prevent data from being released if they think this data could be utilised to shine an unfavourable light on it. Researcher's access to data is often contingent on signing non-disclosure agreement [40]. Being barred from releasing the data used to obtain the results renders these studies non-replicable.

2.4.3.2. ACCESSIBILITY

As previously mentioned, companies may be reluctant to release their data due to its commercial sensitivity or due to data privacy concerns for example. Before releasing the data, the company may be compelled to highly sanitize it. This will typically mean sensitive data is redacted, which could have an impact on the quality of the model. The alternative to degraded proprietary data is open-source datasets but they may be of a second rate, depending on the research context. For this reason, quality contributions to publicly accessible repositories like those provided by the International Software Benchmarking Standards Group [55] and PROMISE [56] with rigorous provenance information should be encouraged.

2.4.3.3. TRUSTWORTHINESS

As the analyst is usually not the one in charge of generating or gathering the data, they therefore usually rely on the people and systems used to collect and verify that data. This point ties in somewhat to the provenance as knowing the provenance of the data gives an insight into its trustworthiness. Provenance systems that could provide analysts with insightful knowledge about the origins of the data should be adopted when possible. These systems can enable data providers and users to also track the changes the data has undergone, such as anonymization of sensitive information. These metadata can help ensure that models are built with integrity.

2.5. THE STEPS OF DATA PRE-PROCESSING

As described in the previous chapter, the data preparation process can be summarised as being comprised of three big steps: gathering the data, cleaning up the data, and optimising the data. Each step can be broken down in multiple sub steps, which are discussed next (Figure 6).

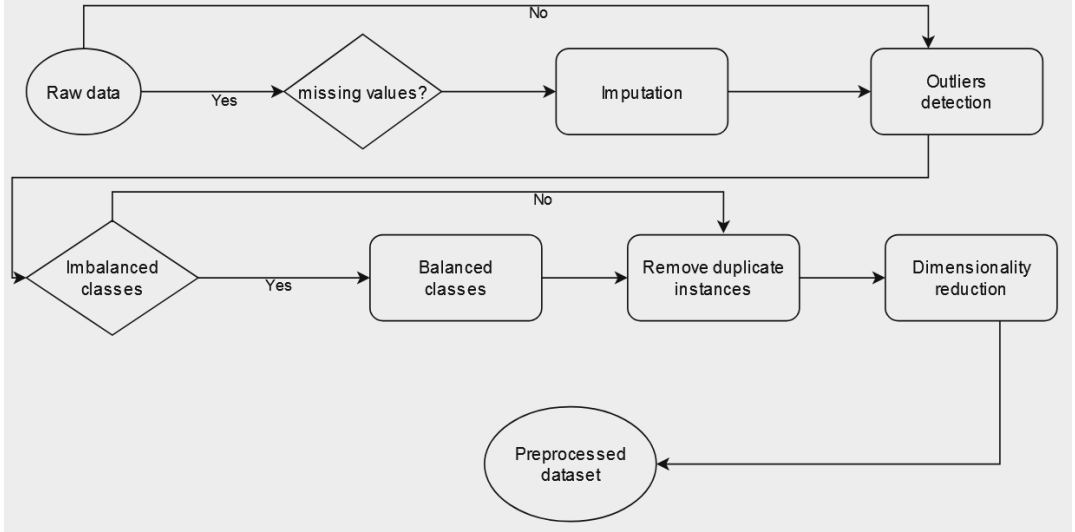

Figure 6 : Process to clean up the data [3]. © Hans Darmstadt-Bélanger

2.5.1. GATHERING THE DATA

Gathering the data seems relatively simple and indeed it could be considered as one of the lighter steps of the pre-processing process. It aims to determine the availability of reliable data pertinent to the subject at hand. The sources for said data should be assessed by looking into the entity that provided the data and the main theme of the dataset.

Basic metadata can also be used to analyse the dataset. A non-exhaustive list of such metafeatures is listed below:

- the size of the dataset,
- the number of attributes,
- the number of instances,

- the time window contained in the data.

To help with choosing which datasets will be selected for use in the project, these meta-features can be visualised in comparative charts. A list of relevant meta-features can be found in [57].

Should there not be enough data or should the data lack the needed information at the end of this phase, the analyst would need to go through the data gathering step again.

These brief overviews of the data to be used allows the analyst to identify the various ways in which the data will need to be cleaned up and which steps to take in doing so.

TABLE 1: List of relevant meta-features identified in [57].

Meta-features (Dataset characteristics).

No	Name	Type	Modifiable
1..2	[Number|Percentage] of Continuous Attributes	Continuous	Yes
3..6	Min[Means|Std|Kurtosis|Skewness] of Cont. Att.	Continuous	Yes
7..10	Mean[Means|Std|Kurtosis|Skewness] of Cont. Att.	Continuous	Yes
11..14	Max[Means|Std|Kurtosis|Skewness] of Cont. Att.	Continuous	Yes
15..17	Quartile [1|2|3] of Means of Continuous Attributes	Continuous	Yes
18..20	Quartile [1|2|3] of Std of Continuous Attributes	Continuous	Yes
21..23	Quartile [1|2|3] of Kurtosis of Cont. Att.	Continuous	Yes
24..26	Quartile [1|2|3] of Skewness of Cont. Att.	Continuous	Yes
27	Number of Categorical Attributes	Categorical	Yes
28	Number of Binary Attributes	Categorical	Yes
29	Percentage of Categorical Attributes	Categorical	Yes
30	Percentage of Binary Attributes	Categorical	Yes
31..33	[Min|Mean|Max] Attribute Entropy	Categorical	Yes
34..36	Quartile [1|2|3] Attribute Entropy	Categorical	Yes
37..39	[Min|Mean|Max] Mutual Information	Categorical	Yes
40..42	Quartile [1|2|3] Mutual Information	Categorical	Yes
43	Equivalent Number of Attributes	Categorical	Yes
44	Noise to Signal Ratio	Categorical	Yes
45..48	[Min|Mean|Max|Std] Attribute Distinct Values	Categorical	Yes
49	Number of Instances	Generic	Yes
50	Number of Attributes	Generic	Yes
51	Dimensionality	Generic	Yes
52,53	[Number|Percentage] of Missing Values	Generic	Yes
54,55	[Number|Percentage] of Instances with Miss. Vals.	Generic	Yes
56	Number of Classes	Generic	No
57	Class Entropy	Generic	No
58,59	[Minority|Majority] Class Size	Generic	No
60,61	[Minority|Majority] Class Percentage	Generic	No

2.5.2. CLEANING UP THE DATA

Here, the problems identified at the end of the previous step is solved. The step by step process of cleaning up data as suggested in [3] can be seen in Figure 6 in section 2.5. Each step is described next.

2.5.3. CHECKING FOR MISSING VALUES

Data can be missing from a dataset for various reasons. A broken sensor can be such a reason. When the absence of this information is considered problematic, steps must be taken to mitigate these issues. Imputing the missing data by inserting the average value, the median value or a constant value is one way of managing missing data. Another way would be to ignore the rows containing missing data altogether.

These approaches may not be sufficient for more complex datasets where the pattern of missing data can induce bias. More advanced techniques like K-nearest neighbours (KNN) and Multiple Imputation by Chained Equations (MICE) offer more sophisticated solutions [58, 59].

The KNN imputation technique relies on the similarity between data points to predict missing values, which aims to preserve the underlying structure of the data.

MICE extends imputation by iteratively generating multiple complete datasets from missing values Initially, it employs simple methods like column median for the first round of imputation. Then MICE estimate a distribution based on the observed data points and imputes the missing data based on the distribution. In short, MICE builds a ML model to impute the missing data and uses the output of this first model to train the model for the next iteration which should give the next iteration an imputation closer to what the value would have been, had it not been missing.

2.5.4. DETECTING OUTLIERS

If a data point is inconsistent with the rest of the dataset, it can be atypical or an outlier. The potential causes for outlier data are similar to the causes for missing values. A faulty sensor could hypothetically produce irregular data instead of no data at all, or some irregularity could happen during a data transfer operation, among other possible causes for this kind of issue. There are three main ways of approaching outlier data.

The first approach involves retaining outliers in the dataset, particularly when these instances represent crucial, rare phenomena important for specific studies such as fraud detection or rare disease tracking. It's essential to consider the potential loss of statistical power and significance that could arise from removing these outliers, emphasizing the need for a nuanced evaluation of their impact on the dataset's overall integrity and the study's findings.

The second approach, the removal of outliers, requires detailed criteria for their identification and exclusion, such as employing z-scores[60] or the interquartile range (IQR) method [61]. To mitigate potential biases, a sensitivity analysis should be conducted to evaluate the impact of outlier removal on research outcomes. This process must be approached with careful consideration, ensuring that the data manipulation does not inadvertently skew the results. This approach is easy to justify as they are extreme observations, but these outliers could be both unusual and valid observations. This approach therefore is not ideal but remains the predominant practice among practitioners.

The third approach consists of using robust algorithms that are resistant to outliers such as least-median squares regression [62] and Bayesian networks for their resistance to outliers. They have been employed to mitigate the drawbacks described in the first two approaches[63].

2.5.5. LABEL CORRECTION

In a similar vain to outlier data or missing values, the label correction consists in detecting contradictory instances in the data such as duplicate samples with different labels. This issue

poses a significant challenge for machine learning models, as inaccurate labels have the potential to impact their performance.

When confronted with the task of label correction, the data analyst has to sift through the dataset to identify labelling inconsistencies. Once identified, these errors must be corrected either by assigning the correct label based on the data analyst's expertise or through automated methods that can detect and resolve inconsistencies.

2.5.6. CLASS BALANCING

Imbalanced class distributions within a dataset can significantly diminish the performance and accuracy of learning algorithms.

Learning from imbalanced datasets is often reported as being a difficult task [64]. This issue is particularly pronounced in situations where the disparity between majority and minority classes is vast, often complicating the learning process. For instance, in medical diagnostics, an algorithm trained on data where instances of a rare disease (minority class) are vastly outnumbered by healthy cases (majority class) faces difficulty. This imbalance can lead algorithms like the k-Nearest Neighbour (k-NN) to inaccurately predict rare disease cases as healthy due to the overwhelming presence of healthy instances in the dataset. Especially in a 1-NN scenario, the nearest neighbour to a rare disease instance is likely a healthy case, leading to a high misclassification rate of the critical minority class, an outcome that is far from desirable [65].

To address this imbalance, the analyst usually employs a resampling strategy to make the various classes less imbalanced. Figure 7 illustrate the impact of resampling on a fictitious medical dataset. Resampling can be split into two subcategories: down-sampling and up-sampling.

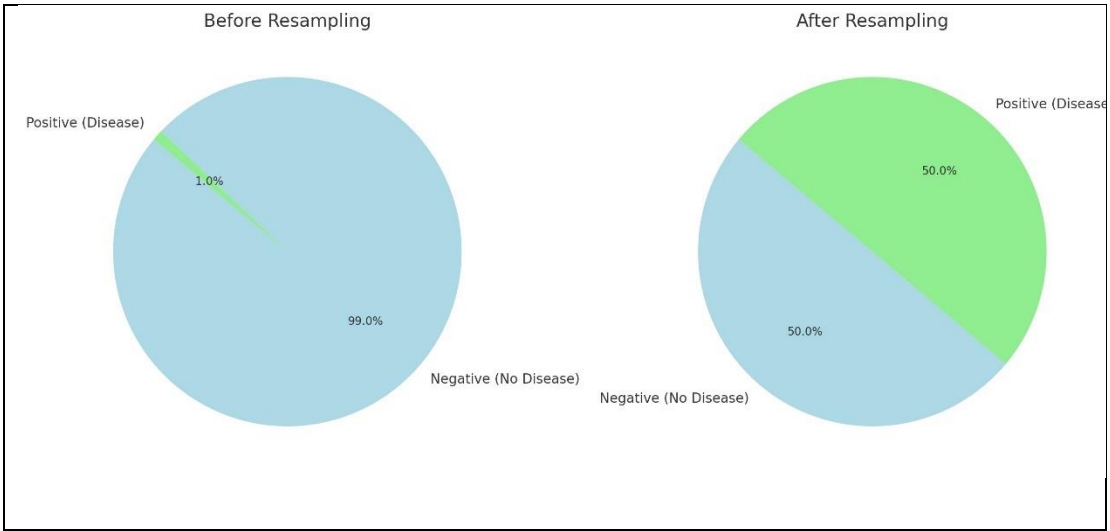

Figure 7 : The impact of resampling on class distribution on a fictitious medical dataset © Hans Darmstadt-Bélanger

-
- Down-sampling involves selectively reducing the number of instances from the majority class to create a more balanced dataset. This process is executed without replacement, meaning once an instance is selected for removal, it is not returned to the pool of potential selections. The goal is to decrease the size of the majority class to a level that is more comparable with the size of the minority class. This technique can be particularly useful when the dataset is large, and removing instances does not significantly compromise the dataset's integrity or the machine learning model's ability to learn the underlying patterns.
-

Up-sampling involves increasing the number of instances in the minority class to equalize the distribution between the classes in a dataset. This approach seeks to augment the minority class by creating additional instances, thereby providing the machine learning model with more data to learn the characteristics of the minority class.

2.5.7. REMOVAL OF DUPLICATE INSTANCES

Redundant instances or redundant attributes can be present in the dataset for various reason. One such example could be the same column being present under different names in two datasets that have been merged together in a previous step of the pre-processing process. These duplicates must be identified and deleted from the dataset to prevent their presences of negatively impacting the performance of the models.

Resulting from this clean up phase are pre-processed datasets and meta-features allowing to quantify the level of cleanliness of the processed data by comparing the same metrics before the cleaning took place. This analysis allows for an iterative process of cleaning up the data. Should the data prove too difficult to adequately clean up, an analyst may find themselves compelled to go back to the data gathering step.

The main pre-processing operations are data transformations that can be applied locally (selected features) or globally (on the whole set of attributes of the dataset), and allowing to obtain mainly continuous or categorical/nominal variables [2]. This includes discretization (continuous to nominal), transformation of nominal variables to binary, normalization, standardization, missing value replacement, and dimensionality reductions. The latter allows to extract only salient information from the data and avoid overfitting of machine learning models [66]. This list is not exhaustive and other pre-processing operations exist (*e.g.* outlier detection).

2.5.8. CHANGING DATA REPRESENTATION

Unlike the steps illustrated in Figure 6 (section 2.5) that are relatively similar from one project to another, the exact steps and goals of this step will vary from one project or learning algorithm to another. Some algorithms may only work with data that is in a certain format, usually numbers or may provide optimal results when the data the algorithm learns from is in a format that is more conducive for the learning algorithm to derive useful knowledge from. In

the former case, the problem can be solved by encoding the data whereas dimensionality reduction would be the answer to the latter.

2.5.8.1. ONE HOT ENCODERS

In the first example where an algorithm takes only numerical values as input, every characteristic that is stored in a text format for example may need to be encoded in a numerical format. Let's suppose a dataset of cars covered under an insurance policy. These records will contain the car's color. One easy way of encoding the color to a number would be to assign each color an arbitrary number. Simply replace the text *yellow* by the number 1, *grey* by 2 and so on. One issue that may be encountered during the learning phase is that the learning algorithm assumes some ranking between the smaller and bigger numbers. It is reasonable to assume that a car with more kilometres on the odometer has more wear and tear than another car with fewer kilometres travelled. No such assumption can be made about the fact that orange is represented by a number twice as big as green. This is where a one hot encoder would come in.

A one hot encoder is a way of encoding label values to new columns with a binary value (recorded as 1 and 0 to keep the numerical format). Assuming that the exhaustive list of possible colors for a car are black, white and grey, the one hot encoder would replace the color column by a column for each possible value. In this case, 3 new columns would be added. The column representing the color of a given car will have a positive value in the field and the other colors which the car does not have will be marked by the negative value. This ensures that one possible value is not seen as inherently bigger or smaller than another one without losing any amount of data. One critical draw back of this method is the potentially sharp increase in size of the dataset due to the potentially large number of columns to be added. Keeping with the car themed examples, depending on the context, it could be viewed as reasonable to use a one hot encoder to encode a list of cars manufacturers but doing so with car models would not be advised.

2.5.8.2. DIMENSIONALITY REDUCTION

Another challenge data analysts can encounter is the one posed by high dimensionality data. As described in previous chapters, data is being generated at an ever-greater pace, and it is not just due to more observations being recorded. An ever-greater number of characteristics can be saved for each observation. Each dataset is different of course and so not every project has a high dimensionality problem, but some domains are more prone than others to encounter this kind of issue. One such field would be the biomedical filed since the advent of complete genome sequences. Subsequent developments lead to high-throughput techniques that are generate enormous amounts of data, resulting in massive growth of biological databases. It is not rare to see biological datasets with more variables than observations these days [67].

When faced with such high dimensionality datasets, statistical and machine learning algorithms are faced with a formidable problem. The only solution in these cases is to perform a dimensionality reduction technique. This approach aims to find a new system in which the large dimensionality data can be expressed with fewer variables without significant error or loss of information. This can be implemented on a global or local basis and can fulfil very different properties.

There are various dimensionality reduction algorithms, each with a different mathematical approach to the problem, but they all follow one of two fundamental approaches [68]:

Feature Selection selectively retains the most relevant variables from the original dataset, thereby simplifying the data without transforming the actual features. Techniques like Recursive Feature Elimination (RFE) work by recursively removing features, building a model using the remaining attributes, and calculating model accuracy to identify the most significant features [69].

Dimensionality Reduction exploits data redundancy to generate a new, smaller set of variables that are linear or nonlinear combinations of the original ones, aiming to preserve as much information as possible. Principal Component Analysis (PCA) is a prime example.

PCA identifies the directions (principal components) that maximize the variance in the data. By projecting the data onto these directions, it reduces dimensions while attempting to retain the dataset's essential characteristics [70].

It might be noteworthy that the output of a dimensionality reduction algorithm will be in a format not easily readable by a human trying to assess the quality of the resulting data.

2.6. SIMILAR RESEARCH

Recent research has tended to propose semi or fully automatic systems to assist the user in one, several or all of the data pre-processing steps. Nevertheless, the systems developed to provide this assistance in data pre-processing are based on three questions: What? How? Why? [57, 71].

2.6.1. WHAT

Under the "What?" question, we are interested in what tasks the system should support (*e.g.*, converting data formats to a single format, cleaning up outliers) and how the system should provide the assistance (automatically or interactively), etc. [57, 71].

2.6.2. HOW

For the second question "How?" we are looking for a way for the system to provide this assistance. In other words, we are looking for an answer to the question: how does the data flow between the different layers of the system (from input to output)?

2.6.3. WHY

Finally, the question "Why?" allows us to know the real intention of the system. For example: is it to impact data analysis (machine learning results) or not?

2.6.4. EXISTING SYSTEMS

Existing systems for data pre-processing (*e.g.* Wrangler, GDR, SST, KATARA, DATAXFormer, DPD, and PRESISTANT) [57, 72-77] have similarities. They allow, for the most part, to prepare data and to clean them. Unfortunately, few of them provide adequate support for managing categorical values, presenting a significant limitation when processing heterogeneous datasets. Categorical data, representing discrete categories without a natural order or scale, complicate operations like sorting and averaging, thereby challenging the systems' ability to handle diverse datasets effectively. Furthermore, none of the systems provide an automatic and interactive mode at the same time. In other words, it would be interesting to have both working at the same time in order to adequately guide users (experts and non-experts). It should also be mentioned that one of the steps that we never find in existing systems is data quality assurance (*e.g.* is the labelling of good quality?). This is an important point that we want to bring to our system. Finally, most systems are based on (*ad-hoc*) rule sets in order to perform all the sub-steps constituting the data pre-processing. Some systems are based on meta-learning which requires a benchmark of datasets that have already been pre-processed by experts, and others are based on the exploitation of optimization algorithms. We can say that no two current systems are alike. Indeed, each research (system) has been realized with different objectives (the questions What? How? Why?) and has been designed with the particular characteristics (*e.g.* image processing, homogeneous dataset) of the targeted domains (*e.g.* e-commerce, health). Therefore, research is essential in this project with CGIC. To further elaborate this point, some of the mentioned systems are described in greater details next.

2.6.4.1. PRESISTANT

The premise of the PRESISTANT tool [57] is that existing solutions either assume that users have expert knowledge, or they recommend pre-processing operators that are only ‘‘syntactically" applicable to a dataset. The problem here is that these solutions fail to consider their impact on the final analysis. Therefore, the researchers aim at aiding non-expert users by recommending data pre-processing operators that are ranked according to their impact on the final analysis.

As there are many available pre-processing operators, the non-expert user does not know which pre-processing options are potentially helpful to the task at hand. By developing a tool that recommends only a small set of beneficial transformations, the researchers aim at reducing the time consumed by pre-processing and at improving the result of the final analysis.

By training an algorithm to learn the impact of pre-processing operators, it can be then used to predict and ultimately rank different pre-processing operators. This is done using meta-learning techniques to develop a system that is capable of recommending pre-processing operators that positively impact the result of some classification tasks.

Sometimes, data must be transformed from one representation to another. An image may for example be transformed into a matrix representation to be suitable for exploration and analysis. These examples of mandatory transformations for an algorithm to function properly are not what PRESISTANT focuses on. Data may need to be transformed with the only goal of improving the performance of a machine learning algorithm [71]. These transformations are what PRESISTANT aims to automate. As these kinds of transformations are a choice rather than a necessity, due to the abundance of choice, finding the best choice can be time consuming.

To conduct an empirical analysis, 5 different classification algorithms (i.e., J48, Naive Bayes, PART[78], Logistic, and KNN) were used. Their performances on datasets before and after the transformations were applied were measured.

The PRESISTANT tool itself, uses a meta-learner to find the best preprocessing transformations to apply. A meta-learner is a method that aims at finding relationships between dataset characteristics and data mining algorithms. A predictive meta-model is trained on the relationship between the characteristics of a dataset and the impact a transformation has on a given data mining algorithm. Taking the example of a classification problem, the meta learner could be used to predict the predictive accuracy of the classification algorithm on a dataset and thus provide support to the user in the pre-processing step.

By using this approach, the tool can produce a list of potential transformations, ranked by their relevance to the analysis.

At first, a meta-learning space is established. The meta-learning space looks at the dataset's characteristics, called meta-data. Alongside the meta-data, the meta-learning space also has performance measures for data mining algorithms on the particular datasets.

Once that's done, the meta-learning phase generates a model used to define the area of competence of the data mining algorithm.

Lastly, when a transformed dataset is generated, the dataset's characteristics are extracted and fed into the predictive meta-model. That meta-model then predicts the performance of the algorithm on the transformed version of the dataset.

Once the classification algorithm is chosen by the user, PRESISTANT can assist the user by presenting them with a smaller number of relevant preprocessing options.

To do this, PRESISTANT applies rules to prune the irrelevant transformations such that the search space is reduced. A model is then trained to learn the impact of the transformations

on the performance of classification algorithms and finally, the trained model is used to rank the newly arriving versions of the datasets.

2.6.4.2. KATARA

Katara [74] is a tool that proposes, a knowledge base and crowd powered data cleaning system that, given a table, a knowledge base, and a crowd, interprets table semantics to align it with the knowledge base. It identifies correct and incorrect data and generates top-k possible repairs for incorrect data.

This project is based on the premise that a plethora of data cleaning approaches that are based on integrity constraints [79-83], statistics [84], or machine learning [85], have been proposed in the past. These previous projects have promising applicability and generality characteristics, but despite their best effort these alternatives, cannot ensure the accuracy of the repaired data. This is due to their very nature, as these methods do not have enough evidence to precisely identify and update errors.

There is an increasing availability of knowledge bases, both general purpose (*e.g.* Yago [86], DBpedia [87]) and special purpose such as RxNorm1. In parallel, crowdsourcing has been proven to be a viable and cost-effective alternative solution to using experts as they may be limited and expensive.

Knowledge bases tend to be incomplete in terms of the coverage of values in the table. This makes it hard to find correct table patterns and associate knowledge base values. Matching dirty tables to knowledge bases proves to be a challenging problem. The tables can lack reliable, comprehensible labels, thus requiring the matching to be executed on the data values. This kind of situation could lead to ambiguity as more than one mapping may be possible.

Therefore, human involvement will eventually be needed to validate matchings and to verify data when the knowledge bases do not have enough coverage. To effectively leverage the crowd, the traditional crowdsourcing issues must be addressed. For example, easy to

answer questions must be formed for the new data cleaning tasks. The order of issuing the questions has to be optimized to reduce monetary cost, among other challenges.

With the goal of combining knowledge bases and crowdsourcing techniques, to annotate and repair data, Katara brings the following contributions:

- Katara introduces a new class of table patterns to leverage table semantics using knowledge bases. This is done by modelling each table pattern as a direct graph where a node represents a type of column and the direction represents a binary relationship between the connected columns. A rank-join based algorithm allows to efficiently discover table patterns with high scores.
- Katara has crowdsourcing powered table pattern validation algorithm. This feature minimizes the number of questions using an entropy-based scheduling algorithm in order to reduce the uncertainty of candidate table patterns as much as possible.
- Lastly, given an established table pattern, it will be used to annotate data with different categories:
  - Correct data validated by the knowledge base,
  - Correct data validated by both the knowledge base and the crowd,
  - Erroneous data identified by both the knowledge base and the crowd.

In case of erroneous data, the top-k possible repairs for the identified errors are generated by another algorithm.

2.6.4.3. DATAXFORMER

The DataXFormer tool [88] is aiming to help users save time when transforming data, but in a different way than what has been discussed so far. Conversions such as litres to gallons, can be easily performed by applying a formula or a program on the input values. Other conversions are not always this easy to apply, such as zip code to city. This and similar conversions will require sifting through a repository containing explicit value mappings. Existing

systems already provide formulae and algorithms for these kinds of transformations. The major unresolved challenge is the automated identification of reference datasets to support value mapping. Fortunately, the Web hosts millions of tables with many containing explicit value mappings, in addition to value mappings hidden behind Web forms. The subject of this tool, DataXFormer, is a transformation engine that leverages Web tables and Webforms to perform this type of transformation tasks. The research project describes an inductive, filter-refine approach for identifying explicit transformations in a corpus of Web tables and an approach to dynamically retrieve and wrap Web forms.

A complex organization needs to integrate various heterogeneous data sources. Often, the same or highly related information can be expressed in different ways. Before tasks such as schema integration or record linkage can be performed, the data must be brought to a uniform representation. This is where data transformation tools that provide mappings across these pieces of information become necessary. Examples can include the need to map stock symbols to company names, to change date formats from MM-DD to DD-MM, or to replace cities by their countries.

As semantic transformations cannot be computed by solely looking at the input data, the use of a mapping table containing the explicit outputs for possible inputs is needed. While it is doable to collect reference data for a specific transformation task, the process cannot be done at scale.

One potential solution to this issue is to crawl the web to find relevant mapping tables. One naïve way of approaching the problem would be to test every web form and web table for coverage, and then use the resource to fill in as many missing values as possible. When multiple resources cover a transformation and provide conflicting values, a resolution scheme would be required to choose the best result. This approach is obviously linear in the number of web forms and web tables. Even if it is more efficient than manually collecting the relevant data, this method still does not scale well.

With these issues laid out, a scalable solution would require the following components:

- The ability to index a large corpus of web tables to support efficient retrieval of Web resources that cover a given transformation task;
- A platform that judiciously uses the two types of resources (tables and forms) given their very different access and response time characteristics;
- The ability to effectively involving a crowd of experts (expert sourcing) to guide the transformation tasks, where necessary, and consolidating the possibly inconsistent mapping results from the relevant transformation resources.

To solve the described problem, DataXFormer proposes two main contributions.

- The project features an inductive filter-refine approach to index and retrieve a large corpus of web tables to efficiently answer transformation queries.
- The researchers demonstrate how the relevant Web forms can be retrieved from the web and wrapped using the input transformation examples and human experts.

DataXFormer's architecture is built on two complimentary transformation engines. The first one is based on locally stored static Web tables whereas the second is based on dynamically discovered Web forms.

When a user enters a query into DataXFormer, the query gets converted into an internal form for each of the retrieval components. These retrieval components return candidate Web tables and Web forms. The table retrieval component works on top of a local repository, unlike the form retrieval component which uses the entire web to find relevant forms.

The table retrieval algorithm gets a set of relevant candidate tables for the current user query using an inverted index. The candidate results will be further analysed by the refinement component. This refinement component verifies the coverage of the candidate tables with respect to the query examples. Should the coverage be above the user-defined threshold, DataXFormer will extract the rest of the required transformation values. Then dealing with Web forms, DataXFormer first uses a Web search engine to retrieve relevant URLs.

When the results of the web search are examined, candidate Web forms can be identified. Each candidate form has an associated “wrapper” generated that allows DataXFormer to query the form and to obtain the transformation values. Candidate Web forms are then queried using the examples provided in the user query. The candidate Web forms with sufficient coverage will then be invoked using the remaining input values. As a last step, the solution integration component consolidates and ranks the various transformation solutions for a given query to obtain the two subsystems to output the desired transformations. When an automatic reconciliation of the values can’t be performed, the expert sourcing is invoked.

Collectively, these research projects represent significant steps toward bridging the gap between the intricate demands of data pre-processing and the practical capabilities of users without extensive programming expertise. However, despite their advancements, fully user-independent pre-processing tools remains a work in progress. Further research and development is still needed to overcome existing limitations and to fully harness the potential of machine learning for non expert users.

There remains an area which none of these tools aims to cover. By focusing on a holistic user experience that minimizes technical barriers to entry this project aims to bridge the gap between advanced data pre-processing capabilities and their pragmatic application by a broader range of practicians. The detailed implementation of which is discussed next.

CHAPTER 3

PRE-PROCESSING PROJECT

The business need expressed by CGIC at the beginning of the present project was to allow non-expert users to perform a large amount of small-scale pre-processing tasks for different projects in the company. One typical profile was the actuarial professional who is trained to understand and to build models in a way not too different from how a data analyst would have been taught these concepts. The key difference being that the actuarial professional does not have the programming proficiency to write a script to pre-process the data before using it to build a model. Having to pre-process the data themselves nonetheless, actuarial professionals require a disproportionate amount of time to complete a model. This is mainly due to the considerable time required for pre-processing. Each project within the company having unique needs, the suggested pre-processing project would need to be flexible and adaptable. At the same time, it should seamlessly integrate into the existing infrastructure and technological environment of the company to minimize friction upon implementation.

3.1. CGIC'S BUSINESS NEED

According to the business need, the suggested program needed to be intuitive and simple to learn, while retaining powerful and flexible capabilities that are required for a complete pre-processing tool. Due to the large number of employees working on different projects using different technologies in different parts of the company, the resulting solution had to be as flexible and technology agnostic as possible. Considering these constraints, it was decided to build a modular extract, transform and load (ETL) platform specialised in pre-processing transformations. ETL is the process of extracting data from multiple sources and transforming the various data into a unified format before loading the data in a data warehouse or another unified data repository.

3.1.1. EXTRACT, TRANSFORM AND LOAD SOFTWARE

Our program is an ETL program and as such we aim to provide codeless platform to execute pre-processing tasks on large datasets without requiring the user to know anything about computer programming.

The typical use case for an ETL program is to import various data with various formats from outside sources, to transform them to normalize them to the company needs, and then to load them in the company's data warehouse. Therefore, ETL programs usually feature transformations meant to standardize the data's format in order to import them. Our transformations are more advanced and specialized to facilitate common data pre-processing tasks such as imputing, normalizing, and encoding.

The final objective of this project was to democratize access to data pre-processing. An application example for the software would be an actuary who needs to pre-process data in order to build a model but who is not too well versed in programming. Current CGIC employees usually manage to get such a task done but due to their lack of expertise this step of a project can take about two weeks we aim to compress that step down to just a few minutes. The different steps of the ETL process are discussed in the following sections.

3.1.1.1. EXTRACT

As a generic company will typically have data that comes from various places such as structured query language (SQL) servers, customer relationship management (CRM) systems, Emails, Web pages, and social media analytics. This raw data need to undergo a data processing process in what is called a staging area.

3.1.1.2. TRANSFORM

In the transform step, the data is transformed and consolidated for analytical use. Some typical tasks at this stage are the filtering, de-duplicating, validating, and authenticating of data. Calculations, translations, and summarizations based on the raw data can be performed. For example, column names can be changed for consistency purposes or converting units of measurements and currencies. Audits for quality purposes and compliance can be conducted. The data can also be transformed due to regulatory frameworks. This can include removing sensitive information or encrypting it.

3.1.1.3. LOAD

Finally, the data is moved from the staging area into a data warehouse. In practice, all the data gets initially loaded, followed by periodic, incremental loading of data changes such as updates to certain data points. On a recurring but less frequent timescale, full refreshes to erase and replace data in a warehouse can occur.

For most companies, the ETL process is fully automated. Indeed, many ETL systems already exist such as Azure Data Factory [89] from Microsoft and DataStage from IBM [90]. Some of the transformations performed in the transform step of the ETL process closely resemble those found in the pre-processing step of the CRISP-DM. However, these two processes do not share the same goal. Using a typical ETL software such as those mentioned above does not solve the issues raised in the problem statement paragraph of the present Master thesis (section 1.3).

Indeed, the primary sales argument that can be found for the previously mentioned systems resembles the following "*Real world data is messy and needs to be homogenized before being stored on a company's system*". The problem at hand, however, would require a tool whose primary sales argument resembles "*The concepts of pre-processing are now easier to apply for those who understand them but have little to no coding experience*". A codeless

platform would make pre-processing accessible to more people. To illustrate this previous statement, a quote from the promotional video for Microsoft's DataFactory on their website follows. "*[…] data is coming in faster and in a greater variety of formats than ever before*" [91].

At first glance, the pre-processing project of the present Thesis is similar to existing commercial programs. However, the transformations we support in the Transform stage of the ETL process focus heavily on the transformations needed to pre-process data.

3.1.2. A GENERALIZED TOOL FOR A FLEXIBLE TASK

The ETL tool we aimed to build during the present project had modularity as one of the characteristics needed for the program. Every user having a unique business need, we build the program in such a way that the transformations supported by the system are stored in a registry, making it trivial to support more until every business need ends up being supported by the program. The same can be said of the modules that import the data into the system. Some users have the data in a .csv file format, others want to query a database to get the desired data, and further users have a different source of data altogether. Here again, we elected to support as many different workflows as possible using a modular system. Exporting processed data, once the manipulations are done, works in the same way.

3.1.3. PROJECT INTEGRATION

We did not believe that it a specific technology has to be used with our platform. Instead, we believe that the end users know best what other tools in their workflow they should use. This paradigm pushed us to support as many import and export modules as needed to ensure compatibility with as many use cases as possible.

This also makes our platform technology agnostic. Regardless of where the data comes from, if there is a way of importing it into the platform, the system continues to work. We felt

this was important as predicting the future can be quite difficult, and not just in the IT world. There are new technologies emerging every day and some are innovative enough to motivate a company to migrate to it every so often. It was crucial to us that the lifespan of our project was not limited to the relevance of today's technologies.

3.2. THE PROJECT'S ARCHITECTURE

The project's architecture has been made to be as simple and modular as possible by relying on as few concepts as possible. The core concepts the project's architecture is based on are listed next.

- Statements and expressions,
- Forms,
- Dynamic variable resolution
- Front and back-end communication using Application Programming Interface (API) calls.

3.2.1. STATEMENTS AND EXPRESSIONS

A statement is, in short, an action performed on the data whereas an expression evaluates and returns some value. Figure 8 illustrates an expression by using a part of the Euclidean algorithm as an example [92]. In the context of the present project, a statement refers to either an importer, a transformer, or an exporter. This operation can be composed of multiple smaller components such as variables or parameters which would be classified as expressions. These different steps have a hierarchy to them, take the average of three numbers for example. The discreet expressions would be to add the first two numbers, to then add the first sum to the third number and then to take that sum and divide it by three. To model the priority of operations, the back-end has a functioning abstract syntax tree as seen in Figure 9. To properly work as described, the expressions are modelled using the interpreter pattern [93].

```
function (a,b)
  while b ≠ 0:
    if a > b:
      a := a - b
    else:
      b := b - a
return a
```

Figure 8 : Code snippet from the Eucledian algorithm © Hans Darmstadt-Bélanger

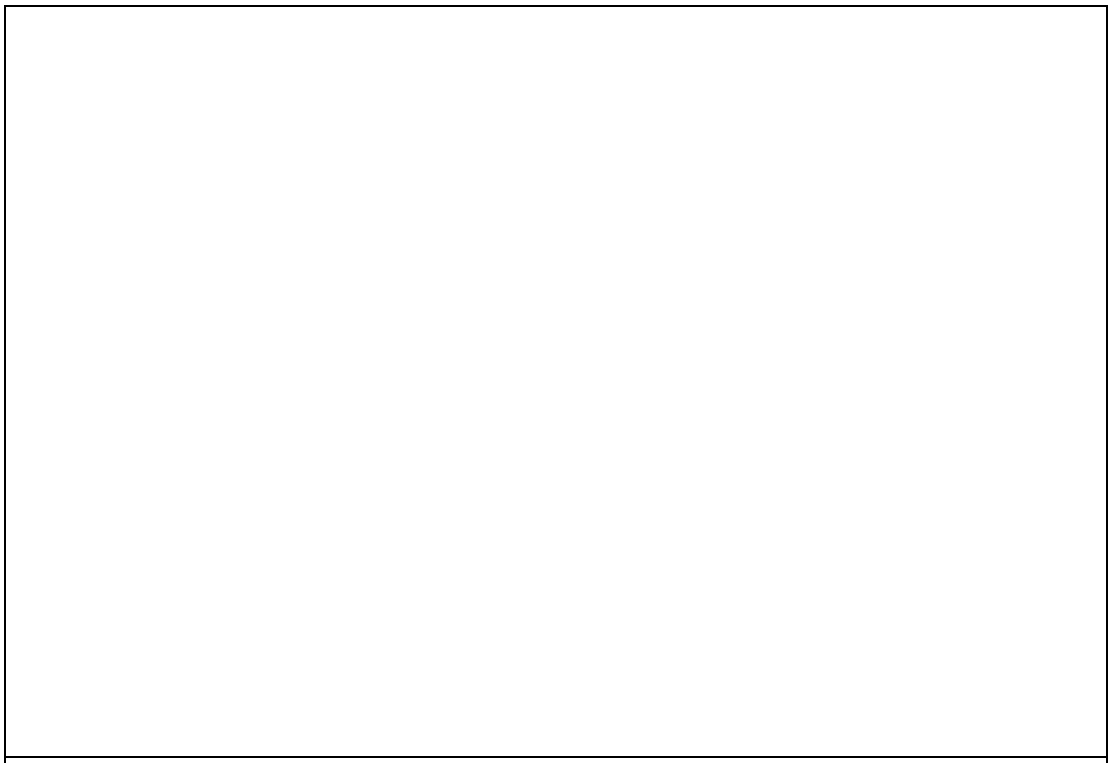

Figure 9 : Abstract syntax tree for code in Figure 8 © Hans Darmstadt-Bélanger

In practical terms, each statement and expression is encapsulated as a node in the tree. Each node, even if they are encapsulating different expressions or statements will function the same using the object-oriented programming (OOP) concept of inheritance to ensure they all implement the required functions to properly run. Notably, the exec() and eval() functions will run the actual code for the statements and expressions.

Each Statement has three core parts:

- The node mentioned above,
- The business class (either the importer, transformer or exporter itself), referenced by the node instance,
- The form.

The form is discussed next.

3.2.2. FORMS

The statements and expressions require user inputs to function properly. The forms manage the needed inputs by being composed of fields corresponding to the inputs. These fields are of one of multiple types, using inheritance principles here too. The available input types are matched to a html form input types when applicable. When building a composite expression, using simple html form elements will not suffice. The user interface covering the custom elements, using drag and drop is discussed in section 4.1 and afterwards.

Each available statement (importers, transformers, and exporters) and expression is stored in a registry. This way, the front end can display a list of the entries in the registry to list all available transformers when the user wants to add another transformer to their pre-processing workflow. Each statement has an associated form in another registry. This is how the front end knows which user inputs are required to properly utilize a given transformer.

3.2.3. VARIABLE RESOLUTION

When the user imports a data frame, among other use cases, the program will need to dynamically store and access this newly created variable when needed. For this purpose, a map has been created with the entry's key being the variable name the user has given to the new data frame.

3.2.4. FRONT AND BACK-END COMMUNICATION

The front and the back-end are both independent programs. The backend exists within a representational state transfer (REST) API that gets called by the front end. The front-end is a webpage. The needed endpoints are less numerous than could instinctively be supposed. Roughly one endpoint for each of the concepts mentioned above is needed.

The */forms* endpoint returns all the statement forms contained in the `statement_forms` registry.

The */expressions* endpoint Returns all the expression forms contained in the `expression_forms` registry.

The */download/<filename>* endpoint Returns the given file from the download folder for client side download.

The */create* endpoint Creates a new statement. It also returns a log trace and the statement id in case of success.

The */execute* endpoint executes a statement. And returns a report and the execution results id in case of success.

The */remove* endpoint Removes a statement and returns a log trace.

CHAPTER 4

USING THE PLATFORM

The visual layout design choice is inspired by the open-source Integrated Development Environment (IDE) Visual Studio Code [94], which was used to develop the application. Specifically, the navigation of the program is based on an Activity Bar and on a Side Bar. The other interface elements present in Visual Studio Code were not replicated in the present project.

When using the program, a typical user navigates from one menu to another by selecting the desired menu in the activity bar. Doing so reveals the various options and settings of the program in the sidebar. Since ease of use for non-experts is a core objective of the present project, the program features a second sidebar at the far right of the screen. It displays documentation relevant to the currently active section of the program. The last UI element is the largest on the screen. In the middle, between the left and right sidebars sits the main activity. This is where the bulk of the work happens. For example, when creating an importer, certain parameters need to be entered by the user to get the desired result. These parameters are set using the menus corresponding to the importer that are displayed in the main activity. Although not always visible, the program also has overlaid pop-up style menus that appear when the user creates a new importer, transformer or exporter. This menu asks the user which one of the supported importers, transformers or exporters the user wants to create using a dropdown menu.

The different buttons and options to be found in each of the described portions of the UI are discussed next.

4.1. THE ACTIVITY BAR

As mentioned in the previous paragraph, the far left of the screen is occupied by an activity bar featuring four icons in a column arrangement. These four icons are aligned at the top of the screen. The icons, from top to bottom are the following:

- Getting started tab,
- Statements tab,
- Dataset viewer tab,
- Expressions tab.

Each tab is discussed in the following paragraphs.

4.1.1. GETTING STARTED

This is the starting screen of the application. The associated sidebar items are the “Welcome” and the “Documentation” tabs. By default, the “Welcome” tab is selected. It hosts the welcome splash screen, containing some light documentation on how to get started.

The “Documentation” tab has a self-explanatory name and reveals documentation about the various UI elements that any proficient user of the program needs to learn. The titles of the various paragraphs of the documentation are:

- Getting started tab,
- Statements tab,
- Creating a statement,
- Filling a statement form,
- Executing a statement,
- Data tab,
- Expressions tab.

Each subject the documentation is covered in the following paragraphs of the present Master thesis.

4.1.2. STATEMENTS

The statements screen is where the user will spend most of their time in the application. As mentioned in section 3.2.1, each pre-processing task is considered as a statement. Every statement is created and managed in the screen presented in Figure 10 and Figure 11.

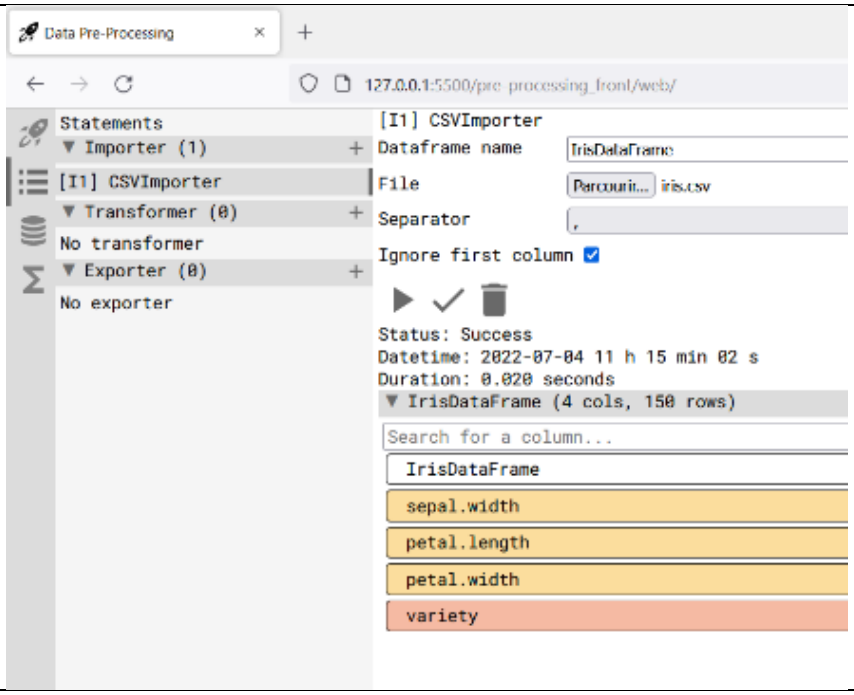

Figure 10 : The statements view with an importer as active statement © Hans Darmstadt-Bélanger

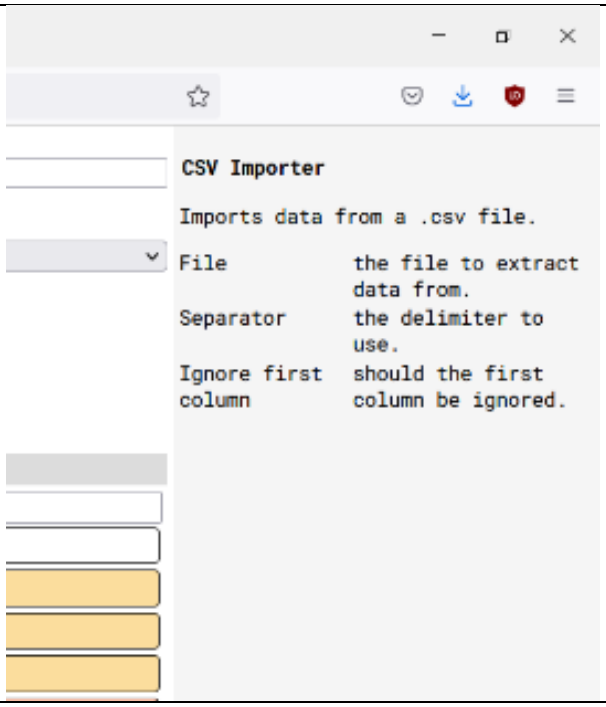

Figure 11 : Right hand short documentation of the statements view with an importer as active statement © Hans Darmstadt-Bélanger

The sidebar reveals the list of created statements. When the statement screen is active, statements are grouped by their type (importers, transformers, and exporters). The main window displays the settings for the statement the user is currently working on. For example, when creating a csv import statement, the program needs to know the following parameters:

- What the name of the resulting dataframe is,
- What the location of the file on the disk is,
- Which character acts as a separator,
- Whether the first row contains column names and should be ignored.

Each statement requires different parameters. The example provided above is for illustration purposes rather than for extensive documentation purposes.

During the normal workflow, the user creates the importer and enters the settings such as providing the location and the format of the data. Once the statement has been properly set, it

can be executed which allows the user to see the resulting data in the data screen, which will be discussed in the following paragraph.

In the backend, the statement is executed, not just queued up to be all executed sequentially at once at the end of the workflow. This was an important design decision, as many steps use the output of the previous statement as input for their tasks. Good examples are importer statements. Their output is the input for the following transformers. Regular transformers such as imputers also have the potential to write their transformation over the column. They are imputing the data into an existing column or to a new column to avoid any loss of data.

After the execution of the importer statement, the typical use case is the user creating and customising the required transformers.

To create a new statement, the user has to click on the + icon to the right of the type of statement he or she wants to create in the sidebar. The created statements of the same type can be seen under their respective titles. After clicking this create button, a pop-up style window appears, prompting the user to select which of the statements of the selected type he or she want to create using a dropdown menu (Figure 12). Once a statement has been selected, the popup menu adds a short documentation paragraph about the statement and the list of parameters the user needs to set (Figure 13).

Once the statement has been selected and the confirm button at the bottom right of the pop-up window has been selected, the window closes and the statement gets added to the list of created statements, as described above.

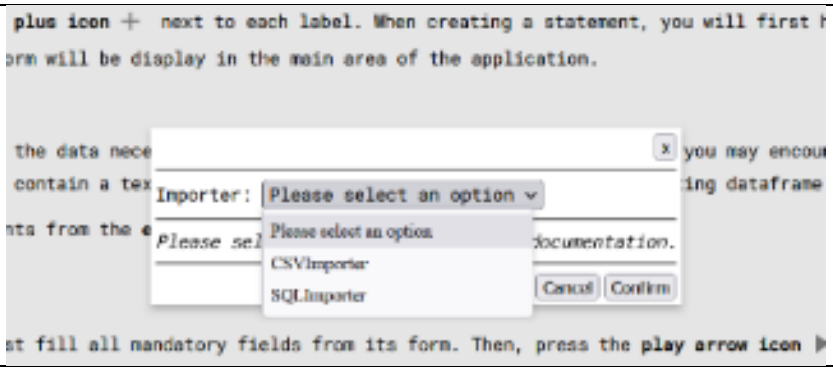

Figure 12 : Dropdown menu, creating a new statement in the pop-up window © Hans Darmstadt-Bélanger

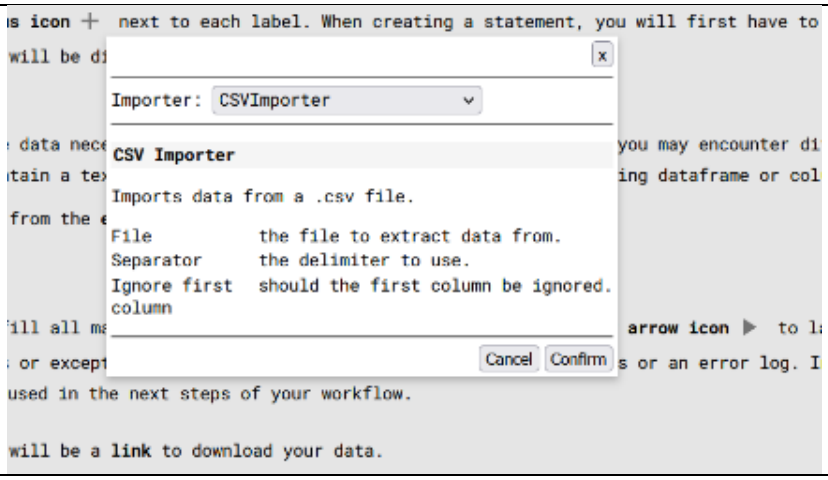

Figure 13 : Selected statement's documentation shown once selected © Hans Darmstadt-Bélanger

4.1.3. THE DATASET VIEWER

The dataset viewer is active when the user selects the third icon in the form of a database symbol in the activity bar (Figure 14). Unlike the other settings until now, when selected, the sidebar changes its content. The main window, however, remains unchanged. The purpose of this menu is to perform drag and drop manipulations to indicate with which dataset and column(s) the transformer and exporter statements should interact. These concepts are explained in detail next.

Figure 14 : The dataset viewer with the iris dataset [95] loaded © Hans Darmstadt-Bélanger

The dataset viewer populates the sidebar with the various datasets that have been loaded in the program in a box with a white background. Under each dataset name is the list of every column in the dataset with a color-coding to indicate at a quick glance what type of data the column stores. The color-coding is as follow:

- Turquoise elements represent a date,
- Powder blue elements represent a Boolean value,
- Mustard yellow represents numerical values,
- Salmon red represents strings of text.

This sidebar also features a search bar to filter out the columns of a dataset that do not match with the query entered. This feature is useful to quickly find the column the user is looking for when dealing with high dimensionality datasets.

The main way the columns and the transformers interact with one another from the user interface's perspective is by drag and drop. When the user creates a new transformer or exporter statement, the user usually needs to input the dataset(s) or column(s) on which the statement should act. To select the desired dataset(s) or column(s), the user drags the relevant item from the sidebar to the form element in the main window that has been displayed when the user last selected or created the statement in question. The list of datasets and columns has its vertical scroll bar, thus ensuring that every element is still accessible even if there are more items than can be displayed in the screen at once.

4.1.4. THE EXPRESSIONS MENU

The expressions menu can be described as the piece of the program that the end user interacts the least with while being the menu with the most advanced set of features.

Sometimes, the user needs to describe a condition to the program, for example, keeping only the records with a date prior to a given date. These types of Boolean checks are called expressions, and they can vary wildly from one project to another.

Visually speaking, the expressions menu reveals the list of constants that the user may need, followed by the list of Boolean functions that can be used.

The types of constants are the same as have been listed when describing the color-coding in the previous sections. These possible types and values are listed next:

- Boolean true value,
- Boolean false value,
- Date variable, the exact value to be entered by the user,
- Number variable, the exact value to be entered by the user,
- String variable, the exact value to be entered by the user.

The exhaustive list of functions the program supports are as follow:

- And,
- Equals,
- Greater than,
- Less than,
- Subtraction,
- Different than,
- Or,
- Addition.

To manage expressions, the same design philosophy applies. When a user creates a statement, the inputs parameters required from the user are visually represented as rectangles

in which the user drags and drops the desired values, be they a dataset, the column for a dataset, or even an expression (Figure 15).

These expressions, just like the transformers, require inputs for both sides of their Boolean question, which will be filled using drag and drop. It should be noted that not every function is technically a Boolean question, namely the addition and the subtraction are not. It is possible to nest expressions within one another. For example, the user can build an expression that reads as follow: only keep the rows where column A is bigger than column B plus column C.

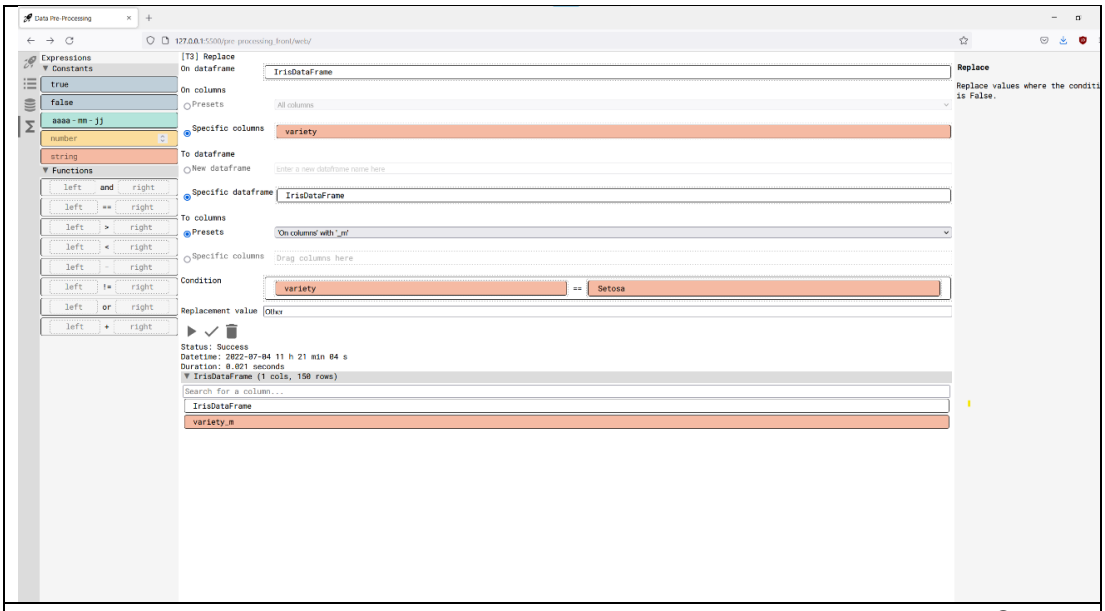

Figure 15 : The expression sidebar open with a statement form in the main activity © Hans Darmstadt-Bélanger

4.2. QUANTIFYING THE USEFULNESS OF THE PROGRAM

Once the project was completed and in an operational state, one of the last steps was to quantify the amount of time saved when using the program as compared to writing the pre-processing scripts by hand.

To do so a simulated pre-processing workload has been selected. The participants were asked to execute the pre-processing tasks once by writing a script in the programming language of their choice. They were then asked to execute the same pre-processing tasks using the project.

4.2.1. THE PRE-PROCESSING WORKLOAD

The participants were given two proprietary datasets that they were asked to use to perform the following tasks.

- Import the first dataset,
- Impute the missing values in the specified column by the letter “O”,
- Import the second dataset,
- Merge both datasets by using the specified column as pivot column,
- Export the resulting dataset to a csv file.

While performing these tasks, the participants were asked to time themselves to measure the difference in time it takes them when comparing the efficacy of the project against the manual scripting approach.

4.2.2. THE RESULTS

The pool or potential participants was limited to CGIC employees who had the technical knowledge to preprocess data. Furthermore, the responders did so on a voluntary basis. These factors put together resulted in only two responders volunteering their time. This number is low and does not lend itself to drawing solid conclusion from the results that follow. Both responders are professional data analysts. They represent the worst case scenario as their skill with pre-processing scripts minimises the potential for time gain compared to the case of the less experienced actuarial professional. Their observations are discussed next.

To briefly sum up the feedback before getting into the details, the responses were in line with what we were expecting. As the program works as intended, the bulk of the comments were suggesting ergonomic improvements or suggesting new features that we were not included in the initial version due to the scope of the project.

One of the first observation both responders brought up is the substantial loading time when importing the datasets. This is due to the tests having taken place on the development and test environments rather than the production environment. Needing to wait roughly 7 and 3 minutes to upload a 50mb dataset is a valid criticism, but it is not one the end users will have.

Another opinion both responders had in common was that it would be useful to be able to visualise the data inside the app as we currently rely on third party tools to do that.

The value to be imputed when using the imputer transformer needs to be passed to the program by the user. It can be a static value such as “-1”, a condition built by the user, e.g. “columnValue > 5” or the actual absence of a value, using the *np.nan* function from numpy [96]. The UI has three radio buttons to let the user pick their desired choice with the dynamic inputs allowing the user to enter the desired value or condition. In the case of the static np.nan option, the UI only displays “Na” which was confusing to one responder.

It is possible for the program to raise an error. A user could for example build a condition smaller than (<) and try to apply it on a text column of a data frame. When this happens, as the backend is an API, the entire backend does not crash and the traceback of the error is returned to the frontend to be displayed. These tracebacks are supposed to be a useful tool for understanding what went wrong. As any experienced developer will know, they can be quite opaque for the end user and did not help them to understand which part of their manipulation led to the error. One responder did not finish the list of pre-processing tasks that were asked of them due to this issue.

One suggestion we had not thought of was to build a feature allowing the users to build their pre-processing workflow using the frontend to then download the corresponding python code. Using a local script would eliminate some of the overheads inherent by using a web-

based solution. The suggestion mentioned that this downloaded code could also be used as a starting point to be adapted later rather than writing a script from scratch.

As one of the responders did not finish the pre-processing tasks, they could only estimate that they were being about twice as fast as when writing the equivalent python script by hand. The other responder could finish the tasks and had a similar experience. The overall times remained the same when writing code and using the platform. Half of the time spend on the platform was due to the abnormal upload times addressed previously.

In ideal conditions, the program seems to cut down the time needed to perform basic pre-processing tasks by half for experienced data analysts. This statement is made with the caveats mentioned in the previous paragraph. The usefulness of the platform is therefore validated and the results demonstrate that even professional data analysts and other experts stand to gain when using the platform.

For more detailed results, it would be pertinent to conduct the experiment again with a greater pool of responders. This is discussed in section 5.3.

CHAPTER 5

USING THE PLATFORM

Pre-processing is an important step in ML and can be one of the most time consuming steps of an ML project. When the person doing the project is not a ML expert, the times required can be expected to increase. This is in part due to the fact that some level of programing is needed to write a script that will perform the desired pre-processing tasks. Therefore, even if the user has the required knowledge to understand what they are trying to do, the programming requirement constitutes a barrier to entry.

5.1. PROJECT'S OBJECTIVES

CGIC identified this issue as many of their employees are actuarial professionals with extensive knowledge of the mathematical concepts behind ML and predictive modelling. This makes these employees qualified to build the predictive models that any insurance company relies on. However, these actuarial professionals usually have programming skills that are limited compared to their mathematical proficiency levels.

To solve this issue, the project described in the present master's thesis has been proposed. CGIC needed a tool that could enable its users to perform pre-processing tasks without needing the need to code. By abstracting the code behind a graphical user interface, the time needed to perform the pre-processing tasks of a given ML project can be compressed down to the time it takes to click the appropriate buttons in the UI. One typical pitfall to use graphical tools is that in the name of making the given program simple to use, the developers of these tools dumb their programs down. This happens since having fewer buttons on screen for each feature makes the program inherently more user friendly. It is easier to learn how to use a program if there are fewer buttons to learn in the first place. This kind of approach comes to a cost to the more advanced features of the program. Since this project is meant to be used by professionals, the project needs to pack the advanced features that will be needed. Otherwise, the end user would still find themselves forced to create pre-processing scripts. All

the while making the program intuitive enough to ensure that navigating the UI remains faster than the current status quo.

5.2. RESULTS

To help reduce these pre-processing times, an ETL type of platform has been developed. The graphical interface boasts a familiar layout and uses intuitive controls to make the user feel at ease, without hiding away advanced settings and parameters. This makes the program more opaque for users that have little knowledge in data manipulation, but ensures the knowledgeable user has the required granularity accessible to them to perform more advanced tasks.

While other ETL programs exist and are commercially available, to the best of our knowledge, no alternative offers the same focus on pre-processing transformations as the present program. The commercial alternatives focus their efforts on homogenizing new data to standardize it before saving it in a company's data warehouse.

5.3. FUTURE WORK

As the present project has been conducted in partnership with and for CGIC, the chances of the program becoming publicly available are slim to none. It would therefore be beneficial to the data science community if the platform could be reproduced and made accessible, either in a commercial or non-profit way.

The potential features discussed next were considered outside the scope of the project by CGIC and were quickly identified as the most interesting features to add in a potential second iteration of the platform. Before new features are added to the platform however, it would be advised to perform a survey at a larger scale to gather more quality data on the amount of time the use of the platform saves. Having more responders could potentially also underline certain unexpected workflows that do not work well with the platform.

Having a way to visualise the data within the program would be valuable to many users as the current implementation relies on third party solutions such as Excel.

As discussed in section 4.2.2, the error messages are opaque. When the user performs operations which lead to an error, the error should ideally be presented in a way that is easier to understand than simply returning the error call stack.

For the sake of automation, it would be beneficial for certain use cases to save the list of transformations to re-apply the same transformations on another dataset. The hypothetical analyst that periodically validates the models with the latest data would be a perfect example of use cases for this feature.

In its current form, the platform does nothing but pre-processing. However, the building blocks are in place to enlarge the scope of the platform. Adding the possibility to not just export the datasets but reports on these datasets would add value. A simple graph or pie chart exported in the form of an image would allow a user remaining on the platform when generating weekly reports. While adding this feature would denature the ETL platform from its pre-processing tool approach, section 3.1.2 describes why we build a modular platform with future expansion in mind.

Similarly, the modularity of the codebase allows future iteration of the program to have transformers that trains and generate a ML model. Most projects that need data to be pre-processed want to do something with the pre-processed data. Adding this functionality for the next step would create an ecosystem, which would be more convivial for the user, compared to jumping from one tool to another for the different steps of the project.

One of the bigger future features that we believe could bring added value to the end user would be the joining of the ETL platform with previous research projects such as the project presented in [57]. This research work consists of finding and applying the best pre-processing transformations to optimise the data to train a ML model. Such an approach is slightly limiting as the user can either accept the transformations or reject them entirely. A meta learner would integrate well with the platform as the user could preview the suggested transformations,

accept, or reject each transformation on an individual basis, and add transformers the user believes has been omitted by the meta learner.

5.4. PERSONAL DISCUSSION

I personally found the present project very interesting for multiple reasons. Most of my experience in IT at large has been acquired in an educational context, with professors handing out assignments with clearly defined expectations. This programming experience differed from my previous experiences due in part to the setting. Being in a research setting, CGIC came to us with a problem and left us the liberty of finding the best way to solve the problem. The metric for success was vaguely defined as to allow us to implement the solution we thought best, rather than forcing us down a predetermined path.

In the earlier steps of the project, we needed to clearly understand the problem that the CGIC employees are facing. Going around different work teams within the company and talking to the employees to understand clearly what it is that is causing problems to them and how we can alleviate the pain points was not something I had done when taking classes aimed at teaching me a clearly defined corner of the IT world.

Following these rounds of discussions, we had to find the best implementation of the solution to support the affected users. Here again, deciding what implementation of the solution we would end up taking was relatively new for me. To come up with a novel solution to an existing problem forced us to utilise critical thinking skills that were quite stimulating to develop. After having presented our findings to the CGIC employees to get their feedback, we iterated on the proposed solution a few times and eventually got a project charter approved by the relevant managers. The next step of the project was to develop the actual program.

Here, something predictable in hindsight occurred that I personally did not anticipate in the moment. Since the start of my Master's studies, the classes I took were centred around big data, machine learning and artificial intelligence. These classes were needed for me to gain

the expertise that would qualify me to work on this project. When talking about the project, all the talks were centred on machine learning and pre-processing, be it before the project started or in its first steps described above. Once we started to write code however, we were no longer doing pre-processing. After all, our goal was to support the employees with pre-processing workloads and not to do it ourselves. The code we wrote was based on OOP principles. The realisation hit me after the fact that in the context of this project, all the machine learning, big data, and artificial intelligence classes helped me to understand the problem. However, the skills needed to develop the solution were acquired prior to starting my master's studies. This statement should not be construed as a criticism. To make a comparison to an unrelated field, take the example of a program simulating aerodynamics to help engineers design aircrafts. It is normal for the programmer building such a tool to thoroughly understand the principles of aerodynamics to recreate them in the program's simulation, but the skills needed to build the tool in the first place are not exclusively related to the laws of physics. If anything else, it gave me a newfound appreciation for software architecture and other OOP related concepts as a way to ensure clean code and maintainable codebases.

To sum up, I really enjoyed how we had the liberty of implementing a novel solution to a known problem rather than spend the duration of the project accomplishing a list of predetermined tasks. I mention this aspect as it is this phenomenon that motivated me to pursue this level of studies. One of my previous internships was the one needed to complete my college degree. During this internship, tasks were handed to me by my team manager one at a time in such a way that there was little room for creative problem solving. This internship motivated me to pursue higher studies to reach a more specialised level of knowledge. My objective was for creative problem-solving skills to be a core part of the typical task description of a person with my background. As these lines are being written months after the end of the project and my subsequent employ at CGIC as a part time employee, I can state that this personal objective of mine has been accomplished.

REFERENCES

[1] J. Alzubi, A. Nayyar, and A. Kumar, "Machine learning from theory to algorithms: an overview," *Journal of physics: conference series,* vol. 1142, no. 1, p. 012012, 2018.

[2] M. A. Munson, "A study on the importance of and time spent on different modeling steps," *ACM SIGKDD Explorations Newsletter,* vol. 13, no. 2, pp. 65-71, 2012.

[3] D. C. Corrales Múñoz, A. I. Ledezma Espino, and J. C. Corrales, "From Theory to Practice: A Data Quality Framework for Classification Tasks," 2018.

[4] S. García, S. Ramírez-Gallego, J. Luengo, J. M. Benítez, and F. Herrera, "Big data preprocessing: methods and prospects," *Big Data Analytics,* vol. 1, no. 1, pp. 1-22, 2016.

[5] A. Famili, W.-M. Shen, R. Weber, and E. Simoudis, "Data preprocessing and intelligent data analysis," *Intelligent data analysis,* vol. 1, no. 1, pp. 3-23, 1997.

[6] G. Anthes, "Data brokers are watching you," ed: ACM New York, NY, USA, 2014.

[7] T. Furche, G. Gottlob, L. Libkin, G. Orsi, and N. W. Paton, "Data Wrangling for Big Data: Challenges and Opportunities," in *EDBT 2016*, Bordeaux, France, 2016, vol. 16, pp. 473-478.

[8] S. F. Crone, S. Lessmann, and R. Stahlbock, "The impact of preprocessing on data mining: An evaluation of classifier sensitivity in direct marketing," *European Journal of Operational Research,* vol. 173, no. 3, pp. 781-800, 2006.

[9] D. Seca and J. Mendes-Moreira, "Benchmark of encoders of nominal features for regression," in *Trends and Applications in Information Systems and Technologies: Volume 1*: Springer, 2021, pp. 146-155.

[10] I. WekaIO. "Weka." https://www.weka.io/ (accessed 2022-10-31.

[11] Rapidminer. "rapidminer." https://rapidminer.com/ (accessed 2022-10-31.

[12] Knime. "Knime." https://www.knime.com/ (accessed 2022-10-31.

[13] S. Institute. "Sas." https://www.sas.com/en_ca/home.html# (accessed 2022-10-31.

[14] scikit-learn. "scikit-learn Machine Learning in Python." https://scikit-learn.org/stable/index.html (accessed 2022-10-31.

[15] G. Biau and E. Scornet, "A random forest guided tour," *Test,* vol. 25, no. 2, pp. 197-227, 2016.

[16] L. Berti-Equille, "Learn2clean: Optimizing the sequence of tasks for web data preparation," in *The World Wide Web Conference*, 2019, pp. 2580-2586.

[17] J. Gerretzen *et al.*, "Simple and effective way for data preprocessing selection based on design of experiments," *Analytical chemistry,* vol. 87, no. 24, pp. 12096-12103, 2015.

[18] D. Pehl, *Die Implementation der Rasterfahndung*. Berlin, Germany: Duncker & Humblot, 2008.

[19] Anonymous, "Direct Compensation Property Damage (DCPD) Reform, Revised Private Passenger Grid Rates and Various Rule Changes for GISA (ASP) Updates Effective January 1, 2022 (New Business and Renewals)," Facility Association, Toronto, On, Canada, 2021.

[20] A. H. Al-Balbissi, "Role of gender in road accidents," *Traffic injury prevention,* vol. 4, no. 1, pp. 64-73, 2003.

[21] P. Baecke and L. Bocca, "The value of vehicle telematics data in insurance risk selection processes," *Decision Support Systems,* vol. 98, p. 69, 2017.

[22] Y.-L. Ma, X. Zhu, X. Hu, and Y.-C. Chiu, "The use of context-sensitive insurance telematics data in auto insurance rate making," *Transportation Research Part A: Policy and Practice,* vol. 113, pp. 243-258, 2018.

[23] F. Duval, J.-P. Boucher, and M. Pigeon, "How much telematics information do insurers need for claim classification?," *arXiv preprint arXiv:2105.14055,* 2021.

[24] J. Kepner *et al.*, "Computing on masked data: a high performance method for improving big data veracity," in *2014 IEEE High Performance Extreme Computing Conference (HPEC)*, 2014: IEEE, pp. 1-6.

[25] D. T. Larose, *Data mining and predictive analytics*. John Wiley & Sons, 2015.

[26] J. R. Quinlan, *C4. 5: programs for machine learning*. Elsevier, 2014.

[27] Y. Bengio, I. Goodfellow, and A. Courville, *Deep learning*. MIT press Cambridge, MA, USA, 2017.
[28] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," *Advances in neural information processing systems,* vol. 17, 2004.
[29] T. Furche, G. Gottlob, L. Libkin, G. Orsi, and N. W. Paton, "Data Wrangling for Big Data: Challenges and Opportunities," in *EDBT*, 2016, vol. 16, pp. 473-478.
[30] M. Lenzerini, "Data integration: A theoretical perspective," in *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 2002, pp. 233-246.
[31] W. Koehrsen, "Overfitting vs. underfitting: A complete example," *Towards Data Science,* 2018.
[32] M. F. Bosu and S. G. MacDonell, "A taxonomy of data quality challenges in empirical software engineering," in *2013 22nd Australian Software Engineering Conference*, 2013: IEEE, pp. 97-106.
[33] Merriam-Webster. "accuracy." https://www.merriam-webster.com/dictionary/accuracy (accessed 19-01-2023.
[34] B. Iglewicz, "Robust scale estimators and confidence intervals for location," *Understanding robust and exploratory data analysis,* vol. 404, p. 431, 1983.
[35] S. Bibi, G. Tsoumakas, I. Stamelos, and I. Vlahavas, "Regression via Classification applied on software defect estimation," *Expert Systems with Applications,* vol. 34, no. 3, pp. 2091-2101, 2008.
[36] A. Folleco, T. M. Khoshgoftaar, J. Van Hulse, and L. Bullard, "Software quality modeling: The impact of class noise on the random forest classifier," in *2008 IEEE congress on evolutionary computation (IEEE world congress on computational intelligence)*, 2008: IEEE, pp. 3853-3859.
[37] T. M. Khoshgoftaar and P. Rebours, "Improving software quality prediction by noise filtering techniques," *Journal of Computer Science and Technology,* vol. 22, no. 3, pp. 387-396, 2007.
[38] C. Catal, O. Alan, and K. Balkan, "Class noise detection based on software metrics and ROC curves," *Information Sciences,* vol. 181, no. 21, pp. 4867-4877, 2011.
[39] T. M. Khoshgoftaar, N. Seliya, and K. Gao, "Rule-based noise detection for software measurement data," in *Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration, 2004. IRI 2004.*, 2004: IEEE, pp. 302-307.
[40] G. A. Liebchen and M. Shepperd, "Software productivity analysis of a large data set and issues of confidentiality and data quality," in *11th IEEE International Software Metrics Symposium (METRICS'05)*, 2005: IEEE, pp. 3 pp.-46.
[41] G. Liebchen, B. Twala, M. Shepperd, and M. Cartwright, "Assessing the quality and cleaning of a software project dataset: An experience report," in *10th International Conference on Evaluation and Assessment in Software Engineering (EASE) 10*, 2006, pp. 1-7.
[42] L. Buglione and C. Gencel, "Impact of base functional component types on software functional size based effort estimation," in *International Conference on Product Focused Software Process Improvement*, 2008: Springer, pp. 75-89.
[43] J.-S. Chen and C.-H. Cheng, "Software diagnosis using fuzzified attribute base on modified MEPA," in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, 2006: Springer, pp. 1270-1279.
[44] T. M. Khoshgoftaar and J. Van Hulse, "Imputation techniques for multivariate missingness in software measurement data," *Software Quality Journal,* vol. 16, no. 4, pp. 563-600, 2008.
[45] A. R. Linero and M. J. Daniels, "Bayesian approaches for missing not at random outcome data: the role of identifying restrictions," *Statistical science: a review journal of the Institute of Mathematical Statistics,* vol. 33, no. 2, p. 198, 2018.
[46] J. Wang, Y. Liu, P. Li, Z. Lin, S. Sindakis, and S. Aggarwal, "Overview of Data Quality: Examining the Dimensions, Antecedents, and Impacts of Data Quality," *Journal of the Knowledge Economy,* pp. 1-20, 2023.

[47] M. J. Kiemele, S. R. Schmidt, and R. J. Berdine, "Regression Analysis," in *Basic statistics: Tools for continuous improvement*, 4th ed. Colorado Springs, CO, USA: Air Academy Press, 1997, ch. 7.
[48] E. Saccenti, M. H. Hendriks, and A. K. Smilde, "Corruption of the Pearson correlation coefficient by measurement error and its estimation, bias, and correction under different error models," *Scientific reports,* vol. 10, no. 1, p. 438, 2020.
[49] W.-C. Chen, A. Tareen, and J. B. Kinney, "Density Estimation on Small Data Sets," *Physical Review Letters,* vol. 121, no. 16, p. 160605, 10/19/ 2018, doi: 10.1103/PhysRevLett.121.160605.
[50] K. S. Button *et al.*, "Power failure: why small sample size undermines the reliability of neuroscience," *Nature reviews neuroscience,* vol. 14, no. 5, pp. 365-376, 2013.
[51] P. Obilikwu and E. Ogbuju, "A data model for enhanced data comparability across multiple organizations," *Journal of Big Data,* vol. 7, no. 1, p. 95, 2020/11/10 2020, doi: 10.1186/s40537-020-00370-1.
[52] K. Dickersin, E. Mayo-Wilson, and T. Li, "The Benefits and Challenges of Using Multiple Sources of Information about Clinical Trials," 2018.
[53] E. Kocaguneli, G. Gay, T. Menzies, Y. Yang, and J. W. Keung, "When to use data from other projects for effort estimation," in *Proceedings of the IEEE/ACM international conference on Automated software engineering*, 2010, pp. 321-324.
[54] Merriam-Webster. "provenance." https://www.merriam-webster.com/dictionary/provenance (accessed 2023-01-18.
[55] ISBJ. "ISBSG International Software Benchmarking Standards Group A non-profit organization helping IT professionals navigate project management." https://www.isbsg.org/ (accessed 19-01-2023.
[56] Promise. "PROMISE'19 CALL FOR PAPERS." https://promisedata.org/2019/index.html (accessed 19-01-2023.
[57] B. Bilalli, A. Abelló, T. Aluja-Banet, and R. Wrembel, "PRESISTANT: Learning based assistant for data pre-processing," *Data and knowledge engineering,* vol. 123, 2019.
[58] S. Uddin, I. Haque, H. Lu, M. A. Moni, and E. Gide, "Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction," *Scientific Reports,* vol. 12, no. 1, p. 6256, 2022.
[59] J. N. Wulff and L. E. Jeppesen, "Multiple imputation by chained equations in praxis: guidelines and review," *Electronic Journal of Business Research Methods,* vol. 15, no. 1, pp. 41-56, 2017.
[60] P. Schober, E. J. Mascha, and T. R. Vetter, "Statistics from A (agreement) to Z (z score): a guide to interpreting common measures of association, agreement, diagnostic accuracy, effect size, heterogeneity, and reliability in medical research," *Anesthesia & Analgesia,* vol. 133, no. 6, pp. 1633-1641, 2021.
[61] D. L. Whaley III, "The interquartile range: Theory and estimation," East Tennessee State University, 2005.
[62] S. Morasca, "Building statistically significant robust regression models in empirical software engineering," in *Proceedings of the 5th International Conference on Predictor Models in Software Engineering*, 2009, pp. 1-10.
[63] Y.-S. Seo, K.-A. Yoon, and D.-H. Bae, "An empirical analysis of software effort estimation with outlier elimination," in *Proceedings of the 4th international workshop on Predictor models in software engineering*, 2008, pp. 25-32.
[64] N. V. Chawla, N. Japkowicz, and A. Kotcz, "Special issue on learning from imbalanced data sets," *ACM SIGKDD explorations newsletter,* vol. 6, no. 1, pp. 1-6, 2004.
[65] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD explorations newsletter,* vol. 6, no. 1, pp. 20-29, 2004.
[66] G. T. Reddy *et al.*, "Analysis of dimensionality reduction techniques on big data," *IEEE Access,* vol. 8, pp. 54776-54788, 2020.
[67] J. Rahnenführer *et al.*, "Statistical analysis of high-dimensional biomedical data: a gentle introduction to analytical goals, common approaches and challenges," *BMC Medicine,* vol. 21, no. 1, p. 182, 2023/05/15 2023, doi: 10.1186/s12916-023-02858-y.

[68] R. Zebari, A. Abdulazeez, D. Zeebaree, D. Zebari, and J. Saeed, "A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction," *Journal of Applied Science and Technology Trends,* vol. 1, no. 2, pp. 56-70, 2020.

[69] H. Sanz, C. Valim, E. Vegas, J. M. Oller, and F. Reverter, "SVM-RFE: selection and visualization of the most relevant features through non-linear kernels," *BMC Bioinformatics,* vol. 19, no. 1, p. 432, 2018/11/19 2018, doi: 10.1186/s12859-018-2451-4.

[70] J. Lever, M. Krzywinski, and N. Altman, "Principal component analysis," *Nature Methods,* vol. 14, no. 7, pp. 641-642, 2017/07/01 2017, doi: 10.1038/nmeth.4346.

[71] X. Chu, I. F. Ilyas, S. Krishnan, and J. Wang, "Data cleaning: Overview and emerging challenges," in *Proceedings of the 2016 international conference on management of data*, 2016, pp. 2201-2206.

[72] S. Kandel *et al.*, "Research directions in data wrangling: Visualizations and transformations for usable and credible data," *Information Visualization,* vol. 10, no. 4, pp. 271-288, 2011.

[73] M. Yakout, A. K. Elmagarmid, J. Neville, M. Ouzzani, and I. F. Ilyas, "Guided data repair," *arXiv preprint arXiv:1103.3103,* 2011.

[74] X. Chu *et al.*, "Katara: A data cleaning system powered by knowledge bases and crowdsourcing," in *Proceedings of the 2015 ACM SIGMOD international conference on management of data*, 2015, pp. 1247-1261.

[75] Z. Abedjan *et al.*, "Detecting data errors: Where are we and what needs to be done?," *Proceedings of the VLDB Endowment,* vol. 9, no. 12, pp. 993-1004, 2016.

[76] A. Quemy, "Data Pipeline Selection and Optimization," in *DOLAP*, 2019.

[77] R. Singh and S. Gulwani, "Learning semantic string transformations from examples," *arXiv preprint arXiv:1204.6079,* 2012.

[78] O. Ivanciuc, "Weka machine learning for predicting the phospholipidosis inducing potential," *Current topics in medicinal chemistry,* vol. 8, no. 18, pp. 1691-1709, 2008.

[79] P. Bohannon, W. Fan, M. Flaster, and R. Rastogi, "A cost-based model and effective heuristic for repairing constraints by value modification," in *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, 2005, pp. 143-154.

[80] F. Chiang and R. J. Miller, "A unified model for data and constraint repair," in *2011 IEEE 27th International Conference on Data Engineering*, 2011: IEEE, pp. 446-457.

[81] X. Chu, I. F. Ilyas, and P. Papotti, "Holistic data cleaning: Putting violations into context," in *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, 2013: IEEE, pp. 458-469.

[82] F. Geerts, G. Mecca, P. Papotti, and D. Santoro, "The LLUNATIC data-cleaning framework," *Proceedings of the VLDB Endowment,* vol. 6, no. 9, pp. 625-636, 2013.

[83] S. Song, H. Cheng, J. X. Yu, and L. Chen, "Repairing vertex labels under neighborhood constraints," *Proceedings of the VLDB Endowment,* vol. 7, no. 11, pp. 987-998, 2014.

[84] C. Mayfield, J. Neville, and S. Prabhakar, "ERACER: a database approach for statistical inference and data cleaning," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, 2010, pp. 75-86.

[85] M. Yakout, L. Berti-Équille, and A. K. Elmagarmid, "Don't be scared: use scalable automatic repairing with maximal likelihood and bounded changes," in *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, 2013, pp. 553-564.

[86] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum, "YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia," *Artificial intelligence,* vol. 194, pp. 28-61, 2013.

[87] M. Morsey, J. Lehmann, S. Auer, and A.-C. Ngonga Ngomo, "DBpedia SPARQL benchmark–performance assessment with real queries on real data," in *International semantic web conference*, 2011: Springer, pp. 454-469.

[88] Z. Abedjan, J. Morcos, I. F. Ilyas, M. Ouzzani, P. Papotti, and M. Stonebraker, "Dataxformer: A robust transformation discovery system," in *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, 2016: IEEE, pp. 1134-1145.

[89] B. Kettner and F. Geisler, "Azure Data Factory," in *Pro Serverless Data Handling with Microsoft Azure: Architecting ETL and Data-Driven Applications in the Cloud*, B. Kettner and F. Geisler Eds. Berkeley, CA: Apress, 2022, pp. 93-119.

[90] IBM. "IBM DataStage." https://www.ibm.com/products/datastage (accessed 12-10-2022.

[91] Microsoft. "Azure Data Factory Hybrid data integration at enterprise scale, made easy." https://azure.microsoft.com/en-us/products/data-factory/#overview (accessed 2022-10-31.

[92] F. Lemmermeyer, "The Euclidean algorithm in algebraic number fields," *Expositiones Mathematicae,* vol. 13, pp. 385-416, 1995.

[93] M. Hills, P. Klint, T. v. d. Storm, and J. Vinju, "A case of visitor versus interpreter pattern," in *International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, 2011: Springer, pp. 228-243.

[94] B. Johnson, *Visual Studio code : end-to-end editing and debugging tools for web developers*, Indianapolis, Indiana: Wiley, 2019. [Online]. Available: https://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=2228945
https://public.ebookcentral.proquest.com/choice/publicfullrecord.aspx?p=5849509
https://proquest.safaribooksonline.com/9781119588184
https://doi.org/10.1002/9781119588238.

[95] "Research Data." Institute for Researcg ib Ubbivatuib & Science. https://iris.isr.umich.edu/research-data/ (accessed 25-10-2022.

[96] Numpy. "NumPy The fundamental package for scientific computing with Python." https://numpy.org/ (accessed 10-11-2022.