William Haavisto

Automating the Certificate Verification Process

School of Technology and Innovations
Master's Thesis
Master's Programme in Industrial System Analytics

Vaasa 2024

UNIVERSITY OF VAASA
School of Technology and Innovations
Author: William Haavisto
Title of the Thesis: Automating the Certificate Verification Process
Degree: Master of Science in Technology
Programme: Industrial System Analytics
Supervisor: Ahm Shamsuzzoha
Year: 2024 **Pages:** 104

ABSTRACT:
Automation has seen a rapid growth during the recent decade and has evolved almost every industry, as it has allowed processes to become more reliable and efficient. One of the main objectives with automation is to reduce or eliminate time-consuming and tedious repetitive tasks to allow time to be allocated to more important tasks. Similarly, this thesis sought to explore how the process of manually verifying certificates at a case company specialized in calibration equipment manufacturing and services could be improved with modern tools such as machine learning to verify that the measurement results in the certificates were correct while simple rule-based approaches could be applied to other parts of the certificate where faults usually occurred to create an assistant to aid technicians during the verification process.

To structure the thesis, a simplified version of the CRISP-DM framework was used, which consisted of four different phases. First, a focus group interview with the chief of the laboratory and technicians was held to map out how the current process worked, what the data in the certificates implied, where faults usually occurred and what kind of solution was desired. These answers were used as the requirements during the development of a potential solution. Second, methods to prepare the certificate data were developed, both to prepare data sufficient enough to train a model as well as being able to extract data from the certificate which had to be verified. The third phase consisted of developing the models, where seven different models were compared and evaluated, out of which four were selected for further evaluation. In the last phase, the performance of the selected models was evaluated where unseen data was used as the input and the prediction the model made as the output.

The results indicated that the selected machine learning models all performed exceptionally well and were able to make accurate predictions, especially the Extra Trees algorithm showed promising results on the two different datasets used during the thesis. With the results, a solution which includes a small modification to the current certificate printing tool as well as a web service which would handle the certificate verification and return the verified certificate to the technician for further analysis was proposed. Due to the time taken to define the requirements as well as experimentations with the machine learning models and data extraction methods, the solution could only be proposed but a small proof of concept was developed to evaluate the feasibility of the solution, which resulted in four managerial implications being identified. These included establishing consistency in the process, improved efficiency, cost reduction as well as continuous improvement. Considering the findings and the conclusions made, the project could be considered a success as the research objectives were met and questions answered but would still require more development and testing before the proposed solution could be deployed to the production environment.

KEYWORDS: Artificial Intelligence, Data Analytics, Database Management, Process Improvement, Software Development

VAASAN YLIOPISTO
Tekniikan ja Innovaatiojohtamisen Yksikkö
Tekijä: William Haavisto
Tutkielman nimi Todistuksien Tarkistus Prosessien Automatisointi
Tutkinto: Tekniikan Maisteri
Oppiaine: Industrial System Analytics
Työn ohjaaja: Ahm Shamsuzzoha
Valmistumisvuosi: 2024 **Sivumäärä:** 104

TIIVISTELMÄ:
Automatisointi on kasvanut nopeasti viime vuosikymmenen aikana, ja auttanut teollisuutta kehittymään, sillä automatisointi on mahdollistanut luotettavampia sekä tehokkaampia prosesseja. Yksi digitalisoinnin tärkeimmistä tavoitteista on vähentää tai kokonaan poistaa turhaa aikaa vieviä tehtäviä, jotta aikaa voidaan priorisoida tärkeämpiin tehtäviin. Tässä opinnäytetyössä pyrittiin tutkimaan, miten eräässä kalibrointilaitteiden valmistukseen ja ratkaisuihin erikoistuneessa yrityksessä manuaalisesti suoritettavaa sertifikaattien tarkistusprosessi voitaisiin parantaa nykyaikaisilla työkaluilla, kuten koneoppimisella, jolla olisi mahdollista tarkistaa sertifikaattien mittaustulokset tarkistusprosessin aikana sekä soveltaa yksinkertaisia ratkaisuja muihin kohtiin sertifikaatissa, joissa yleensä virheitä esiintyi. Tämän avulla voitaisiin kehittää teknikoille avustaja, joka helpottaisi ja parantaisi tarkistusprosessia.

Opinnäytetyön struktuurina käytettiin yksinkertaistettua versiota CRISP-DM-mallista, joka koostui neljästä eri vaiheesta. Ensimmäisessä vaiheessa järjestettiin laboratorion päällikön ja teknikkojen kanssa ryhmähaastattelu, jossa kartoitettiin, miten nykyinen prosessi toimi, mitä todistusten tiedot tarkoittavat, missä yleensä ilmeni virheitä ja miten optimaalisen ratkaisun pitäisi toimia. Näitä vastauksia käytettiin vaatimuksina mahdollisen ratkaisun kehittämiseen. Kehitettiin menetelmiä sertifikaattitietojen valmistelemiseksi, sekä mallin kehitykseen riittävien tietojen valmistelemiseksi että tapa kerätä tarkistettava tieto sertifikaatista. Kolmas vaihe koostui itse mallin kehittämisestä, jossa vertailtiin ja arvioitiin seitsemää eri mallia, joista neljä valittiin lupaavien tuloksien perusteella. Viimeisessä vaiheessa valittujen mallien suorituskykyä arvioitiin sertifikaattitiedoilla, jota mallit eivät olleet ennen nähneet ja mallien ennusteiden perusteella tehtiin lopullinen arvio.

Tulokset osoittivat, että kaikki valitut koneoppimismallit toimivat poikkeuksellisen hyvin ja pystyivät tekemään tarkkoja ennusteita, erityisesti Extra Trees -algoritmin tulokset olivat lupaavia. Tuloksien perusteella ratkaisua ehdotettiin, johon sisältyy pieni muutos nykyiseen sertifikaattien tulostustyökaluun sekä uusi verkkopalvelu, joka hoitaisi sertifikaattien tarkistamisen ja palauttaisi tarkistustulokset teknikolle. Vaatimusten kartoittamiseen sekä koneoppimismallien ja tiedonkeräys menetelmien kehittäminen vei enemmän aikaa, kun aluksi oletettiin, minkä takia ratkaisua ei pystytty muuta kuin ehdottamaan. Ratkaisun kelpoisuuden todentamiseksi oli kumminkin mahdollista kehitettä konseptitodistus, jonka tuloksista oli mahdollista kartoittaa neljä johtamisvaikutusta, joihin kuuluivat vakaammat tulokset sertifikaattiprosessista, tehokkuuden parantaminen, kustannusten vähentäminen sekä jatkuva parantaminen. Kun huomioon otetaan havainnot sekä tehdyt johtopäätökset, tutkimusta voidaan pitää onnistuneena, sillä tutkimustavoitteet saavutettiin ja tutkimuskysymyksiin vastattiin, mutta jotta ratkaisu saataisiin käyttöön tuotantoon, palvelu vaatisi enemmän kehitystä sekä testejä, jotta ehdotettu ratkaisu pystyisi todeta olevan toteutettavissa.

AVAINSANAT: Tekoäly, Tietoanalytiikka, Tietokannan Hallinta, Prosessien Parantaminen, Ohjelmistokehitys

Contents

Abbreviations

AI – Artificial Intelligence

CER – Character Error Rate Word Error Rate

CM – Confusion Matrix

COL – Chief Of Laboratory

CPT – Certificate Printing Tool

CRISP-DM – Cross-Industry Standard Process for Data Mining

CV – Cross Validation

DL – Deep Learning

ETC – Extra Trees Classifier

GBC – Gradient Boosting Classifier

GNB – Gaussian Naïve Bayes

KNC – K-Nearest Classifier

LR – Logistic Regression

ML – Machine Learning

OCR – Optical Character Recognition

ORM – Object Relational Mapper

PDF – Portable Document Format

PO – Product Owner

POC – Proof Of Concept

RPA – Robotic Process Automation

RFC – Random Forest Classifier

RQ – Research Question

SMOTE - Synthetic Minority Oversampling Technique

SQL - Structured Query Language

SVC – Support Vector Classifier

VGEN – Voltage Generation

VMEAS – Voltage Measurement

WER – Word Error Rate

List of Figures

1 Introduction

Repetitive tasks are a common challenge often encountered in almost every industry. These can range from manually filling out a spreadsheet, working at a factory stamping station or checking that all of the elements required in a document are correct. While these tasks are not difficult to do, they often consume a significant amount of time that could otherwise be allocated to other more important tasks. One concept which has seen huge advancements in recent years is machine learning (ML), which has the ability to perform complex calculations and take appropriate actions on its own (G. Shobana, 2018). Similarly, this thesis focused on how a repetitive task could be automated and if modern tools such as ML could be incorporated to improve the process.

However, completely automating tasks poses its own problems. Take for example the popular chatbot ChatGPT. In a study where the impact of artificial intelligence (AI) in higher education was conducted at public universities in Saudi Arabia, both students and faculty members were interviewed, and the findings revealed that some of the biggest concerns related to using AI tools within higher education included overreliance on the tools and a negative impact on the academic integrity. The reason why these were a big concern had to do with the fact that it could potentially hinder the student's development of critical thinking and ability to solve problems. The academic integrity was of concern due to the fact that AI could be misused to plagiarise or cheat during exams. While not completely identical to this thesis, the overreliance on an automation is one of the concerns as fully automating the process had to be avoided. Instead, a balance between human intervention and automation had to be established. The solution should not seek to replace the technicians but instead act as an assistant, as some of the repetitive tasks the technician has to conduct are some of the core concepts the technician must be aware of and can only learn through experience (Hasanein & Sobaih, 2023).

There have been similar studies conducted where ML had been experimented with to automate repetitive tasks and processes. Take for example a study conducted by Hedberg (2020) where the objective was to determine whether ML could be implemented as an assistant to provide support for accountants to make decisions on what accounts and cost centres to potentially use for invoicing. By doing so, the purpose was to identify the benefits, risks and evaluate the performance related to automating the invoicing process with ML. In another similar study conducted by Johansson (2022) the aim was to develop a system using ML to classify purchase invoice accounting lines into analytic accounts where text was pre-processed before comparing several different models to determine the final performance. While an invoice is not completely identical to a certificate, as a certificate contains much more data, both of these studies explored ways to improve a process by implementing ML.

1.1 Case Company

The case company the thesis was conducted for is a medium-sized company located in Europe. The company is a world-leading manufacturer of calibration equipment as well as calibration services. The case company has customers from different fields such as the medical, manufacturing, and oil industries, hence the case company follows several industry standards as well as serving and manufacturing all of the equipment in accredited laboratories.

1.2 Current Process

The current certificate verification process at the case company is that when a clients device comes in for service, it is calibrated in one of the systems by an authorized technician. After all of the calibrations are done, the measurement results are saved to the company's databases which are then used to generate the certificates with an in-house printing tool.

If the measurements results are within the specified limits, the device is working as expected and a certificate including all the functions that were calibrated and the measurement results can be generated for the device. If the results are outside of the limits, the device must be adjusted and re-calibrated with the system until the measurement results are within limits. When the technician has verified the certificate, it is then forwarded to the manager for their signature. This certificate acts as proof that the device is working properly, and both the technicians and the supervisor applies an electronic signature to the certificate after both parties has verified the contents.

1.3 Challenges in the Process

The main challenges with the current certificate verification process are, that the process is currently manually performed thus the process also requires the technician to be able to assure that the certificate is valid before forwarding it to their manager. This process relies on the technician and manager being able to spot if anything is incorrect and/or missing. For example, if the technician or the manager applies a signature to a certificate and notices a mistake afterwards, the certificate must be re-generated with the printing tool. While the tool is able to warn the user if there are missing functions or measurement types, it cannot detect certain faults such as variations in the data that requires years of experience the technicians have, incorrect names between the technician who generated and signed the certificate, missing functions that should always be included and a mismatch between the functions measured and those reported in the service report that is usually attached at the end of the certificate for devices that has been in for service. While the task of verifying the certificate is not a difficult, it can be time consuming and tedious, as there can be up to 10,000 documents per year. If a fully equipped device is served, the length of the certificate can reach up to 15 pages, meaning that by automating parts of the verification process, more consistent results as well as both time and cost reductions could potentially be achieved.

1.4 Research Objectives

The main goal of the project was to research whether parts of a currently manually performed task could be automated to improve the process. To be able to achieve this, the following objectives was set for the thesis:

- **Literature Overview:** To study and collect information about modern tools and potential ways a process such as the certificate verification process could be automated to make the process efficient and produce reliable results.

- **Current and Proposed Process Mapping:** To discuss and interview relative parties with experience about the current certificate verification process to get a general overview of the process to then being able to develop and suggest a solution.

- **Identify Challenges in the Process:** To identify and map the challenges in the current process that has to be eliminated or minimized to reduce the number of certificates that has to be re-made.

- **Comparing Different Methods:** To be able to propose a solution for the process, an understanding of what tools and methods are available that can aid in improving the process while also comparing the different methods to be able to determine in which scenario each tool and method works best.

- **Continues Feedback:** To discuss and update the product owner (PO) i.e., the individual responsible for the outcome of the project, about the current progress and potential solutions to improve the process.

1.5 Research Questions

Hence, from the current problems it can be concluded that the biggest reason faults occur in the verification process are due to the human errors. Based on the discussions and background information from the case company, two research questions (RQs) were developed to guide the thesis. These RQs were:

> ***RQ1:*** *How should the process be structured to allow the certificate verification to be automated?*
>
> ***RQ2:*** *How could the human intervention and automation be balanced in the certificate verification process?*

RQ2 identifies a potential research gap, i.e., how to balance human interaction with automating processes such as the certificate verification. Similar questions were discussed by Flemisch et al., (2012) where it was argued that the proper balance between humans and machines is not yet determined, especially with the rapid growth of AI, while the findings in another study indicated that AI tools had a negative impact on decision-making and led to laziness increasing (Ahmad et al., 2023).

This is relevant to today's world where technology is getting constantly more advanced and humans growing more and more reliant on it. Similarly, a service technician at the case company has to be able to manually validate a certificate without fully relying on any kind of automated services but instead use it as a tool in parallel during verification, as being able to manually verify a certificate is a crucial skill for a service technician at the case company.

1.6 Study Limitations

The thesis focused on exploring and experimenting with different types of tools such as ML to be able to verify that the measurement results in the certificates were correct. Due to the experimental nature of the project, no deployable solution to automate the process was included in this thesis, as such services would require a significant amount of time to test and validate before being able to confidentially deploy in a production environment. The automated solution would most likely always have to be operational as downtime could halt or slow down production, and as the certificate contains customer information, the solution would have to be secure. Thus, this thesis only laid the groundwork for a potential solution.

1.7 Study Structure

The thesis was divided into six parts, with the introduction being the first chapter. The second chapter focuses more on the theoretical aspects and gives a basic understanding of the contents that this thesis includes. The third chapter focuses on the research methodology used in the thesis, how the thesis was structured and how each phase was conducted. The fourth chapter focuses on the development of the automated solution itself, what tools were used and how the different tools were tested and evaluated. The fifth chapter discusses the results from the fourth chapter and presents a potential solution to automate the certificate verification process while the last chapter includes a retrospect of the thesis, what was learned, future research suggestions and challenges encountered during the thesis.

2 Theoretical Framework

According to Grant & Osanloo (2015) the theoretical framework is one of the most important components in the research process. The theoretical framework serves as the structure and supports the rationale, purpose and significance of the study while also supporting the problem statements and research questions. Without it, the vision of the study will not be obvious to the reader in the same sense that a building cannot be constructed without a blueprint.

Thus, this chapter contains an overview of relevant literature and earlier conducted studies, with a focus on task automation using modern tools. As for ML, this chapter aims to give a general understanding of the subject as ML is a wide concept.

2.1 Task Automation

As quickly mentioned in the introduction, automating repetitive tasks usually always has the same goal; reduce cost and time to complete a task. Task automation is however not a new concept but has existed for centuries. Take for example the windmill, where wind power is converted into rotational energy to mill grain or the printing press, developed to mass produced printed pages. Both of these serves as examples where labour-intensive repetitive tasks were reduced and productivity increased.

2.1.1 Modern Tools

One of the tools often used today to automate tasks is Robotic Process Automation (RPA) which is a tool that is used simulate the work performed by humans. For example, logging in, filling in boxes and copy data from a website. RPA can be used to simulate this behaviour by automating these kinds of repetitive tasks, allowing workers to focus on more important and/or complex tasks.

Generally, RPA can be considered as the umbrella term for tools that operate on user interfaces. Simply put, RPA is a bot that can include technologies such computer vision and ML to automate high-volume repetitive tasks following a rule-based approach i.e., the programmer has to define exactly what steps the bot must take such as *IF x, THEN DO y* (Casey, 2020).

The difference between RPA and a traditional automation process such as macros, is that RPA's core function is to identify elements instead of screen coordinates and RPA can interact with several applications simultaneously while macros usually can only interact with one at a time, resulting in RPA having a more intelligent and adaptive behaviour.

Ribeiro et al., (2021) conducted a study where they presented RPA tools combined with AI technologies to evaluate the feasibility of these tools for enterprise resource planning related processes. The aim of using these tools was to improve organizational processes associated with Industry 4.0 i.e., the fourth industrial revolution which implements smart and modern technologies to manufacturing such as AI, internet of things, and cloud technologies to increase the efficiency, reliability, and speed of the manufacturing process. The study concluded that most of the tools used for these processes combine AI functionality such as recognition, optimization, classification, and data extraction with RPA. However, they also concluded that most of these tools has an expensive licensing cost, with open-source tools still being under development which can be one of the factors why such tools would not be selected (Ribeiro et al., 2021).

In the case of this thesis, RPA could potentially be used to automate or semi-automate the process of filling in the required data to generate a certificate with the tool the technicians use to speed up the process. However, this would not improve the certificate verification process, as the certificate has to be verified after the certificate has been generated.

Additionally, selecting the right parameters when generating the certificate is an essential skill of the technicians, as there can be multiple calibrations performed on the same day in case the device had to be re-calibrated, hence RPA was not included as an alternative in this thesis. But as mentioned, RPA could potentially be used to semi-automate some of the parameters which usually always stay the same.

Computer vision is also a concept that has seen significant growth. Areas where computer vision has been used include surveillance cameras, autonomous cars, and facial recognition. Computer vision works similarly to how a human would view an object and learn from it, the difference being that computer vision systems usually require large quantities of data. For example, if the goal is to recognise if a car is a taxi or not, a vast number of images of taxis and normal cars would have to be collected and labelled to be able to train it. And as having to train a computer vision model would imply, computer vision is considered to be part of the field of AI (Islam Khan & Al-Habsi, 2020).

The technology relevant to this thesis however is Optical Character Recognition (OCR), which is encompassed in the wider field of computer vision. OCR is used to extract data from different types of documents, such as scanned documents, portable document formats (PDFs), or text from images with the purpose of converting it into digital text i.e., selectable, and editable text. The process of OCR starts of with scanning the text in an image. The image is then analysed by the OCR model where each individual character is identified, and the shapes and patterns of the characters are identified by utilizing pattern recognition algorithms. The identified characters are then compared to libraries containing known characters to determine what they are. It is also possible to apply OCR to handwritten text, however this would require more pre-processing as the quality of the handwritten text may not always be of the greatest quality and characters not easily recognizable compared to machine written text as the style of the characters are not consistent.

Lastly, the recognized text is converted into digital text by matching each of the characters with corresponding Unicode values, where Unicode refers to the standard used to assign a character to a unique code. For example, the character *b* would have the value *98.* An example of a use case for OCR would be to aid an accountant extracting certain elements from invoices, either scanned or handwritten, and forwarding the data to an Excel spreadsheet (Smith, 2023).

During the initial meeting of the project, the PO suggested exploring OCR as a potential solution to automate the certificate verification process as the data in the certificates could potentially be scanned with OCR and forwarded to an application that would perform the verification. Another point the PO made was that there was a lot of old scanned certificates that could be scanned with OCR to then be able to determine the aging of the devices. Hence, OCR was decided to be tested and evaluated during this thesis.

By now, an idea of what technology has seen rapid growth in nearly every field to automate tasks in the current age should have become clear. This would of course refer to AI technologies. In a study by Rayhan (2023), current trends within AI and automation integrations were analysed and the challenges and opportunities identified as AI and automation has the potential to revolutionize job roles and reshape the workforce. AI implementations can be seen in many businesses, as the use of intelligent chatbots and virtual assistants has become more common. Automation on the other hand has always been prevalent in the manufacturing sector such as assembly lines. Many of the manual tasks has been replaced by robotic arms and autonomous vehicles which has streamlined operations and saved on cost. The combination of AI and automation can also be seen within robotics as smart robots has the capability to adapt to certain situations and circumstances, perform complex tasks and collaborate with human operators, which has seen an improvement in quality control, process optimization and inventory management.

However, the integration of AI and automation is not without its own challenges as earlier mentioned. Social challenges that have been a concern related to this include where issues such as data privacy and algorithmic bias has been some of the concerns, hence ethical frameworks to ensure transparency and fairness has to be set in place as the use of AI only becomes more common. Another concern relates to job displacement, as automated solutions for repetitive tasks replaces manual labour resulting in potential unemployment. However, this can also be an opportunity, as it would allow for a shift of skills in the workforce, allowing workers to learn new skills and improve instead of having to perform the same tasks over and over. Hence, a balance between human labour and AI is crucial to improve both the workforce and the economy (Rayhan, 2023).

Similar to what was discussed in the study by Rayhan (2023), this thesis explores different ways AI could be implemented to help improve the certificate verification process without negatively affecting the skills or quality of the technician's work. However, this thesis focused on ML, which is a subset of AI. The difference between these two being that AI has the ability to mimic human cognitive functions but only according to what it has been taught while ML refers to the concept of allowing machines to learn from data and make decisions autonomously.

2.1.2 Similar Studies

As mentioned in the introduction, similar case studies and projects have been conducted where it was investigated if ML could be used to improve processes by automating certain phases in the process. In the earlier study mentioned conducted by (Hedberg, 2020), the ease of use and the trust of the automated process would increase if the ML model could tell the user why the suggestion was made and why it was the correct decision. Hedberg also presented different levels of automation, ranging from level 1 (no assistance) to level 10 (no human intervention). In Hedberg's study, the level of automation chosen was level 4, i.e., the system suggests an alternative. In this thesis case, this was the level that was focused on, as the system should only suggest what the technician should take a look at and not correct it by itself. Hedberg could lastly conclude, that one

of the risk factors with implementing a ML solution to a process is misuse, which Hedberg (2020) defined as *Users assuming that the decision is correct, even if it's wrong*. In the case of the certificate verification process, the results from the automated verification solution should not be taken for granted but should merely work as an assistant and suggest what the technician should verify in case irregularities in the certificate are detected.

Another study, while not focusing on automating a process, conducted by Brasjö & Lindovsky, (2019) explored ways to provide guidance for ML projects from a project management perspective. The findings in this study were valuable, as it provided an idea of how ML projects should be planned and managed, which can be difficult due to the complexity of said projects. An observation made by Brasjö & Lindovsky (2019) was that most ML projects used sprint concepts from Scrum but did not strictly follow the process due to the fact that using these concepts within ML projects is still a learning process, as the rules needed to be broken or modified when required. One example of rule breaking is the immediate internal sharing of insights which goes against the Scrum methodology. The biggest challenge with ML implementations seems to be due to the facts that companies miss out on establishing a data culture to strategically collect data before attempting to implement ML to automate processes or increase customer experiences. As is often discussed, arguably the most important part of any ML project is the data collection and pre-processing phases as the ML model will only be as good as the data it has trained on. From a project management perspective, data should be considered as the main stakeholder and be the first factor to be evaluated during risk assessment. Another problem with ML projects is the unrealistic expectations made, most likely due to hype related to AI technologies. The role of the project manager in this case should be to make it clear to the executives what can realistically be achieved and what cannot, as ML is not the solution to every problem even if it's a powerful tool. This is an excellent point, as most of the certificate verification process would most likely not be possible or worth to automate using ML, hence the available data and resources needs to be evaluated and experimentations conducted before a suggestion of an automated solution can be

made. Furthermore, one of the procurement options Brasjö & Lindovsky, (2019) discussed was the in-house development option. The most important skills within this option were deemed to be the data pre-processing, business understanding and the ability to communicate the complex ML problems to managers and those unfamiliar with the technology, which was an important finding, as the results from this thesis would be used to develop and implement an in-house solution. Hence, discussions and potential demos must be conducted with the PO to achieve the best results and ensure that the PO is aware of how the technology would work and its limitations.

2.2 Machine Learning

Machine Learning (ML) can be defined as *computational methods using experience to improve performance or to make accurate predictions.* Experience in this context refers to data, such as labelled training sets or historical data which the ML can analyse to *learn* using different types of algorithms to perform predictions on the data (Mehryar Mohri et al., 2018, p. 18). These algorithms are also often referred to as *models* and a model is usually chosen depending on the data, the problem to be solved and the desired result. According to Nichols et al., (2019) when working with large datasets in the range of 10^6 unique data points, more complex deep learning (DL) models is likely required while a limited amount of data points work better with traditional techniques such as linear regression or decision-tree methods. In other words, the larger the sample, the easier the task. However, the sample size is not the only thing deciding how difficult the task may be, as other factors such as quality of the labels assigned to the samples as there is a chance that not all labels are correct (Mehryar Mohri et al., 2018, p. 18).

There are no concrete rules that determine what can and what can not be solved by ML as the practical application of ML is constantly expanding, but some examples according to (Mehryar Mohri et al., 2018, p. 19) include text or document classification, natural language processing and computer vision. Text classification can be used to determine whether a website is safe or not, either due to explicit content, spam detection or/and malware detection and natural language processing for part-of-speech tagging, where

each word in a sentence is labelled according to its appropriate part of speech, such as verbs, pronouns, and nouns. This tagging is often used in machine translation and question answering products e.g., ChatGPT.

2.2.1 Algorithms

There are multiple different types of ML algorithms and different algorithms excel at different tasks, hence it's important to understand what kind of data is available and what the desired outcome is. For example, three main types of algorithms used in ML are classification, regression, and clustering algorithms. (Mehryar Mohri et al., 2018, p. 20). This thesis, however, focused mostly on classification algorithms; thus, the other two algorithms were only briefly summarized.

Regression can be thought as the problem of predicting a real value for objects (Mehryar Mohri et al., 2018, p. 20). When compering classification to regression problems, the difference between these two problems is, that in a classification analysis the goal is to predict unique labels for new unseen data which has been learned from a dataset containing similar labels and data while regression helps determine the relationships among variables by estimating how each variable affect each other. In a ML context, regression and classification models are usually a supervised technique and can for example be used for sales forecasting, maintenance prediction and financial performance analyses.

Compared to classification and regression, clustering on the other hand is an unsupervised technique used to identify relations and group data from large datasets without expecting a certain outcome. This technique can be used to discover patterns or trends such as specific types of consumers depending on their behaviour. Thus, clustering algorithms is often applied in areas such e-commerce, cybersecurity, health analytics and behavioural analytics (Sarker, 2021).

When data from a dataset must be categorized, a classification analysis can be developed. The basic functionality of a classification analysis is to map a function from input variables to output variables as categories or labels. Three common classification problems include binary classifications, which involve two class labels “true” or “false” i.e., a normal and an abnormal state, multi-class classifications which is similar to binary classification but where there can be more than just two states or labels, and multi-label classification, where a sample can have multiple labels. Algorithms used in multi-label classification can predict various mutually non-exclusive labels unlike traditional classification tasks where the labels are always mutually exclusive. To give an example of a multi-label classification, imagine a news article that can belong to multiple topics such as politics, entertainment, and sports (Sarker, 2021).

Algorithm	Use Cases	Strength	Weaknesses
Classification (Black et al., 2023)	Profiling patients' disease risk, clinical support, fraud detection	Can outperform traditional statistical models when it comes to predictive performance and be more flexible when it comes to unstructured data.	Requires larger datasets compared to traditional statistical models e.g., while a ML model might require 200 events per candidate, a statistical model only requires 20. More often prone to bias due to errors in model design or biased data.
Regression (Kurama, 2023)	Temperature prediction, stock market prediction	Able to establish and determine relationships among different variables. Linear models can be easily updated with new data.	Poor performance when there are non-linear relationships in the dataset as the model is not as flexible in capturing complex patterns requiring additional polynomials or interaction terms which can be difficult to get correct and high variance can lead to poor prediction performance.
Clustering (Aravind, 2023; Pitafi et al., 2023)	Finding patterns and groups within areas such as marketing, politics, ecology, and genetics	Dataset does not have to be labeled and give an understanding of a dataset as clustering algorithms can group the observations in the dataset.	In cluster analysis, determining the correct number of clusters beforehand is difficult due to factors such as no prior experience with the data i.e., not knowing what kind of groups to be expected and the varying sizes, forms, and densities of groups inside a dataset also make it difficult to draw conclusions.

Figure 1: General Use Cases, Strengths, and Weaknesses of Each Machine Learning Algorithm.

Different types of classification algorithms were evaluated in this thesis. The classification algorithms chosen were the commonly used Support Vector Classifier (SVC), K-Neighbors Classifier (KNC), Random Forest Classifier (RFC), Extra Trees Classifier (ETC), Gradient Boosting Classifier (GBC), Logistic Regression (LR) and Gaussian Naïve Bayes (GNB).

Support Vector Machine

SVC implements the Support Vector Machine algorithm where the goal is to find the hyperplane i.e., the decision boundary that separates the observations or samples in a dataset into specific classes. The desired outcome is to identify and select a hyperplane where the margin between the hyperplane and the training dataset is the greatest thus increasing the possibility of new data being correctly classified. However, if the data is scrambled and non-separable, the view must be changed from 2D to 3D to be able to classify the data. This method is often referred to as *kerneling*. SVC refer to the fact that the algorithm is specifically used for classification problems, such as image recognition. However, SVC also has its own weaknesses, such as excessive computational cost if the dataset is large due to the fact that the kernel matrix grows in a quadratic form with the size of the dataset, is better suited at solving binary-classification problems than multi-class problems and struggles with imbalanced datasets where the minority class has to be correctly classified compared to other algorithms (Cervantes et al., 2020).

Random Forest

Random forest is an ensemble learning technique which can be used for both classification and regression problems, where in this case RFC focuses solely on classification problems. Ensemble methods refer to the fact that it is a combination of multiple ML models to solve a problem. In this case, RFC uses a combination of the bagging sample approach and random feature selection to construct decision trees, which as the name implies resembles a tree where it starts of at a root node and splits into several branches (internal nodes) which refers to a decision point where the algorithm evaluates the best split to ensure the separation between the different classes according to the different features in the dataset. For example, each internal node might check for a specific condition such as *Is the current feature greater than 15?* If it is, the data is again split into two or more branches and if it isn't, an end point has been reached (leaf node). Leaf nodes cannot be split, and represent the decision made by the algorithm. In this case, as the aim is to answer a classification problem, the leaf node would correspond to a specific class (Fawagreh et al., 2014).

The algorithm constructs a forest consisting of several decision trees during training which consists of different subsets of the training dataset. Lastly, when the results have been obtained from all of the constructed trees, a majority vote decides the final prediction. The randomness in the algorithm comes from the fact that the subsets from the training data is randomly selected as well as the randomly selected features during the splitting. The random forest algorithm is a robust algorithm that can handle noisy data with outliers but can suffer from overfitting in case the number of decision trees are too high, or the trees has branched too deep (Schonlau & Zou, 2020).

Extra Trees

The extremely randomized trees or extra trees algorithm used in ETC is also an ensemble learning technique and works similarly to the random forest algorithm, the differences being that the extra trees algorithm constructs trees using every single observation in the training dataset. The features are again randomly selected at each split, but unlike the random tree algorithm, it will select the split points at random, hence the name. By using every single observation, the risk of bias is reduced while also reducing variance as certain features and patterns do not influence the final prediction as much as it might do in the RFC (Brownlee, 2021b).

Gradient Boosting

Boosting has been considered one of the most influential ideas introduces within ML. Unlike random forest, where the predictive model is generated by directly averaging the predictions, boosting uses a stagewise strategy where models are added in a sequence until a strong learner has been generated. Hence, the gradient boosting algorithm is also classified as an ensemble learning technique as it combines the prediction of multiple weaker learners to create a single strong learner which is able to give accurate predictions. The performance is improved by iteratively evaluating and gradually reducing the residuals i.e., errors of the previous models, usually decision trees, until an optimized model with enhanced accuracy is generated and can be used to solve both classification and regression problems.

The algorithm works well with dataset that includes heterogenous features i.e., a combination of numerical and categorial features such as a person's age and nationality and is often able to give the most accurate predictions compared to other models. However, the model might suffer from overfitting as it may be influenced by outliers as it tries to minimize errors and can be computationally expensive as the number of trees required increases (Guillen et al., 2022).

Gaussian Naïve Bayes

GBC is based on bayes theorem, a fundamental concept within probability theory where the probability of an event can be determined given the probability of an event that had earlier occurred. The algorithm assumes each continuous feature follows a normal distribution and that each feature is independent from each other, which can work well in practice but be different in real-world situations. GBC is a simple and efficient algorithm while also having a low variance as it ignores the feature interactions. However, the trade-off with this is that bias is introduced in case the features are dependant and may result in patterns being undetected and in datasets where the features are dependant, the performance of the GNB may most likely suffer (Anand et al., 2022).

Logistic Regression

Often used to solve binary-classification problems but can be extended to solve multiclass-classification problems i.e., multinomial logistic regression. Compared to linear regression, which predicts a continuous outcome such as house or stock prices, logistic regression predicts the probabilities for a categorial outcome such as true/false or yes/no. Logistic regression helps understand how each predictor variable influences an outcome from occurring while also simultaneously assessing the other predictor variables when estimating the probability of the said outcome occurring i.e., the odds ratio. Logistic regression is a simple algorithm which is easy to implement and interpret but struggles with high-dimensional datasets and makes assumptions of linearity between dependent and independent variables (Rymarczyk et al., 2019).

K-Nearest Neighbors

KNC, which implements the K-Nearest Neighbors algorithm, is a non-generalizing or "lazy" learning algorithm which does not construct a general internal model but instead stores all instances in an *n*-dimensional space that corresponds to the training data. The main idea is the assumption that similar points can be found near each other, and this can be calculated using different functions such as the Euclidean or the Manhattan distance function. While KNC is easy to implement and requires only a few hyperparameters, the biggest issue lies in choosing the optimal number *k* neighbors, as too few can have a larger influence on the result and too many increases the computational cost as it does not scale well. Similar to Random Forest and Gradient Boosting, K-Nearest Neighbors can also be used to solve regression problems (Sarker, 2021).

2.2.2 Types

The algorithms themselves are also often categorized as each algorithm each excels at solving different types of problems. Some of the most common ML types will be discussed next.

Supervised

In supervised learning, the data used has been labelled and the desired outcome is known. A supervised ML model is trained with labelled input and output data until it is able to present accurate predictions with data it has not seen before but is comparable to what it has been trained with, hence the term *supervised learning.*

Classification and regression fall into the supervised learning category, as trained classification algorithms are able to categorize new unseen data into correctly, such as spam mail, image recognition and customer churn prediction (Shahzadi, 2023). Similarly, regression algorithms can be used to forecast sales based on historical data and current market trends (Madhumala, 2020).

Unsupervised

Unsupervised learning on the other hand is the opposite of supervised learning, as the data used is unlabelled. While labelled data used to train supervised models to map inputs to output is not possible with unsupervised models, the data can still be mapped to find unknown relations and patterns.

As there are no labels, the results cannot be evaluated similarly to supervised results and incidentally, it also means that classification and regression problems cannot be solved with unsupervised learning. Instead, clustering algorithms are more widely used in unsupervised models (Jones, 2017). Clustering algorithms can for example be used for customer segmentation to identify similar purchasing habits and traits, document classification and clustering observed cases of natural disasters to identify areas prone to them (Mishra, 2017).

Semi-Supervised

Semi-supervised is a combination of supervised and unsupervised learning. For example, in a classification problem, additional unlabelled datapoints can be used to improve and aid the classification process while in a clustering problem, the model can benefit from knowing which specific data points belong to which class or category (van Engelen et al., 2020). In other words, semi-supervised models use a small amount of labelled data to train on after which it aims to improve the learning procedure when predicting on a larger unlabelled dataset (Ligthart et al., 2021).

However, according to van Engelen et al., (2020), it has proven to be difficult or even impossible to do so. One reason for this is, that unlabelled data is only useful if the data contains relevant information which enables label prediction that is not contained in the labelled data or cannot easily be extracted.

To be able to successfully apply a semi-supervised model, the algorithm used must be able to extract the earlier mentioned information but unfortunately, it is difficult to define the conditions under which semi-supervised algorithms may be able to do so and evaluate whether the model satisfies the conditions. What's more, there is a possibility that the performance degrades due to the introduction of unlabelled data. Hence, semi-supervised learning should be treated as a possible direction in the learning algorithm process instead of only using semi-supervised learning algorithms.

Reinforcement Learning

Reinforcement learning is a type where desired behaviours and outcomes are rewarded and undesired ones punished. An agent i.e., the model being trained, examines the environment and its condition, and accordingly chooses appropriate actions through trial and error. Reinforcement learning includes two entities that must be balanced, exploration and exploitation. Exploitation refers to when the agent attempts to maximize the reward depending on previous routes while expedition refers to when the agent always tries out different ways to reach the destination. The advantage of these types of models is, that they do not need as large of a dataset as other models, as it learns through trial and error (Mauro et al., 2023).

There are multiple types of reinforcement learning models, two of which are the value-based and the policy-based models. In value-based methods, to extract policies that will be used for deciding actions, are obtained by iteratively optimising the value function. Algorithms that can be utilized for this are Q-Learning and State-Action-Reward-State-Action. On the other hand, policy-gradient methods, no value functions are computed but instead the policies are directly optimized from the expected rewards (Warnell et al., 2021). Reinforcement learning can be applied in various fields, such as self-driving cars for controller optimization, motion planning and scenario-based learning policies, and within healthcare, it can be used to recommend treatment (Mwiti, 2023).

Learning Type	Features	Use Cases	Strengths	Weaknesses
Supervised (Crisci et al., 2012)	Uses labelled data to make predictions according to what is expected.	Spam detection, sales forecast, image recognition	Useful in a wide range of fields as it is possible to make accurate predictions and to generalize on patterns.	Requires sufficient amount of labelled and structured data to be able to accurately make predictions which can be time consuming and/or expensive to acquire.
Unsupervised (Naeem et al., 2023)	Can find patterns and group data previously unknown	Social network analysis, customer segmentation	Uses unlabelled data to make the predictions and groupings, useful when conducting an exploratory analysis	Results can be difficult to evaluate as results might not be accurate
Reinforcement (AlMahamid & Grolinger, 2022)	Decides the next action based on the policies made through trial and error	Game AI, robot navigation, self-driving cars	Learns from doing, has the capability of improving as time passes	Requires a logical reward and punishment system, requires more computational resources and time to achieve results.
Semi-Supervised (van Engelen & Hoos, 2020)	Uses both labelled and unlabelled data	Disease progress prediction, medical image analysis, web content classification	Useful when the amount of unlabelled data is abundant, but the labelled data is not	Unlabelled data potentially degrades the performance of the models.

Figure 2: Summary of Different Machine Learning Types and Their Characteristics

2.3 CRISP-DM

Cross-Industry Standard Process for Data Mining (CRISP-DM) is a process model that provides an overview of data mining projects lifecycles (Wijaya, 2021). According to surveys, CRISP-DM is considered the de facto standard process used for data mining and knowledge discovery projects (Ayele, 2020).

The CRISP-DM framework is over two decades old as of this day, first developed by a consortium of companies including Daimler-Chrysler, Teradata and NCR Corporation and was founded by the European Union between 1996 and 2000 (Nadali et al., 2011). The aim was to create an industry standard framework for data mining projects by identifying good practices and common mistakes (Martinez-Plumed et al., 2021). Two decades later, CRISP-DM is still widely used within many different fields such as healthcare, signal processing and education.

Other similar processes usually use techniques such as similarity measure, association rules and co-occurrence of term analyses while also being depicted as simple workflows and illustrations of processes. CRISP-DM on the other hand is independent of data mining technologies and industry sectors, meaning it can be adapted to any given scenario. Other benefits with using the CRISP-DM methodology are the reduction of cost and time, minimal knowledge requirements for the data mining project, standardization, innovative activity encouragement and components such as expediting training, knowledge transfer, documentation and best practices are all included in the CRISP-DM framework (Ayele, 2020).

The term *data mining* is more commonly today known as *data science* in the context of knowledge discovery and can be defined as *the automated or convenient extraction of patterns representing knowledge stored in large databases or other large data repositories.* Compared to statistics, where data is collected with the purpose of testing specific hypothesises or estimating parameters, data mining usually focuses on collected historical data that was not necessarily meant to be analysed, but rather collected as a by-

product. Another difference compared to statistical methods, is that data mining focuses on data-driven discoveries i.e., instead of starting out with a hypothesis against the data, the data generates the hypothesis (Calders & Custers, 2013).

According to Martinez-Plumed et al., (2021) there appears to be two different kinds of perspectives of the word *data science* which are: 1) *the science of data* and 2) *applying scientific methods to data.* The former term refers to studying data in all it's different forms together and manipulating, analysing and visualize it by using different methods and algorithms, which is often used within academia while the latter term spans both academia and industry and refers to extracting knowledge from the data by using methods such as statistical hypothesis testing or ML i.e., data is used to develop models, design artefacts, and increase the overall understanding of corresponding subjects. If the two perspectives were to be labelled, the former would be considered *theoretical data* while the latter would be *applied data science.* Keeping that in mind, when the CRSIP-DM methodology was developed, data mining was defined as being goal-oriented and concentrated on the process while data science today is defined as data-oriented and exploratory. Thus, when following the traditional CRISP-DM methodology, one should have a clear business goal and the data should already be available and ready to be processed.

2.3.1 Level Breakdown

According to Wirth & Hipp, (2000), the CRISP-DM model consists of four levels of abstraction, in the form of different types of tasks, within a hierarchical process model. These levels are referred to as phases, generic tasks, specialised tasks, and process instances.

Phases

The top level of the hierarchy consists of the so-called *phases* which the data mining process is organized into. Each phase contains several *generic tasks* (Wirth & Hipp, 2000).

Generic Tasks

The second level consists of generic tasks. These tasks are referred to as generic as they are expected to cover all possible situations within data mining. Furthermore, they are also expected to be as *complete* and *stable* as possible. In this case, complete refers to covering the whole data mining process and all the possible data mining applications and stable refers to the fact that the tasks should be compatible with yet unforeseen developments such as new modelling techniques (Clinton et al., 1999).

Specialised Tasks

The third level is used to describe how actions in the generic tasks in certain situations. For example, in the second layer, there might be a task defined as *remove zero values.* For cases where datasets containing only integers, this might refer to the integer 0 while in categorial datasets, this might refer to blank cells (Wirth & Hipp, 2000).

Process Instances

The fourth level consists of records related to the actions, decisions, and results from the data mining procedures. Process instances represents what actually happened rather than what happens in general and are organized according to the tasks defined at the higher levels (Anirudh, 2019; Clinton et al., 1999).

What also is worth keeping in mind, is that the steps of the phases and tasks are the idealized sequence of events, meaning that the tasks can be performed in a different order instead of a linear order. Sometimes backtracking and repeating certain actions might be necessary. The CRISP-DM framework does not attempt to capture every single possible route, as it would require an overly complex process model with very low expected benefits (Wirth & Hipp, 2000).

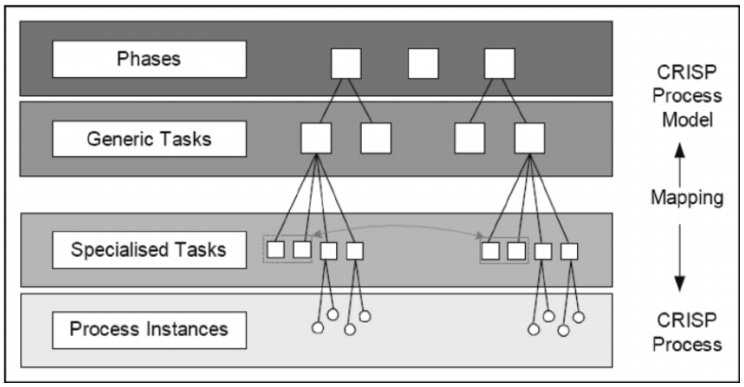


Figure 3: CRISP-DM Four Level Breakdown (Alnoukari & El Sheikh, 2011)

2.3.2 Process Model

The main purpose of the CRISP-DM process model is to provide an overview of all the different stages of a data mining project i.e., its lifecycle. Each phase of the process model contains the tasks and their relationships. The relationships can vary depending on the goals of the project, background, the interest of the developer and the data itself, meaning identifying every single relationship is difficult or even impossible. There are six different phases in the process model, referred to as business understanding, data understanding, data preparation, modelling, evaluation, and deployment.

The phases do not follow a strict sequence, as moving back and forth between the phases is often required. The next task to be performed always depends on the outcome from the current phase in progress. The arrows shown in Figure 4 only indicate the most frequent and important dependencies between the different phases (Wirth & Hipp, 2000).

The outer circle of the process model represents the cyclic nature of data mining, as data mining projects are not over once deployed since discoveries from the process and after deployment might spark new business questions (Clinton et al., 1999). Understanding the idea of each phase is important to being able to fully utilize the CRISP-DM framework.

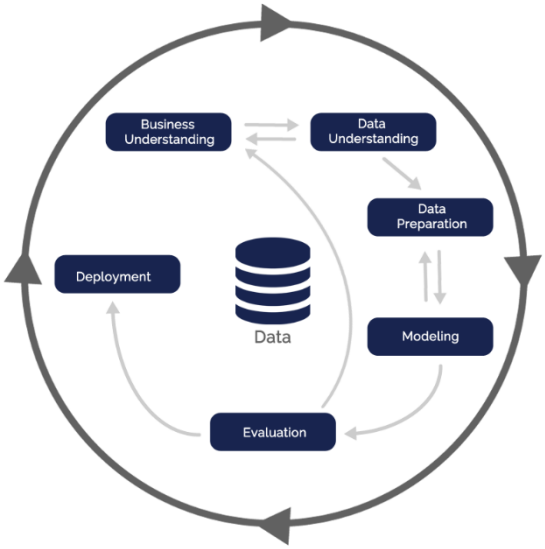


Figure 4: The Six Phases of the CRISP-DM Framework (Anirudh, 2019)

Business Understanding

In the initial phase, the focus is on understanding the objectives and requirements of the project from a business perspective. After this, the information is converted into a data mining problem definition and a project plan to achieve the objectives. It is important that this phase is carefully completed, as the base of the whole project will be determined at this stage, akin to the fundamentals of project management activities (Tripathi et al., 2021).

Tasks that are often included in this phase are understanding what the client is trying to achieve, business success criteria, determining available resources, project requirements, risk assessment, cost-benefit analysis and defining a detailed plan for each phase of the project (Nadali et al., 2011; Saltz, 2021).

Data Understanding

This phase builds further upon the initial phase by identifying, collecting, and analysing the datasets that will be used or can aid in the project. According to Saltz (2021), this phase includes four tasks. The first tasks are to collect the initial data that will be used for further analysis. Then, the data should be examined and the structure of it described.

After doing so, further analysis can be done by querying the data i.e., requesting certain information from the dataset, visualising the data and by identifying relations in the data used. Lastly, the quality of the collected data should be assessed, and any quality issues documented (Saltz, 2021).

Data Preparation

As the name implies, this phase focusing on preparing the collected data to be used in the project, which can be referred to as *data munging* where raw data is initially refined and manipulated into formats according to the specifications of the project.

In data preparation, the first task is to determine which datasets to use and document reasons why data was or was not selected. When the dataset has been selected, the next step is to clean it, e.g., remove duplicates, remove errors and imputing the data, which can be a lengthy process and should be done carefully, as bad data will result in bad results. After having cleaned the data, new attributes can be obtained from it and new datasets can also be created by combining data from different sources. For example, the ranking of students according to their results can be determined depending on their points and grade. Data can also be re-formatted if needed. For example, string values can be converted to numerical values to perform mathematical operations. It is important that any changes to datasets are documented (Nadali et al., 2011; Saltz, 2021).

Modelling

Having laid out the groundwork, the modelling phase focuses on developing the model itself. A specific model does not have to be selected immediately. Instead, various models can be assessed using different modelling techniques can be utilized to determine which model(s) performed the best. According to Saltz (2021), the CRISP-DM guidelines suggest that the model building process should be iterated and assessed until the *best* model(s) have been determined, while in reality, the model selected might not have been the *best* but instead *good enough.*

The data should be separated into training, test and validation sets before starting the model building process. Lastly, based on factors such as success criteria, experience and test design, the performance can then be assessed (Martinez-Plumed et al., 2021; Saltz, 2021)

For example, two model selection techniques that can be used are probabilistic measures and resampling methods, where the former technique involves analytically scoring a potential model depending on its performance on training data and the complexity of the model itself while the latter aims to estimate the performance of the model on unseen data or out-of-sample data (Brownlee, 2019).

Evaluation

After potential models have been selected and tested, the evaluation phase focuses on how well the performance of the models meets the business requirements and which model(s) the project should proceed with. The tested models should also be seen in retrospect to verify that nothing was overlooked and that the testing was properly performed after which the results can be summarized and corrected if any deviations are detected. Before continuing on with the project, potential next steps to continue should be listed with the aim of meeting the business criteria (Saltz, 2021).

Deployment

The last phase of the project is deploying the developed model. Deploying the model depends on the how complex the project is, as the deployment can either only require a transfer to production document or implementing a real-time predictive model live. This phase should include plans of how the model(s) will be deployed and how the maintenance and monitoring of the model when operational will be conducted. Lastly, a master document of the project should be made and if needed, a presentation can be held. A retrospect of the project can also be conducted at this point to determine what went well and what did not, challenges during the project and potential future improvements (Saltz, 2021).

3 Research Methodology

As earlier mentioned, CRISP-DM is a proven framework for data mining projects, hence the framework was chosen to guide this project. This chapter focused on how CRISP-DM was used to structure the project and how it was modified to better suit the project.

As this project only had a single developer and one PO, it did not make sense to make it any more complex by trying to implement a modernize version of the framework for this project. The development part of the project i.e., the data preparation, modelling, and evaluation, was however part of a product increment, i.e., all previous and current tasks completed from the sprints. Thus, during weekly meetings, the progress of the project was reported and discussed.

In this project, the standard version of the CRISP-DM was followed, however some small modifications were made. First, the first two phases were combined and the data collection and analytical tasks of the second phase moved to the third phase. The reason for this was, that the certificates contained most of the data that would be required for the project, hence, to understand the business, the data had to be understood and vice versa.

The other modification of the framework was to not include the last phase, i.e., deployment. Due to the nature of the industry the case company was in, the certificates had to be verified carefully, as it was the proof that the device was working properly. Automating the certificate verification process cannot be deployed into production without sufficient testing, documentation, and validation, which could not be realistically included in this project's deadline. Hence, this project focused more on experimenting and evaluating different methods and tools to produce a proof of concept (POC) which in turn would provide a potential suggestion which could be further developed until a production ready solution was achieved.

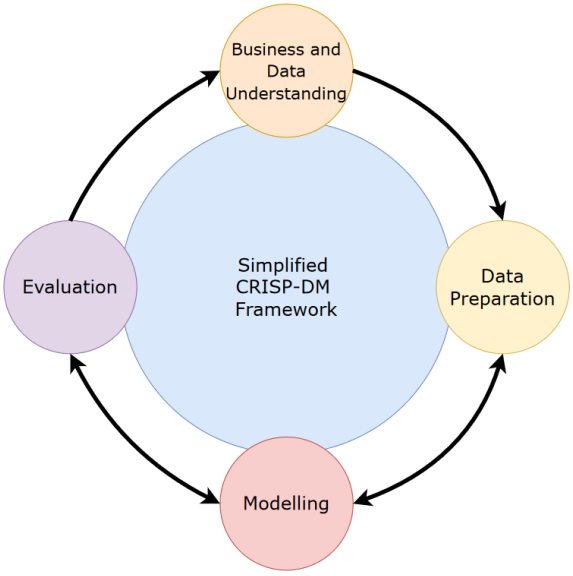


Figure 5: Diagram of the Simplified Framework Used to Structure the Project.

3.1 Business and Data Understanding

The qualitative data collection method used in this project which also allowed to get a better understanding about the certificate verification process and its data, was a focus group interview in which the Chief of the Laboratory (COL) as well as four service technicians participated. The sample size was quite small due to the fact that this project focused only on the case company's service department instead of also including the production department, which was larger, as the service certificates contained much more information and required more time to verify compared to the production certificates. Another factor that contributed to the small sample size was due to the fact that the number of service technicians with experience and the authority to generate and sign certificates was limited. By conducting a focus group with both the technicians and the manager, it was possible to hear both perspectives of the verification process and to pinpoint where automation could improve the process. The focus group interview was built around seven semi-structured questions to allow more open discussions with the participants. According to Kontio et al., (2004) the focus groups method is low cost, easy to conduct and can provide empirical experience quickly, hence it was chosen as the primary method for the qualitative data collection.

How does the current verification process work?

The technician uses the certificate printing tool (CPT) where input parameters such as the technicians name, the serial number of the device as well as which measurement results to include in the certificate. After this, a draft of the certificate in the form of a PDF is created which the technician then manually verifies before uploading it to the certificate server from which one of the managers, such as the COL, takes over. The manager then repeats the manual process of verifying the certificate. If everything is correct, the manager applies his or her electronic signature to the certificate and returns it to the certificate server. If something is missing or incorrect, the technician would be informed to re-check the certificate and correct the errors that were found. Lastly, when the certificate has been verified and everything is as it should be, the technician also applies his or her electronic signature to the certificate and the certificate is then sent to the customer together with the device.

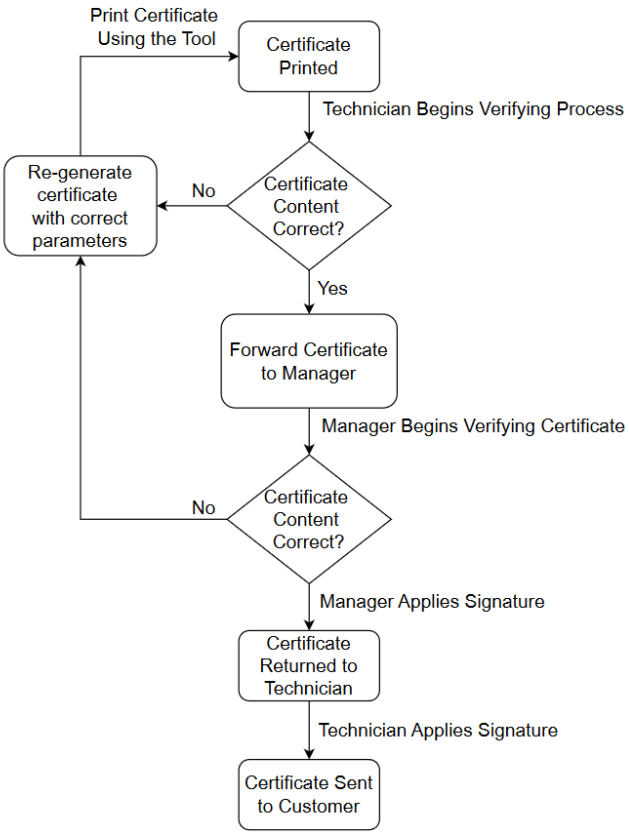


Figure 6: Flow of the Current Certificate Verification Process.

What challenges are faced in the current validation process?

The whole process of creating a certificate takes 10-20 minutes depending on the length of the certificate, including the verification process of the certificate draft, which only takes a few minutes to perform. The difference between the workload of the technicians and the managers are, that while it does not take long for the technicians to verify the certificate, the managers must also manually verify every single certificate sent by all the technicians before applying a signature. The COL explained that to manually validate 50 certificates, it could take up to 2 hours to ensure the certificates are correct. This process, while not complex, is very time-consuming for the managers who also has other tasks that require attention, leading often to overtime.

How often are faults detected in the certificates?

The faults would always vary, depending on the device and the number of measurement results that had to be included in the certificate. There was no concrete number of faulty certificates. However, the COL estimated that on average, 2 out of 10 i.e., 20% of the certificates had to be corrected.

How do these faults impact the overall process?

According to the COL, while it does not take a long time to correct the certificate, the faults should be spotted the first time the certificate generated as it forces the COL to spend unnecessary time on doing the same verification the technicians does. The technicians agreed that it's mostly frustrating when the wrong type of data has been mistakenly used or data forgotten to be included in the certificates and suggested that the faults should be caught much earlier.

Why do you think automation of the certificate verification process is needed?

The biggest reasons why the process should be automated was to reduce the number of unnecessary faults that could occur, to get consistent results and to reduce the workload of the managers. As mentioned earlier, the verification process was not difficult to conduct, but tedious and repetitive.

Do you have any concerns about automating the certificate verification process?

One of the concerns regarding automating the process was that an essential skill of the technician is being able to spot if there are incorrect readings in the measurement results and understand why it is incorrect. Hence, if the process is automated, it must be implemented in a way that does not negatively impact the skills of the technicians or remove the need of being able to verify the certificate results manually. Another concern was if the results were reliable, which meant that how and what results the model predicted had to be made clear.

How and where should the automated certificate verification be implemented?

The process should ideally be faster or as fast as when manually verifying. Other suggestions included that the automated process should be implemented in a way that, when the technician prints the certificate, it should at that moment begin verifying the certificate. The reason for this is so the technician doesn't have to generate a certificate, manually verify it, upload it to the server and then suddenly get a notification from the verification service that the certificate is incorrect, in case the service would send the verification results through email after the service has finished verifying the certificate.

As for the data understanding, the data on the certificates had to be understood and the potential sources of the data identified. The case company had three sources where the data could be extracted from, i.e., digital copies of the certificates in the form of PDF's, the object model i.e., the collection and structure of objects and classes that is used when generating a certificate with the CPT and the databases which the CPT used to gather data and generate the certificates. The certificates contained information such as device information, customer information, measurements results and signatures.

The COL provided a list with common faults found in the certificates that had to be checked. These included checking so all of the required measurement functions were included in the certificate, verifying the measurement types, the name of the technician who created the certificate and who signed it not being the same and the service report being incorrect. The service report was only present in the service certificates and included for devices after they had been tested and validated by a service technician. If a measurement function had to be adjusted to stay within the specifications, the adjusted function had to be included in the service report while the results after adjustment also had to be indicated as *As Left* results. If a device was not adjusted, the measurement results were indicated as *As Found* results. Additionally, every measurement result with an *As Left* results had to have a corresponding *As Found* result as it had to be clearly indicated that the function had been adjusted and why it was adjusted, hence both results needed to be included.

3.2 Data Preparation

In the Data Preparation phase, the focus was on how to extract the data from the sources and how to manipulate the data. The digital copies of the certificates also differed. In 2022, electronic signatures were implemented for the certificates while the copies of the certificates saved before 2022 were first printed, verified page for page, signed and then scanned to get a digital copy. The biggest difference between the electronically signed certificates and the scanned copies were that the newer version had selectable text while the scanned version did not and varied in quality, i.e., blurring, rotation, boldness, and saturation. Due to this, the data couldn’t be extracted in the same way from the certificates and data extraction methods depending on the type of document had to be developed.

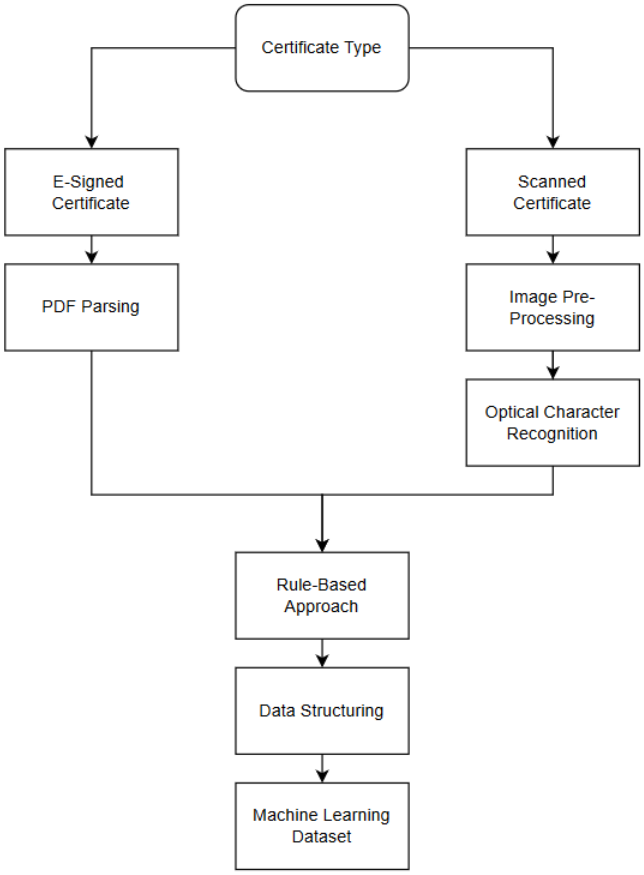


Figure 7: PDF Extraction Flowchart

The figure above is a simple flowchart of how the data could be extracted from the certificates. The *E-signed Certificate* refers to the newer certificates with selectable text that were signed with a tool to get an electronic signature while the latter were the scanned copies. For the certificates with electronic signatures, PDF parsing tools could potentially be used extract all text found in the document and apply a rule-based approach to extract the data of interest that needed to be verified. The older certificates could not be used with these kinds of tools; hence an OCR model would have to be developed. The scanned certificates had to first be pre-processed, where the quality of the certificate had to be improved while also converting it to an image as the certificate was upscaled to improve the results when applying OCR. A rule-based approach refers to a set of predefined rules for decision making to arrive at certain conclusions e.g., if and else statements (Saxena, 2023).

In this case, as the certificates always had the exact same structure, the data could be extracted following a predefined set of rules. For example, some of the tables containing measurement results would contain six columns in total, five columns containing a number with six decimals and one column containing the status which could be either *PASS*, *FAIL* or *UD*, where *UD* meant the results were *undefined* i.e., the measurement result being above, below, or equal to the specified upper and lower limits by a margin less than half of the uncertainty interval defined by the case company, meaning it wouldn't be possible to determine if the result was *PASS* or *FAIL* based on a 95% level of confidence. Hence, every measurement result could be extracted using regular expressions to search for patterns with this structure.

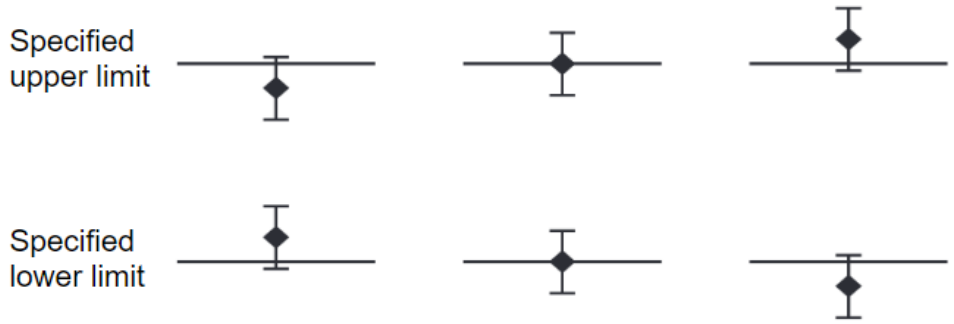


Figure 8: Definition of UD

Another alternative to get relevant data from a certificate was to modify the CPT and add a few lines of code that would convert the object model into a JSON object and then parsing out the relevant data from it. While this method would force the already existing digital copies of the certificates to be printed again if the history of the device was desired to be saved, it could potentially be the far easiest way of getting out the relevant data needed when automating the process, as the data would already be correctly structured and easy to parse as JSON is a fairly popular data format that can be used by more or less by every software application to handle data. Below is a figure which demonstrates how it could be used in this project to determine the viability of this solution.

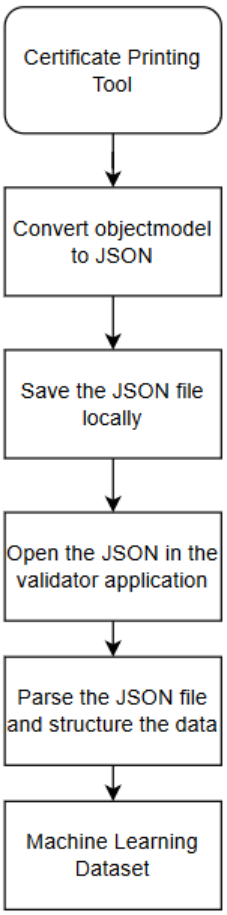


Figure 9: Certificate Printing Tool Flowchart

As for the database, the certificates are generated from data stored in different databases when using the CPT. In this project, only the measurement database which included all the measurement results and could be used to train a ML model. The measurements also included results in different units depending on which function had been used, e.g., voltage and resistance. There were two types of results, measured and generated results. In measured results, the device only measured what the system was generating i.e., indicated values and reference values while generated results would generate and the system measure, which was then compared to get the results.

Hence, the data was different as generated results included different type of data and setpoints. Other things to consider was which type of device had been used as the different devices had different specifications and pass limits, meaning each device would most likely require a model trained on its specific data.

3.3 Modelling

For the certificate data extraction, a script which extracted plain text out of the electronically signed certificates had to be made while OCR had to be used to extract the text out of the scanned certificates. As for the CPT, a way to convert the certificate data into a usable format had to be explored.

To be able to implement ML to the certificate validation process, it had to be determined if the model was able to correctly predict the measurement result classes, i.e., the model needed to be able to predict whether a measurement was *PASS, FAIL* or *UD*. Therefore, the type of model that was required would have to be a supervised model solving a multi-class classification problem as the desired outcomes were known and each measurement could belong to one out of three possible classes. And as there are many different algorithms, the first thing that had to be done was compare several algorithms on the datasets and select those that performed well for further evaluation.

3.4 Evaluation

The evaluation phase would focus on the performance of the models developed and find out what the predicts made were. To do this, unseen measurement results had to be used as the input and the output of the model would then be compared to the ground truth to determine how well the model performed. Before a ML model could be developed, the datasets had to be analysed and any noise or missing values removed to ensure optimal performance was able to be achieved.

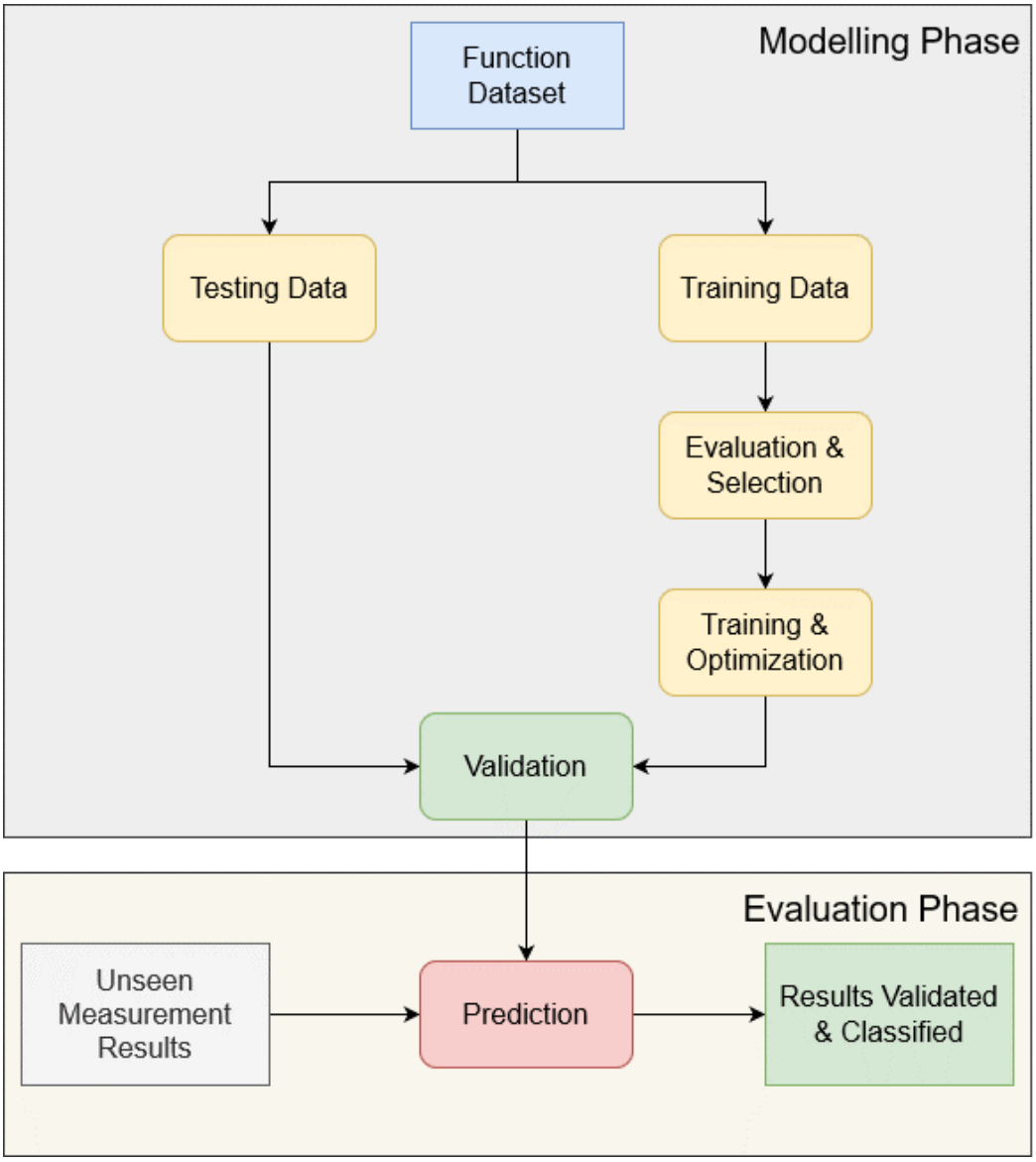


Figure 10: How Data was Split, Models Selected and Optimized, and then Used to Predict the Class for New Unseen Measurement Results

4 Project Development

This chapter focuses on the development of the components that would be used for the automated verification process, which includes how the development environment was set up, how the data was collected and analysed, and what algorithms were selected.

4.1 Environment

Starting of with the environment, the development environment chosen for this project was Visual Studio Code or VSCode, which is lightweight code editor developed by Microsoft. VSCode was chosen due to it's ease of use and large library of extensions. The programming language selected for this project was Python, as most of the recent machine and deep learning libraries are Python-based. Python is a high-level programming language, i.e., easy for a human to read and learn to use, yet it's still able to perform complex tasks similar to the low-level languages such as C and C++ when required. Thanks to these features, Python has since become the most popular language to be used for analytics, data science and more relevantly, machine learning projects (Raschka et al., 2020).

Additionally, as the project focused on experimenting with different methods and tools, Jupyter Notebook was used to document and separate code into smaller chunks. Jupyter notebook is essentially an interactive notebook document, that allows developers to share live code, images, description, and outputs. Code is easy to write but without proper documentation, it can be difficult and time-consuming to figure out how the code works and why certain functions were made as they were after a few weeks of not having read the code or sharing it to another developer. As the deployment part was not included in this project, these notebooks were used to build the foundation of the system as it allowed for much easier experimentation and documentation of the findings. While it's possible to use Jupyter Notebook's within production, it's generally not recommended as they are not computationally efficient.

4.2 Dataset

As earlier mentioned, there were three different sources where data could be extracted from in this project. Essentially, the data found on the certificates would act as the data to be validated by the ML model while the data from the database would be used as training data.

4.2.1 Digital Certificates

Two different methods had to be developed to be able to extract data from the already existing certificates. For the electronically signed certificates, a simple PDF extractor could be used while the scanned certificates required an OCR implementation.

Electronically Signed Certificates

To extract data from the selectable PDFs, PyMuPDF, which is a high-performance python library for data extraction and manipulation of PDF documents was used after which the extracted data was saved to a .txt file in plaintext. After that, the rule-based approach was applied to search for and extract data of interest. For example, to be able to find measurement data and its corresponding function, the keywords “Measurement” and “Generation” were searched for. If found, the word in front of the keywords would also be read, which acted as the start index. The text after the keywords would be read until the word “results” appeared, as this was used as the end index. The results from this were again saved to a .txt file as plaintext, the difference this time being that the .txt file only contained function names and measurement results.

To find the measurement data corresponding to the function name, regular expressions were used to match the pattern of the measurement results. For example, a simple regular expression could look like this:
(-?\d+\.\d+|±\d+\.\d+|(PASS|FAIL|UD).

Where *(-?\d+\.\d+)* searches for a number that can be negative and a decimal. It also has to consider the number as a float, allowing for values such as *-0.0000025* as the measurement results often had a high degree of precision. *±\d+\.\d+* works in a similar manner, but specifically looks for the ± sign Infront of the number. Lastly, *(PASS|FAIL|UD)* looks for either one of the three words listed and where the | sign works similarly to a logical OR, meaning the patterns searched for is either the pattern before or after the sign.

Lastly, after saving and cleaning up the extracted function names and measurement results, the data is converted from plaintext to the CSV format to make it easier to work with when using it as testing datasets for the ML models. Each CSV was labelled with the same name as the function name, e.g., if the function was *Voltage Measurement*, the CSV file would be labelled *Voltage_Measurement.csv*. Each function also had various number of functions, as some functions could include the ambient temperature or reference resistance as a column in the measurement results. Hence, when converting, a config file called *header config* was created where the section was the same as the function name and the headers or columns needed for the measurement results were included to. This was done to be able to structure the CSV correctly by placing each measurement variable under the correct column.

```
[Voltage_Generation]
headers = Generated;Measured;Difference;Uncertainty;Low Limit;High Limit;Status

[Voltage_Measurement]
headers = Reference;Indicated;Difference;Uncertainty;Low Limit;High Limit;Status
```

Figure 11: This Figure Shows the Headers Used to Structure the Measurement Results under the Correct Column.

A CSV file had to be created separately for each function, meaning the plaintext file containing the measurement values were looped through until every function had been found and its measurement data converted into a corresponding CSV which was then saved locally.

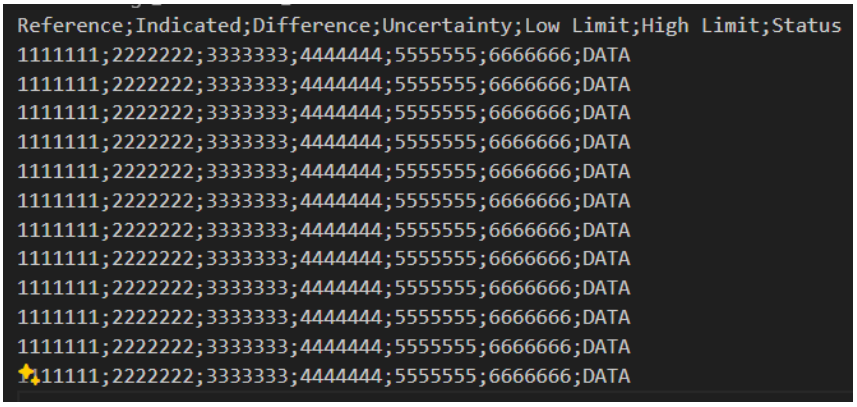


Figure 12: This is An Example of How the Data Would be Structured After Creating the CSV with The Structure Earlier Mentioned

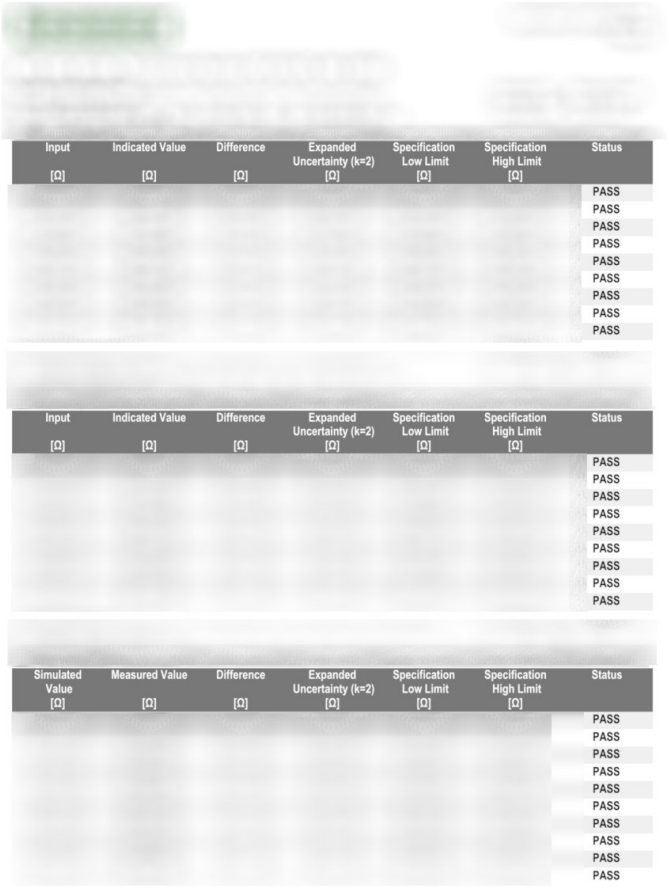
In reality, a version meant for production use would probably not have been made in this manner, but as it was mostly used for experimental purposes it was better to split the extraction process into several parts to be able to easier debug and see what kind of data was being extracted.

Scanned Certificates

Another potential data source was the older certificates that had to be scanned. To be able to extract the data from the images, OCR had to be used. The OCR used in this project was the pytesseract package which is a wrapper for Google's Tesseract-OCR and to improve the quality of the various scanned images, OpenCV was used to enhance and resize the images.

First, using OpenCV, the certificates were resized to 1024x1024 pixels to ensure each certificate was the same size. Then, binarization was used to convert the certificate to black and white, removing any colour. Lastly, skew correction was applied to correct the alignment as most of the scanned certificates had various degrees of skewing.

Having pre-processed the image, pytesseract was used to extract the text from the image and save it as plaintext. Having extracted the text, a similar rule-based approach was applied to the scanned certificates to be able to structure the data for the ML model.



Input [Ω]	Indicated Value [Ω]	Difference [Ω]	Expanded Uncertainty (k=2) [Ω]	Specification Low Limit [Ω]	Specification High Limit [Ω]	Status
						PASS
						PASS
						PASS
						PASS
						PASS
						PASS
						PASS
						PASS
						PASS

Input [Ω]	Indicated Value [Ω]	Difference [Ω]	Expanded Uncertainty (k=2) [Ω]	Specification Low Limit [Ω]	Specification High Limit [Ω]	Status
						PASS
						PASS
						PASS
						PASS
						PASS
						PASS
						PASS
						PASS
						PASS

Simulated Value [Ω]	Measured Value [Ω]	Difference [Ω]	Expanded Uncertainty (k=2) [Ω]	Specification Low Limit [Ω]	Specification High Limit [Ω]	Status
						PASS
						PASS
						PASS
						PASS
						PASS
						PASS
						PASS
						PASS
						PASS
						PASS

Figure 13: A Scanned Certificate Containing Different Measurement Results Calibrated Functions.

Some problems that occurred were that due to the varying differences in quality and skewing, the pre-processing did not always work and often some measurement result columns were missing from some functions.

4.2.2 Certificate Printing Tool

While getting familiar with the source code for the CPT, it was discovered that the data used for the certificate could potentially be directly taken from the CPT before printing instead of having to first print the certificate and then validate the data by using on of the earlier mentioned methods. To be able to extract the data and get it in a usable format, the first step was to convert the object model into JSON.

As the CPT was written in C#, Newtonsoft's JSON framework was used to serialize the object model. The converter only required a few lines of code and was set to be triggered when the *Print* button was pressed. When the function was triggered, a copy of the object model was made, converted into JSON, and then saved locally.

Figure 14: A Snippet of the JSON Structure When Converting the Object Model to JSON.

The conversion worked as expected and the data had a clear structure. This method also meant that it would be possible to avoid first printing the certificate and then having to extract the data. As this was only experimental, it didn't matter that the file was saved locally, but in a production version this data would have to be somehow sent between the automated certificate service and the CPT.

4.2.3 Database

As a precaution, recently made backups of the case companies' databases relevant to the project were copied and restored locally using the structured query language (SQL) management studio, a tool used to manage SQL databases and infrastructure, as not to tamper with the databases used in production. Three different types of databases were copied but data was queried only from one of these databases as the others were only used for things such as looking up function names, Ids and setpoints.

The calibration database contained every measurement result to date. To be able to query data for a specific device and function from the database, SQLAlchemy was used, which is a SQL toolkit for Python and an Object Relational Mapper (ORM) which allows for flexible and efficient database access and querying using a simpler Python syntax. To be able to extract the data from the SQL database and use it with Python, a schema had to be defined matching the tables, columns, data types and the relationships in the found in the database using SQLAlchemy's ORM system.

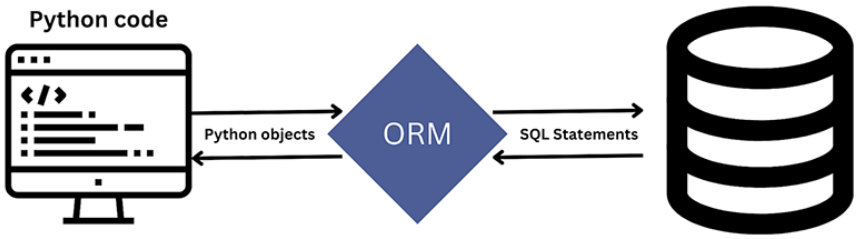


Figure 15: ORM Functionality Between Python and SQL to allow data from SQL to be used within Python Applications (Oyelekan, 2023)

Then, having created a connection string to the local database server and an engine, a session could be created to communicate between the application and the SQL database to query data. The queries included the device type, module type and function type. The reason for this is, as different device types had different set points and ranges, the data

and accuracy would differ between the devices, meaning a reading that might have passed for one device type might have failed for another. As for modules, each device had often multiple types of different modules which were used for specific units. For example, a device typically always consisted of two electrical modules and one pressure module. In this case, the functions chosen to train the ML models in this thesis were the Voltage Measurement (VMEAS) and Voltage Generation (VGEN) functions as these functions had a lot of data stored in the database as they were often calibrated. Lastly, the query results were again converted into the CSV format to be able to easier work with when developing the ML models. The VMEAS file contained 440707 samples while the VGEN file contained 345579 samples, both of which had seven features or columns. Having successfully queried the data, the next step was to analyse it and clean it up before training sets could be generated.

4.3 Data Analysis

The CSV files containing the data extracted from the case companies' databases were loaded in as a pandas dataframe. Pandas is a popular tool used within python to manipulate and analyse data and is very often used within ML development. Before loading in the data, the data types were declared for each feature to speed up the process and to ensure that each feature was loaded in with the correct datatype.

Having loaded in the data, the next step was to clean it up. The first thing to do was to remove missing data. As the data already had a solid structure, there were not much that had to be cleaned up. Only 516 rows were empty in the VMEAS dataset while the VGEN dataset had 160 empty rows. Normally, duplicates would also be removed in this phase, but neither of the datasets contained any duplicates. The next step was to perform feature engineering, which often refers to the process of creating or transforming features. In this case, new features had to be created as the features in the dataset did not match the features in the certificates. For example, the *Difference, Low Limit* and *High Limit* features all had to be created following the case companies' specifications.

Looking at the current dataset, a clear imbalance in the dataset could be seen as the *PASS* status made up most of the measurements found in both of the datasets. Imbalanced data is not rare, rather it is a very common problem. If there is too much of one class, the ML model will most likely become biased towards the majority class. Hence, the data had to be further pre-processed to balance the datasets. There also was an *Error* class which did not bring any value as *Error* usually meant something, either internal or external, had caused the device to give incorrect readings or completely fail. This class would only have added noise to the datasets; hence any *Error* result was dropped (Werner de Vargas et al., 2023).

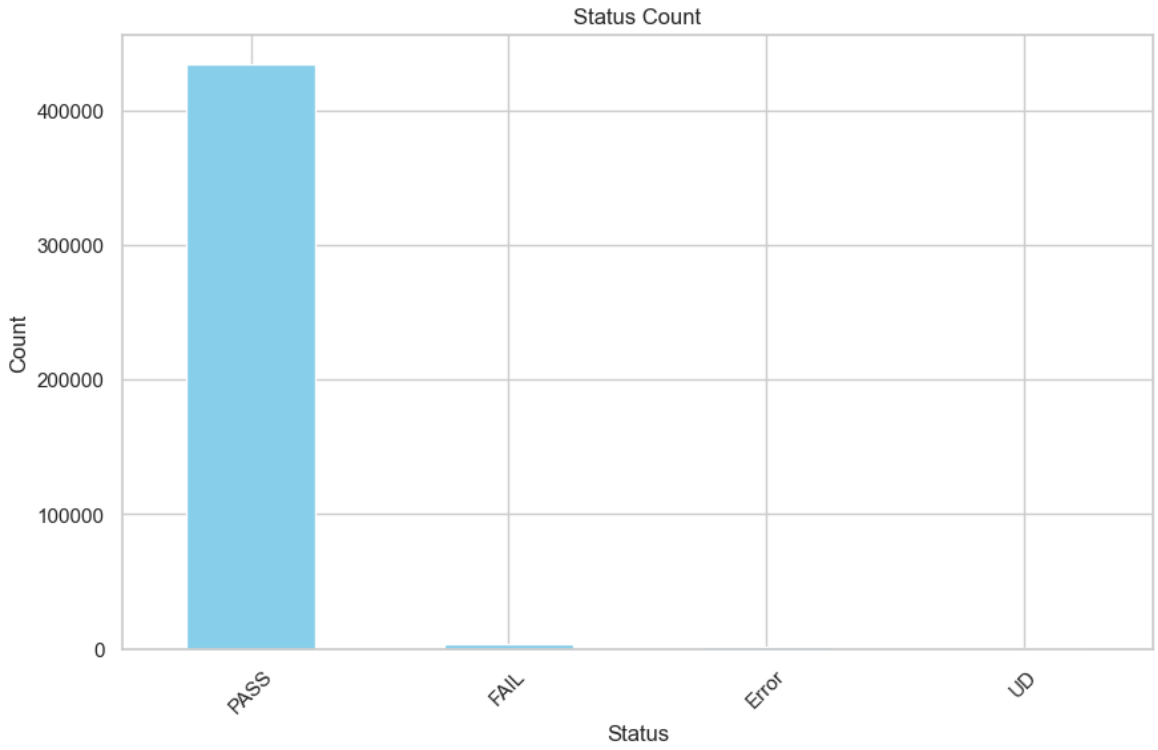


Figure 16: Before Balancing the Dataset Where a Clear Class Imbalance is Present

To balance out the data, sklearn's *resample* function was used, which resamples arrays or matrices in a consistent way and is often used to over- or under-sample datasets. In this case, the *PASS* class was decided to be under-sampled, which refers to the action of reducing the observations or samples found in the majority class to create a much more balanced dataset while the over-sampling method would increase the observations in a class (Werner de Vargas et al., 2023).

The value of the under-sample was decided by summing the total amount of observations from the *FAIL* and *UD* classes and then dividing it by the *PASS* class to get the under-sampling ratio. The under-sampling ratio was then multiplied with the *PASS* class to get the sampling value that it would be under-sampled to.

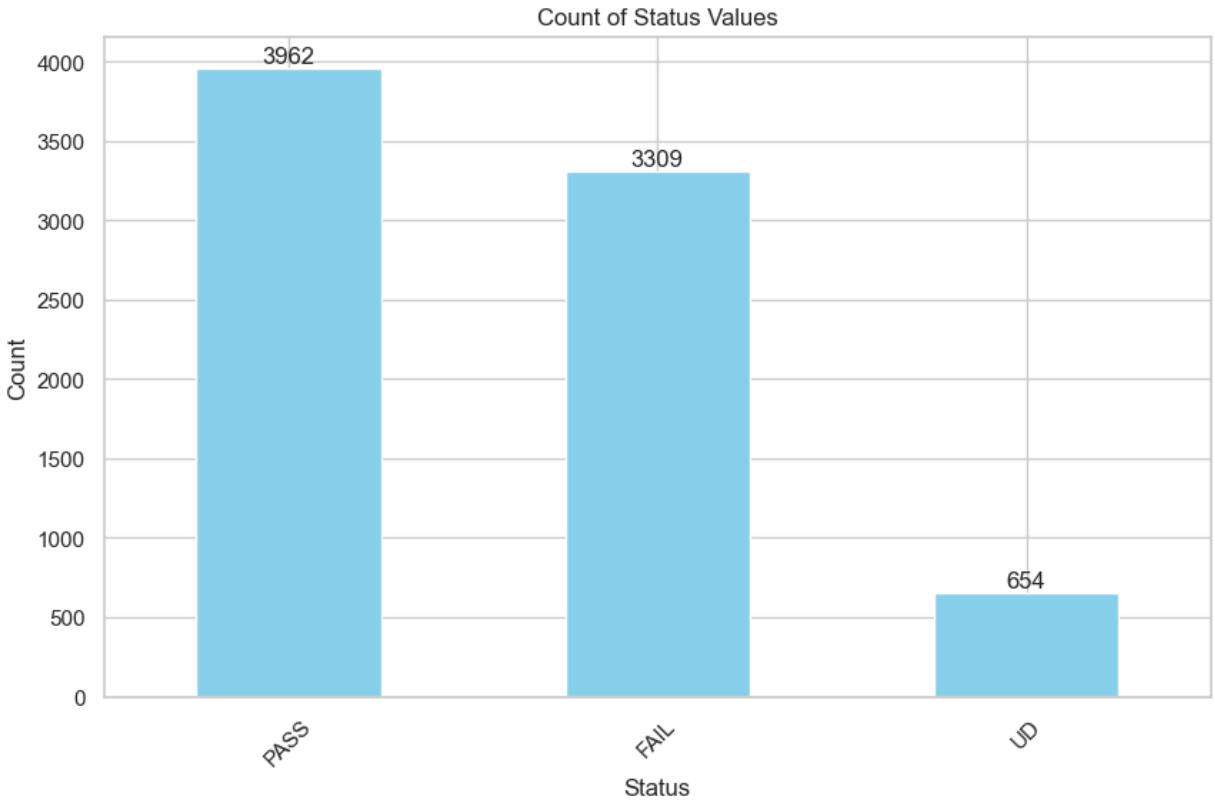


Figure 17: After Balancing the Dataset with the Under-sampling Method.

Another technique that can be used to balance imbalanced dataset is called Synthetic Minority Oversampling Technique (SMOTE) which is used to synthetically over-sample minority classes to balance the class distribution and is based on the k-nearest neighbors' algorithm.

SMOTE creates new synthetic samples by drawing lines between the nearest neighbors in the feature space of the minority class samples. In other words, when SMOTE draws a line between the samples of the minority class and its nearest neighbors, it calculates the difference between the two samples after which the difference is then multiplied by

a random number between 1 and 0. This result is then added to the corresponding minority class sample which then generates the new value for the synthetic sample thus creating similar but unique samples to balance the dataset (Brownlee, 2021a).

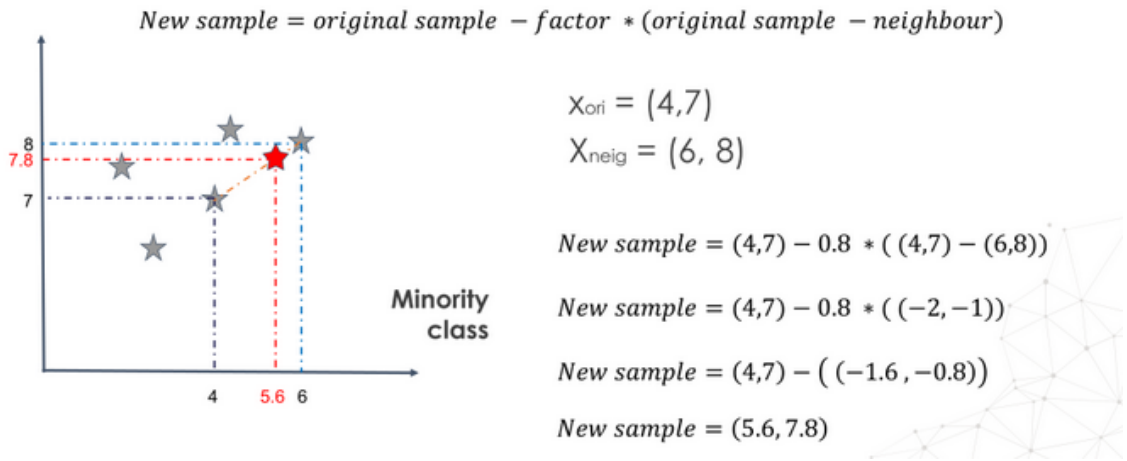


Figure 18: The Process How SMOTE Generates Synthetic Samples (Galli, 2023)

As discussed, SMOTE helps improve the performance of the models by reducing the amount of bias and reducing the risk of overfitting instead of randomly over-sampling with already existing samples. However, SMOTE also has its own limitations. When SMOTE generates new samples, it does not consider the quality or relevance of the samples it has generated which may not accurately represent the minority classes underlying distribution thus negatively impacting the performance of the model. Another possible drawback is the fact that SMOTE can potentially generate samples that are too similar to existing minority class samples and is far from the decision boundary. Hence, there exists a modified version called Borderline-SMOTE which was developed to combat this issue. Essentially, the only difference with this modification is that synthetic samples are near the borderline between the majority and minority classes, as theoretically, creating more samples close to the decision boundary should allow ML models to better separate between the classes. In the figure below, an example where the number of *FAIL* samples was chosen as the standard, *PASS* was under-sampled and SMOTE used to over-sample *UD* to balance the VGEN dataset (Galli, 2023).

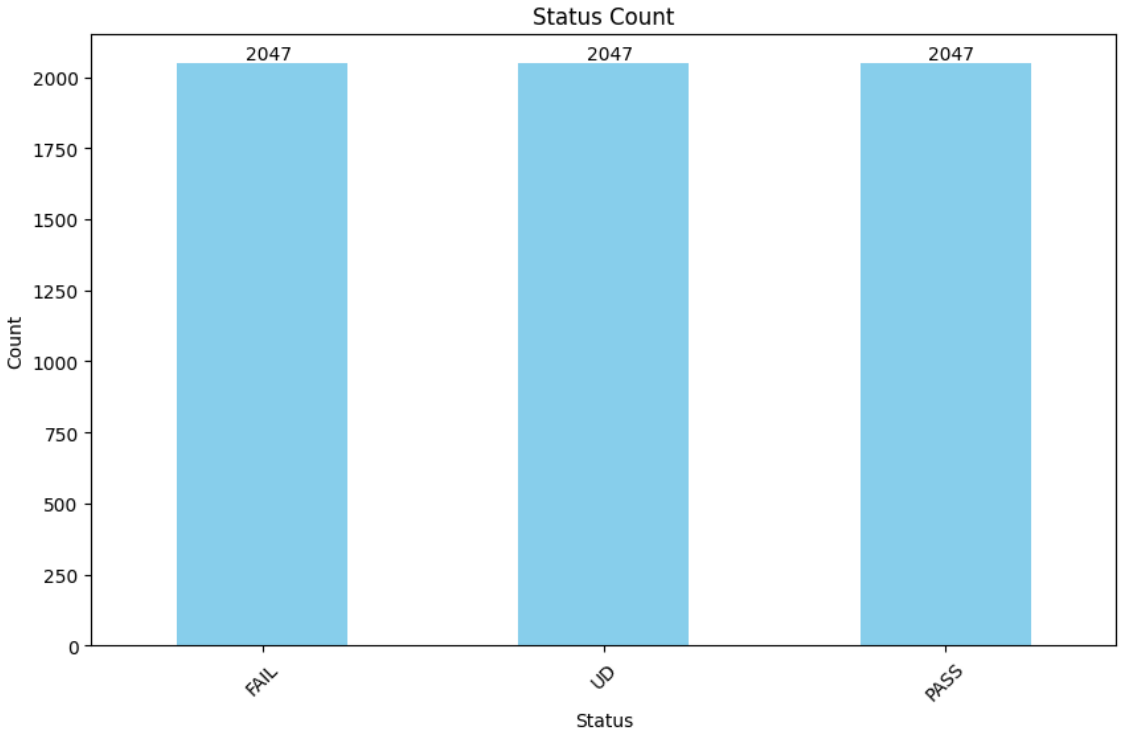


Figure 19: Using SMOTE to Balance the Dataset

4.4 Model Development

As the problem was a classification problem, the models chosen had to be classification models. After the cleaning, the data was split into testing and training datasets with a 20/80 split, meaning 20% was used as testing data and the remaining 80% as training data. The 20/80 is a common way to split the datasets and the reason for this is that out of all possible *p* values, where *p* is the parameter of the model based on the training set. The *p* value that should be selected is the one where the product of the function $p * (1 - p)$ is as large as possible, and when $p \geq 0.8$, the product starts to decrease. Hence, the 20/80 split is empirically the best split for training and testing datasets according to Gholamy et al., (2018).

As mentioned earlier, in this project ML would be utilized to verify that the measurement results had been classified correctly hence a classification model was needed. To be able to select a model, a comparison between the performance of different models had to be conducted. The models or algorithms mentioned earlier in Chapter 2 were the models chosen. The number of models was selected as it is relevantly easy to test and exchange models using Python, and because of the *No Free Lunch theorem* which states that there is not always a single model that will always outperform other models (Lones, 2021).

The first model performance test included training the models with the training dataset without tuning any hyperparameters to get an overall understanding of how the models performed. To be able to compare the performance, the accuracy, precision, recall, F1 and cross-validation (CV) metrics from the SCIKIT metrics library were used. CV was used as the main metric to choose the models to be further evaluated, as it is possible to estimate the performance of a model on unseen data. Essentially, CV consists of two steps: (1) splitting data into subsets of approximately the same size i.e., folds and (2) rotating training and validation sets among the subsets. One of the folds will always be used as a validation set while the model will train on the remaining folds. This process is then repeated using different folds as validations sets for each iteration. The final score would be the average score from all the performed iterations (Alhamid, 2020).

First, the performance on the VMEAS dataset had to be determined and models evaluated. To be able to determine if the performance was affected depending on the amount of folds used, several tests with different a different number of folds were conducted, and the results of these tests can be seen in figure 11. According to Lones (2021), the 10-fold CV is often used as a standard, hence it was chosen as the base value.

Ranked Models based on Cross-Validation Score				
Model	**cv = 5**	**cv = 10**	**cv = 20**	**cv = 30**
RFC	0,9763	0,9779	0,9789	0,9798
ETC	0,9759	0,9759	0,9771	0,9771
GBC	0,9726	0,9741	0,9751	0,9749
KNC	0,9696	0,9713	0,9732	0,974
GNB	0,656	0,6566	0,6563	0,6569
LR	0,5573	0,5573	0,5573	0,5573
SVC	0,5237	0,5237	0,5232	0,5232

Figure 20: CV Score Results on the VMEAS Training Set Using Different Folds

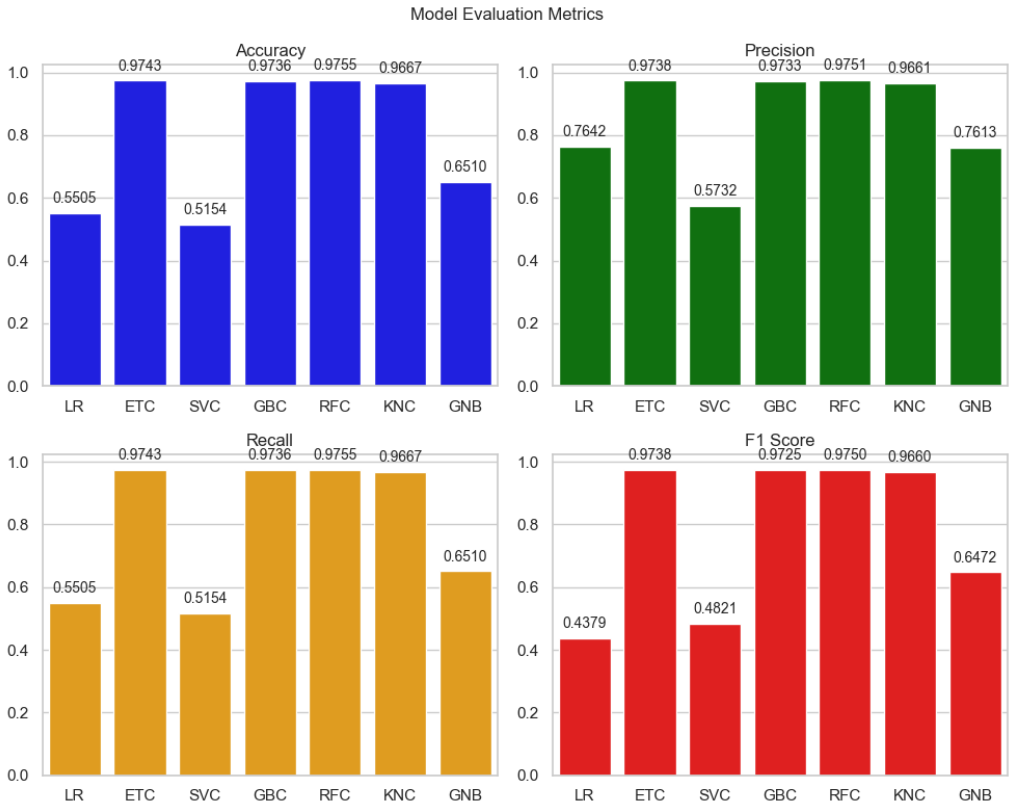


Figure 21: Model Performance on the VMEAS Dataset and the Resulting Metrics

Evaluating the results, the first four models (RFC, ETC, GBC and KNC) seemed to have the best performance on the VMEAS dataset which was also evident when plotting the other metrics. To better understand how the metrics used to evaluate classification models are calculated and used, refer to figure 22 which includes the formulas used to calculate the metrics.

Metrics	Formula	Description	Example
Precision (Brownlee, 2020)	$\frac{True\ Positive}{True\ Positive + False\ Positive}$	Calculating by dividing the correctly predicted positive observations by the total predicted positive observations. Measured on a scale from 1 to 0 where 1 is a perfect score	How often the model predicted a observation to be PASS out of all the actual PASS observations in the dataset
Accuracy (Jierula et al., 2021)	$\frac{Correct\ Predictions}{Total\ Number\ of\ Predictions}$	Calculated by dividing the total number of correct predictions by the total number of predictions. Measured on a scale from 1 to 0 where 1 is a perfect score	How often the model was able to correctly predict which observations were PASS out of all the predictions made
Recall (Huilgol, 2023)	$\frac{True\ Positive}{True\ Positive + False\ Negative}$	Calculated by dividing the total number of true positives by all of the actual positive observations. Measured on a scale from 1 to 0 where 1 is a perfect score	For all the PASS observations, recall tells how many the model correctly identified
F1 Score (Kundu, 2022)	$2 * \frac{Precision * Recall}{Precision + Recall}$	The weighted average of Precision and Recall where both variables are important. Measured on a scale from 1 to 0 where 1 is a perfect score	In case a PASS observation is predicted to be UD it could be a sign that the observation might have passed when it shouldn't have

Figure 22: Brief Explanation of the Metrics and How to Evaluate the Results

The true positive, true negative, false positive and false negative are variables used to evaluate the performance of classification models. In this thesis case, true positive would refer to when a model correctly predicted an observation to be *PASS* while false positive would refer to when the model predicted the observation to be *PASS* while in reality, the observation was *FAIL* or *UD*.

Similarly, different folds were also used when determining model performance on the VGEN dataset. At first glance, the average scores seem to be better for all models compared to the VMEAS dataset. The same four models that performed the best with the VMEAS dataset also had the greatest performance with the VGEN dataset.

Ranked Models based on Cross-Validation Score				
Model	**cv = 5**	**cv = 10**	**cv = 20**	**cv = 30**
RFC	0,9789	0,9792	0,9792	0,98
ETC	0,9775	0,9783	0,978	0,9792
GBC	0,9769	0,9766	0,978	0,9775
KNC	0,9766	0,9763	0,9766	0,9772
GNB	0,7248	0,7251	0,7245	0,7248
LR	0,5559	0,5554	0,5554	0,5571
SVC	0,5554	0,5562	0,5568	0,5554

Figure 23: CV Score Results on the VGEN Training Set Using Different Folds

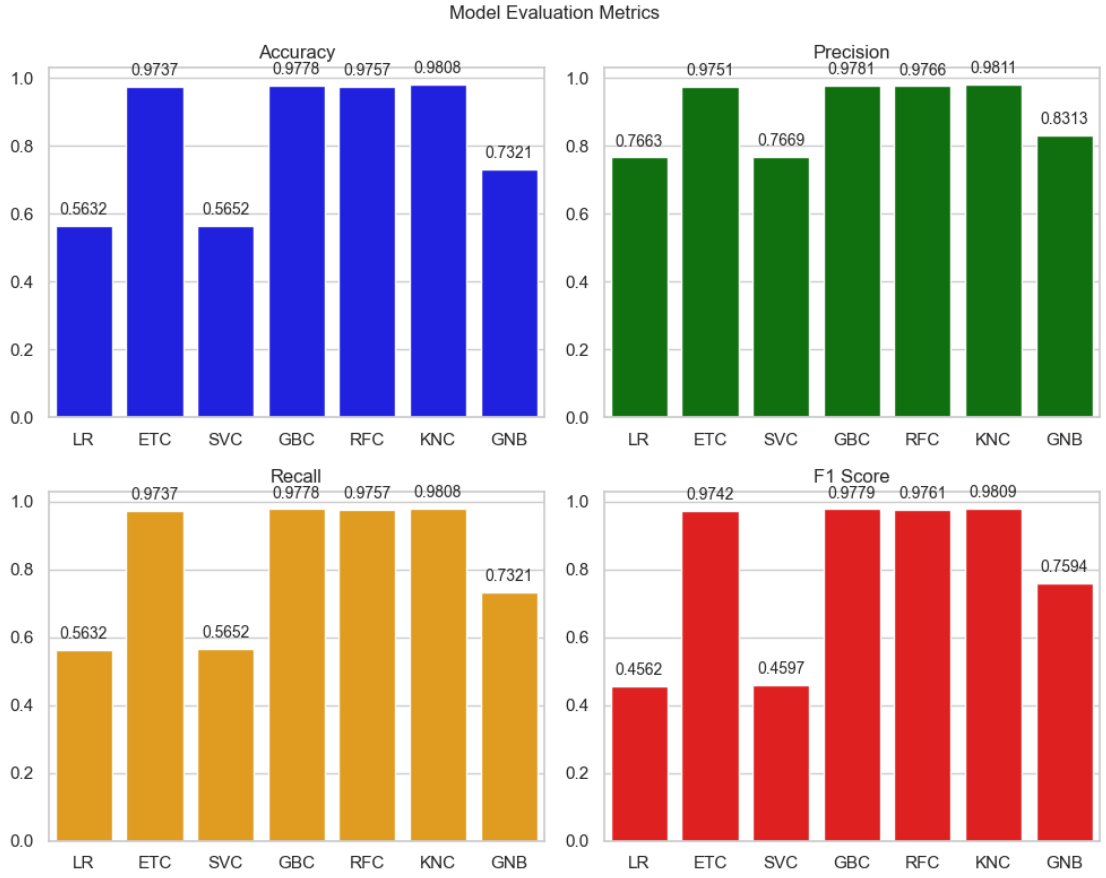


Figure 24: Model Performance on the VGEN Test Set and the Resulting Metrics

Lastly, the hyperparameters for the selected models had to be tuned. To do this, the selected models were given different hyperparameter values to try and from these values, the best ones selected. After the hyperparameter tuning was done, the models were then saved using the joblib package, which is essentially provides lightweight pipelining for large datasets. Using joblib, the models created were preserved so they could be loaded and used in other projects. The results of the chosen parameters according to the selection can be seen in the figure below.

VMEAS	
Model	**Parameters**
RFC	Max Depth = None Min. Samples Leaf = 1 Min Samples Split = 2 N. Estimators = 100
ETC	Max Depth = None Min. Samples Leaf = 1 Min. Samples Split = 5 N. Estimators = 50
GBC	Learning Rate = 0.5 Max Depth = 10 N. Estimators = 500
KNC	Algorithm = Auto N. Neigbors = 3 P = 2 Weights = Distance

VGEN	
Model	**Parameters**
RFC	Max Depth = None Min. Samples Leaf = 1 Min Samples Split = 10 N. Estimators = 400
ETC	Max Depth = 20 Min. Samples Leaf = 1 Min. Samples Split = 10 N. Estimators = 100
GBC	Learning Rate = 0.2 Max Depth = 2 N. Estimators = 200
KNC	Algorithm = Auto N. Neigbors = 3 P = 2 Weights = Distance

Figure 25: Selected Model Hyperparameters for Each Dataset

Tuning the hyperparameters will always depend on the type of problem that is to be solved and every model often has their own hyperparameters that can be tuned. The decision to tune a model or not would depend on different factors such as if the baseline performance, i.e., the initial performance with the default parameters is satisfactory or not. In this case, the model's performance was already excellent, but for experimentation purposes, the hyperparameters of each selected model were tuned to evaluate if tuning the hyperparameters affected the performance.

Hyperparameter	Description
n_estimators	Defines the numbers of trees in the forest, while for GBC this defines the number of boosting stages. For GBC, a larger number usually results in better performance as GBC is fairly robust to overfitting.
max_depth	Defines the depth of the tree. If None is specified, it expands until all leaves contain less samples than min_samples_split or until leaves are pure. For GBC, this parameter defines the maximum depth limit the number of nodes in the tree and controls the tree complexity.
min_samples_split	The minimum numbers of samples required to split an internal node. If the number of samples in the node is less than specified, the node will be a leaf.
min_samples_leaf	The minimum numbers of samples required to create a leaf node. Before generating the node, this parameter will evaluate whether a potential split will create a child node with fewer samples than specified in the min_samples_split parameter. If it does, the split will be avoided.
max_features	Defines the number of features to consider when determining the best split. Can be used to control overfitting.
learning_rate	Defines how each tree impacts the final outcome. A smaller value might require more trees but can result in a better performance.
n_neighbours	Defines the K value i.e., the number of neighbours to use for the majority voting.
weights	Defines the weight function to decide an outcome. Can be either uniform (all points weigh equal) or distance (the closer the neighbour the heavier the weigh)
algorithm	Defines the algorithm to use when computing the nearest neighbours. Can be either 'auto', 'brute', 'kd_tree' or 'ball_tree'
leaf_size	If kd_tree or ball_tree is selected, this parameter can affect the speed of the prediction as well as memory requirements.
p	Defines the metric to use which is the Minkowski Distance by default. If p = 1, the Manhattan Distance will be used and if p = 2, the Euclidean Distance will be used.

Figure 26: Different Hyperparameters that Were Tuned During the Development Phase (Arnold et al., 2024; Bartz-Beielstein, 2023; Scikit-Learn, 2024; Tools et al., 2023)

4.5 Testing and Evaluation

To try out the tuned models, a new project was created to verify that it was possible to load the saved models. To determine the accuracy and performance of a classification model, confusion matrices (CMs) can be used. A confusion matrix essentially summarizes the performance of a model with respect to some test data, i.e., true labels are compared to the models predicted labels (Ting, 2011).

Confusion matrices include two types of classes, one positive and one negative class, i.e., the true and false variables which were earlier discussed. In this case, true positive cells would be the diagonal line starting from the top right down to the bottom left. Similarly, if the true label was *FAIL* and the model predicted the measurement to be *FAIL,* this would be considered true negative result as *FAIL* represents the negative class. False positive represents when the observations true label was *FAIL* but the model predicted it as PASS (Ting, 2011).

UD however adds a bit more complexity, as it cannot be confidently said whether a *UD* result is *PASS* or *FAIL*. But considering that *UD* results are not allowed to be included in the certificates and the device would have to be re-calibrated until the measurements results are within limits, *UD* was considered as a negative class.

A dataset containing unseen data was used to determine how the models were able to predict. This data was acquired using the CPT where certificates containing VMEAS and VGEN measurements printed and then converted into JSON. The VMEAS test dataset contained in total 120 samples, where 100 were PASS, 10 FAIL and 10 UD. Similarly, the VGEN test dataset contained in total 110 samples, where 90 were PASS, 10 FAIL and 10 UD.

The first models to be tested and evaluated were the VMEAS models. The first thing that had to be done was to again split the data into X and y variables. X contained all of the measurement data while y contained the labels.

Then, the measurement data was fed to the model and a new y variable called *y_pred* was generated, which contained the labels the model had predicted for each model. With these values, it was then possible to create CMs out of all the models to compare the results.

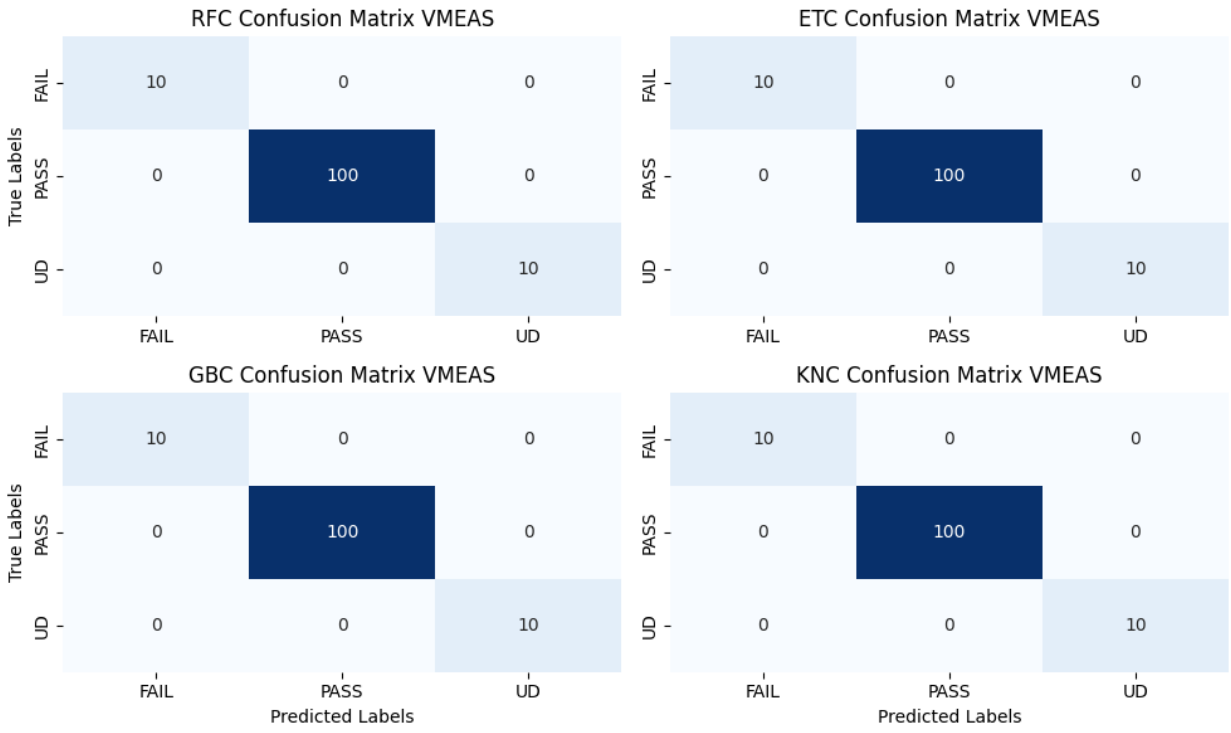


Figure 27: VMEAS Confusion Matrix from the Under-sampled Dataset

Evaluating the results, all of the models were able to correctly predict the labels of each of the measurement values from the unseen VMEAS dataset, meaning only true positive and true negative classes were present.

Model	Precision	Recall	F1 Score	Accuracy	CV Mean	CV Std
RFC	0,9630	0,9333	0,9415	0,9825	0,790	0,0699
GBC	0,9762	0,9714	0,9719	0,9657	0,8285	0,0571
ETC	0,9712	0,9625	0,9645	0,9721	0,775	0,0499
KNC	0,9712	0,975	0,9759	0,9611	0,875	0,0968

Figure 28: Tuned Model Metrics (VMEAS)

Other metrics were also used to evaluate the performance of the tuned model. The results are nearly perfect, which could potentially be a sign of overfitting. However, considering that the metrics from the training data are nearly identical with the unseen data and what the models were able to predict, this would not indicate overfitting, as the predictions on the unseen data would have been much worse as the models wouldn't have been able to generalize the data.

This time, Stratified K-Fold for the CV was used to evaluate the score. As the under-sampled datasets with a small data imbalance was used, Stratified K-fold was used which ensures that each fold has the same proportion of observations with a specific label, meaning both observations from the majority and the minority classes was included in the training and validation datasets. This also evaluates if there are any generalization problems, as it would be fairly obvious if the results were poor. The last column in the metric table refers to the standard deviation of the different folds i.e., the consistency of the performance. The lower the score, the lower the variation in the performance.

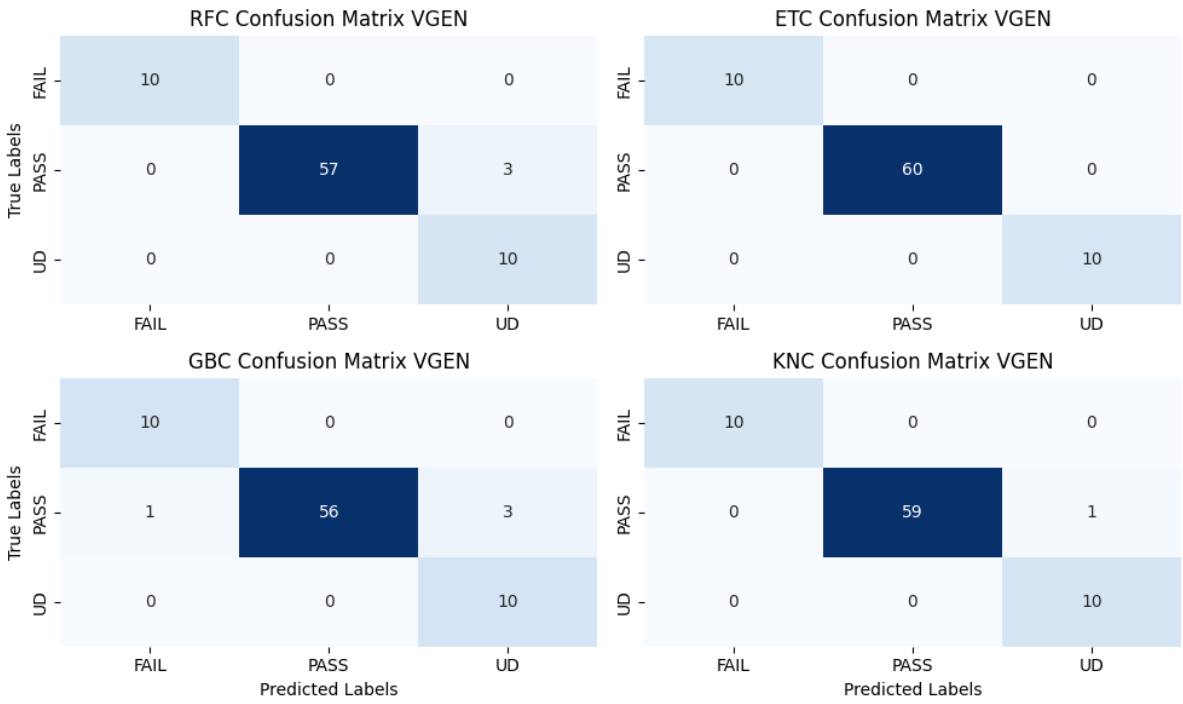


Figure 29: VGEN Confusion Matrix from the Under-sampled Dataset

The CM from the unseen VGEN dataset was also accurate, with nearly all labels correctly predicted. Most of the false negative results were due to the fact that some models were not confident enough if a measurement could be considered *PASS*, with the GBC model being the only model incorrectly labeling one *PASS* measurement as *FAIL*. However, it's worth keeping in mind that values that are close to the specification limits are difficult to label as *UD* or not, but in practice, if a measurement were to get a *UD* grade, the device would have to be re-calibrated as earlier discussed. Similar metrics to determine the performance were also used for the VGEN models which can be seen in the figure below.

Model	Precision	Recall	F1 Score	Accuracy	CV Mean	CV Std
RFC	0,9712	0,9625	0,9645	0,95	0,8125	0,0838
GBC	0,9762	0,9714	0,9719	0,925	0,8454	0,0913
ETC	0,9712	0,9625	0,9645	1	0,8032	0,0499
KNC	0,9712	0,975	0,9759	0,9875	0,875	0,0968

Figure 30: Tuned Model Metrics (VGEN)

Tests with the balanced dataset using SMOTE and Borderline-SMOTE were also conducted to evaluate if the technique could be used. The models trained on the balanced VMEAS datasets resulted in a worse performance than the under-sampled models as the amount of synthesized samples most likely affected the ability to separate between the classes as the margin between classifying a result as *PASS* or *UD* can be small, hence some of the synthesized samples most likely affected the model's ability to separate which measurement result belonged to which class.

On the other hand, the models that were trained and evaluated using the balanced VGEN dataset had a slightly better performance than the models trained on the under-sampled VGEN dataset when testing on unseen data. The Borderline-SMOTE however resulted in similar drops in performance as the VMEAS models, most likely due to the synthesized *UD* results crossing into or being too close to *PASS* results.

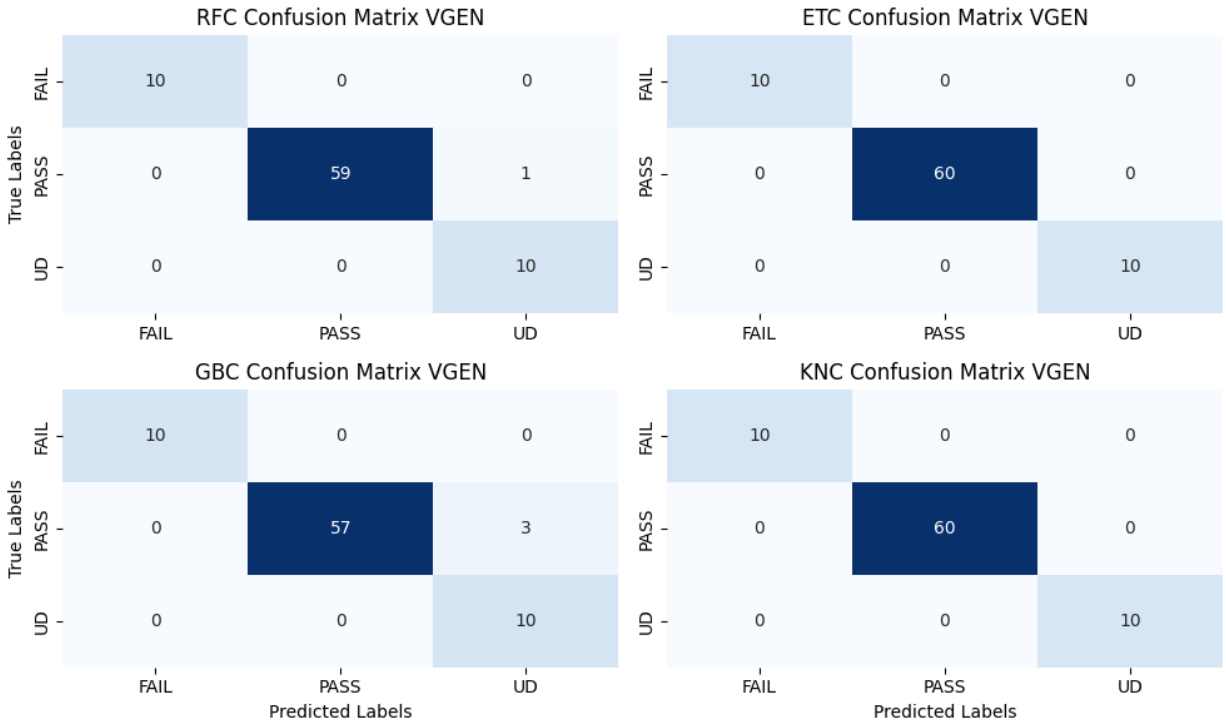


Figure 31: VGEN Confusion Matrix from the SMOTE Balanced Dataset

5 Results

This chapter focused on the findings made in the previous chapters and a potential solution to automate the certificate validation process that would not negatively impact the essential skills of the technicians while also fulfilling the requirements earlier defined.

5.1 Data Extraction Methods

Three methods to extract data from the certificates had been developed and the methods had various results. Starting of with the certificates with electronic signatures where a PDF extraction tool had been implemented, the results were promising as it was possible to extract all of the selectable text in the PDF as well as apply a rule-based approach to extract all of the relevant data. When comparing the measurement data found in the certificate to the data extracted with the extraction tool, the measurement data were identical to each other. The only problem with this method was that it required a lot of rules to be able to extract the data of interest without including non-desirable text, as the tool extracted everything that was selectable in the document. While not a big concern, this method expected the certificates to always have the exact same structure, meaning that a change to the certificate would require changes to the rules to be able to extract data and achieve the correct structure.

As for the scanned certificates with the OCR model, this method could have potentially worked better if the pre-processing had been developed further, as the age of the certificates ranged from 5 to 15 years where the structure and style of the certificates had been changed several times during this time range, resulting in the scanned data missing rows of measurement data or the numeric values being incorrect. Compared to the other methods, this method would have required more time to develop and test, hence the trade-off from implementing such method did not equal the amount of work it would have required compared to the two other methods.

For example, when evaluating the performance of the OCR model, metrics such as Character Error Rate (CER) and Word Error Rate (WER) were used. The CER metric takes inspiration from the Levenshtein distance, where it measured the minimum number of operations needed to transform one word into another. In the context of OCR, this would mean when the OCR model reads one word from the certificate, which would then be compared to the identical word from the ground truth used. If the word to be read was *Voltage* and the resulting word after applying OCR would be *Voltaeg*, this would mean the distance would be 2, as it would require two edits, namely swapping *g* with *e* and *t* with *g,* thus resulting in the OCR model having made two errors when the word *Voltage* was read. The CER metric is calculated by dividing the total Levenshtein distance for a set of OCR outputs with the total number of characters from the ground truth. A lower CER indicates a more accurate OCR system, with 5% representing a precise data recognition and 2-3% for high performance (Martinez, 2024; Rani & Singh, 2018; Timalsina, 2023).

WER on the other hand shows how well the performance of reproducing the text word by word was. WER can be calculated by summing the substantiations, insertions and deletions and dividing it by the number of words in the ground truth where substitutions refer to when a word in the ground truth gets incorrectly recognised and replaced e.g., the word *Resistance* gets replaced as *Renaissance,* insertion occurs when a word that was not in the ground truth gets added and deletion when a word found in the ground truth gets left out. The difference between CER and WER is, that while CER focuses on the characters, WER focuses on the words, hence the names. However, according to Alvermann, (2019) WER is rarely used to evaluate the performance of a model, as the WER score is not particularly meaningful when evaluating the quality of the OCR model as the WER can be three to four times higher than the CER due to the facts that a word only needs one different character or a different length to be incorrect. Hence, even if the OCR models CER is good, the WER can still be high.

However, if certain words are to be searched for in a text and the WER was relevantly high, the keywords might not be found. Say for example OCR has been applied and the WER was 15%. In this case, every time a keyword is searched for, every 15th search would result in the keyword not being found even if it should exist (Alvermann, 2019; Martinez, 2024).

To evaluate the performance of the developed OCR system a ground truth and a recognized text file was created. The ground truth was created by extracting data from a newer certificate with the PDF extraction tool. Then, the same certificate was converted into an image to make it comparable to a scanned certificate and then apply OCR to extract the text from the image. The structure was then modified so both the ground truth and the recognized text files both had the same structure so the words in the documents would be in the same position. The CER from the test ended up being 29%, i.e., 29% of the characters including letters, punctuations, and numbers, were incorrectly recognised when compared to the ground truth, indicating a rather bad performance which needed significant improvements as the minimum requirement would have been to at least achieve a 5% CER. In reality, the CER might have been even higher, as the structure had been modified to only evaluate the performance of the OCR system.

Similarly, the test ended up with a WER of 20% i.e., the model 20% of the words from the ground truth were transcribed incorrectly. A WER of 20% is an acceptable while a score of 5-10% would be considered high quality, similar to CER. This means that the OCR model was able to recognise words in the ground truth at an acceptable level but struggled with certain characters, as the certificates included several pages of measurement data and different unit characters such as ± and Ω. One solution to improve the CER might have been to blacklist special characters or certain terms, as they would not have been necessary to include.

These two data extraction methods were identified during the first project meeting. Initially, one of the objectives aside from automating the process with using these two methods was to be able to determine the aging of the device as the certificates included sufficient data to do so. The reason the history hadn't been saved earlier, was due to a decision made by the case company that the data already existed in the databases, hence it wouldn't be necessary to save the history when printing a certificate. However, when a technician prints a certificate, they choose which measurement results to include, meaning that while the same data is stored in the databases, the combination of the different functions and their measurement results as well as the date when the device was calibrated had to be known to be able to replicate the data that was included in the certificates, hence the third method of modifying the CPT would allow the certificate to only be printed once and the whole certificate could be saved as a JSON object, which was easy to work with and already had the correct structure.

These two methods were also explored before the focus group interview as it was difficult to schedule a focus group session that would have worked for both the COL and the service technicians. From the interviews, it became evident that the technicians wanted a solution that would verify the certificate as the certificate was being generated and not after, which the two first methods would have required. It was during this time that the source code for the CPT was reviewed and the possibility to convert the entire object model discovered. By implementing this feature to the CPT, it would be possible to meet the most important requirements set by the service technicians and save the entire certificate as a JSON object, which could then be stored and if needed, converted back into the original object model if required to be able to print it again. This again served as a good example why multiple points of views are important when conducting interviews, as had only the PO and COL been interviewed, this potential data extraction method would not have been considered and both the requirements and results could have been completely different. Similarly, had only the service technicians been interviewed, the challenges with the verification process would not have been as clear and the automated solution might not have been designed to be aware of these.

5.2 Model Selection

Seven common classification models were evaluated of which four performed exceptionally well. The result from the previous chapter is similar to what the *No Free Lunch theorem* implies, i.e., there is no single model that works the best for every situation which could be seen from the results from the two datasets, where the data and features were similar but had different results during evaluation. This would imply that each device's function as well as the measurement results would have to be modelled and evaluated similarly to as in Chapter 4.

One of the reasons why some models performed so well on the data is due to the fact that the datasets were low-dimensional i.e., very few features for each sample. Compared to the iris dataset which is often used within exploratory data analyses thanks to the dataset's simplicity, the iris dataset contains only four features. Similarly, the VMEAS and VGEN datasets also contained four features, with a total of seven features after feature engineering. A high-dimensional dataset would be the opposite. A good example of a high-dimensional dataset would be a grayscale image analysing problem where the images contained 75x75 pixels, the work would have to be done in a 5625-dimensional space. As the dimensions increase, the risk of encountering the *curse of dimensionality* also increases, a term which refers to the fact that as the features increase, the computational complexity and cost also increases while the data quality and model performance decreases due to the increase in sparsity which makes it difficult to collect data that represents the observations and population (Altman & Krzywinski, 2018).

Another reason why the performance was excellent on both the training and testing datasets might be due to the fact that the dataset consisted of several setpoints the device has to measure at and these setpoints would be the same for each device, meaning the datapoints always followed the same pattern.

These setpoints have been deemed to be the optimal points to measure to be able to determine the performance of the device by the case company. Each device however

had their own setpoints and accuracy, meaning that even if the setpoints were identical for two different types of devices, a model trained on data from one device would not work on the other. As mentioned in chapter 2 where project management for ML project was discussed, the case company has a solid data collection culture and thanks to the quality and simplicity of the data, the results were as excellent as they were.

The only noise in the dataset was found in the *Error* class, and by dropping this class, only a few missing values had to be removed. Data is not always this well-structured and simple as it has been in this thesis, hence a ML model might not even be the best solution to automate the certificate verification process as the problem that had to be solved was not complex and could most likely have been replaced by a rule-based approach. Another challenge often encountered in ML projects is the data imbalance in the datasets, which was also encountered in this thesis. Using the different sampling methods, it was possible to generate datasets that could be used to train the models.

However, these models were relevantly simple to develop and the biggest advantage with these models is that instead of having to create a rule-based implementation for each device, by using the tools that were mentioned in this thesis it would be fairly easy to create multiple models by using the different datasets and re-training them when necessary. Another advantage would be that when a new device or if a setpoint was ever to be changed, the implementation of a new model would not require anything else than to be quickly re-trained, evaluated and deployed as the ML models would be much better at adapting to new data while a rule-based approach would require carefully predefined rules as well as not being as adaptive.

Had the dataset been a high-dimensional dataset, a feature selection process would have had to be conducted using selection methods such as the wrapper, filter, and embedded methods which are often used when developing classification models where the aim is to drop features that do not have any influence on the final prediction or only add noise in the dataset. However, as all of the models performed better than initially

expected both on the training and the testing datasets, this process would most likely not have added any value as both the dataset and the problem to solve were fairly simple as well as the accurate predictions made by the models.

Selecting a model from the ones tested is not a simple task, as is evident from the evaluation phase, as all of the models performed nearly equally. More tests using unseen data and further analysis of the other datasets which were not used in this thesis should be conducted to make decisions on which model to use for which device and function. However, considering the performance of each model and the requirements to solve the classification problem in question, ETC could be used for the two datasets used in this thesis, as ETC is robust to both noisy and irrelevant features, computationally efficient as it construct the decision trees in parallel, is faster to both train and tune compared to the other models which was observed during the modelling and evaluation phase, and handles both imbalanced datasets and multicollinearity without issues, where multicollinearity refers to when two or more independent variables are highly correlated and can lead to unstable and/or unreliable predictions. Multicollinearity is more of a challenge within regression than classification problems, hence it might not have as significant of an impact in this thesis case.

5.3 Proposed Solution

Having explored different options and tools, the proposed method would be to develop and implement a web service which would handle the verification. The data to be verified would be sent from the CPT as JSON which the web service would then start parsing and searching for the sections that has to be verified. In Figure 34, the first version of the sequence has been created to give an idea of how the service would work.

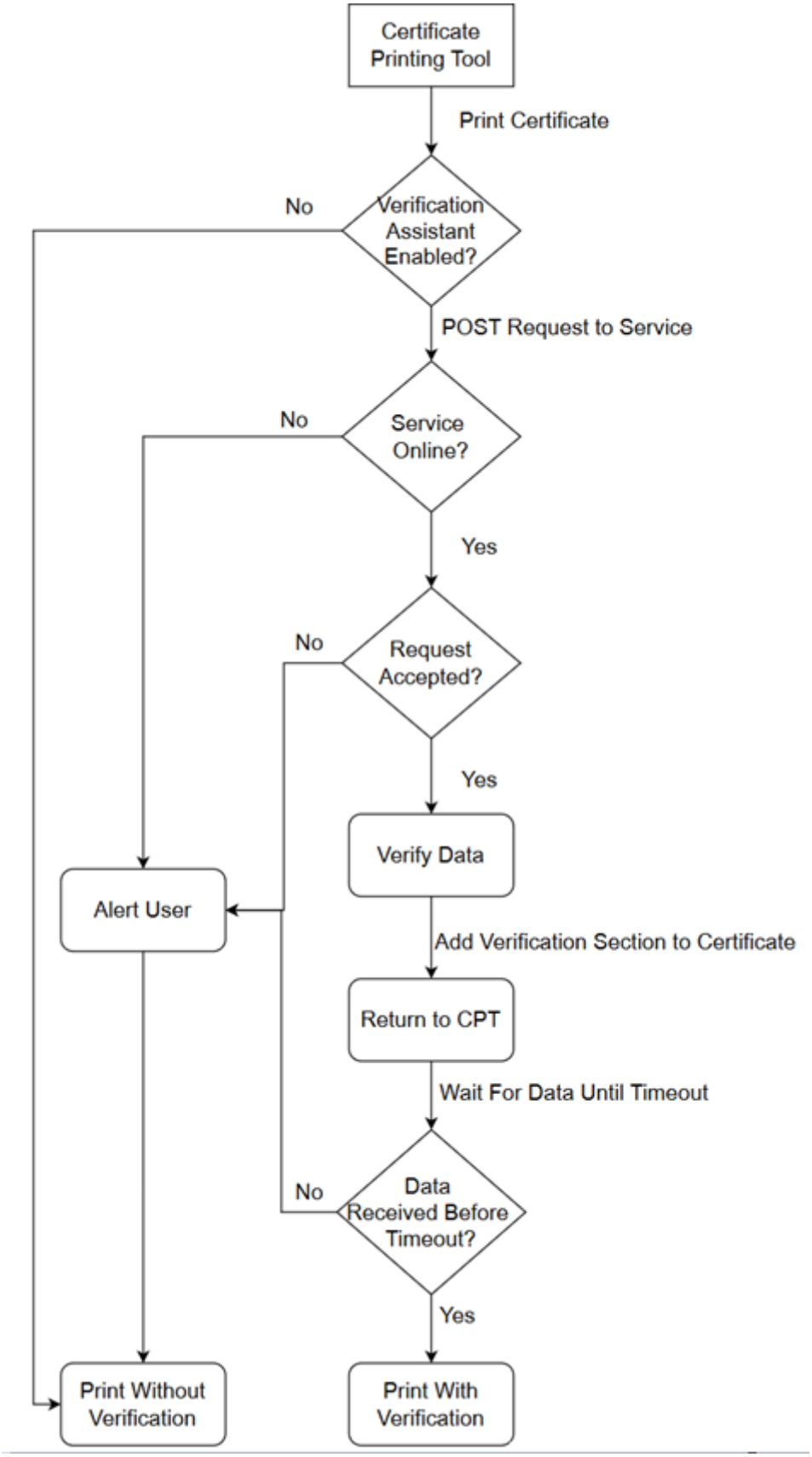


Figure 32: General Sequence of Certificate Verification Using the Proposed Web Service

Small modifications would have to be made to the CPT. First, a checkbox which the technician can enable or disable would be added to the tool which would enable the verification service. By default, the checkbox would be enabled and if the technician so wants, the service can be disabled to print the certificate in the exact same way it currently is being printed, thus not altering the existing tool, and allowing the certificates to be printed without having to rely on the service.

If the service is enabled, the CPT would send a request to the service to verify the certificate which it has converted to JSON and wait for an answer. If the service is offline, the technician would be alerted with a popup window asking if the technician wishes to continue or cancel the printing. If continue is selected, the certificate would be printed without any verification, similar to as if the assistant was to be disabled.

Then, if the service if online, the request has to be accepted. The request can be denied due to several factors, such as the expected data being incorrect i.e., expecting a JSON but receiving the data in an incorrect format, the technician sending the request is not authorized, the request being sent from an unauthorized domain or network, or due to some other internal errors. A version within the production environment would be required to include some kind of authentication and authorization, as all of the technicians need to have the correct roles to verify that they have the right to print certificates. Additionally, the service should block all requests except those sent from whitelisted origins i.e., the case companies' domain or specific networks. As this thesis only focused on creating a POC, this was not implemented but instead tested on a development server.

When the request is accepted, the data could be verified. The verification service would start parsing the JSON received and execute the verification logic for each of the different elements that has to be verified, such as the measurement results, technician ID and the measurement types. After the verification is complete, a new section would be added to the end of the JSON which includes everything that had been verified.

The object model would be modified to include a verification class so the verification data could be printed as a new page at the end of the certificate. By default, this section would always be empty and not included unless the validation service was used.

The CPT would then wait until the verification service had finished and returned the JSON. However, error handling in case the service was to go offline during the verification or if the verification took too long had to be implemented so the CPT wouldn't freeze and not print anything in case of problems. If an error was encountered during the verification, the technician would again be alerted that a timeout has occurred and be allowed to choose to print the certificate without verification. If everything worked without problems, the verified JSON file would be returned to the CPT which would then deserialize it back into the object model and print it in the same way it normally would, with the verified data being included at the end of the certificate. When the results had been verified and the certificate verified, the last page could simply be deleted before sending it to the customer.

Alternatively, instead of attaching the data to the certificate, only the verification results could be returned to the CPT after completion which could then present the results in a separate pop-up window together with the certificate so the technician could use the verification results as a check-list and verify that the results were correct or check what the verification service found and re-generate the certificate if needed.

Additionally, by implementing this method, the whole certificate as a JSON object could be forwarded and saved to a database, thus allowing the certificate history of the devices to be stored. For example, stored certificates could have the ID of the certificate as well as the device serial number as the keywords when searching for a specific certificate. Then, another modification could be implemented to the CPT where it would be possible to print or view the stored certificates instantly by querying the database storing the JSON files.

The selected JSON would then be deserialized back into the object model and the certificate could be printed. By doing this, the technicians wouldn't have to save any digital copies of the certificates manually, as this could be done either by the validation service or by the CPT itself.

To test the proposed solution, a small POC was created. Flask, which is a web framework for Python which allows web applications and services to be easily created, was used to build, and deploy the service. The CPT was also modified to enable the certificate to be sent to the service. As the communication occurs over the web, the protocol used was HTTP, where the CPT sent a HTTP request i.e., a POST command, which included the converted certificate. CPT then waited for the service to finish the verification and after receiving the response, deserialized the JSON back to the object model which allowed the certificate to be printed.

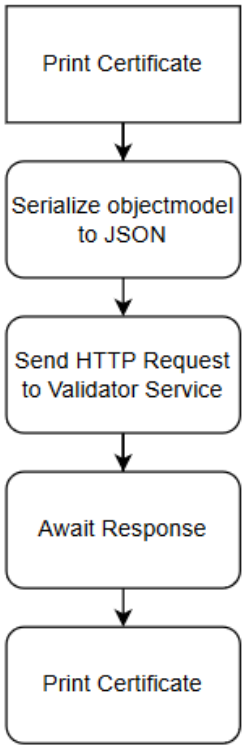


Figure 33: The Sequence When Printing a Certificate with the Modified CPT

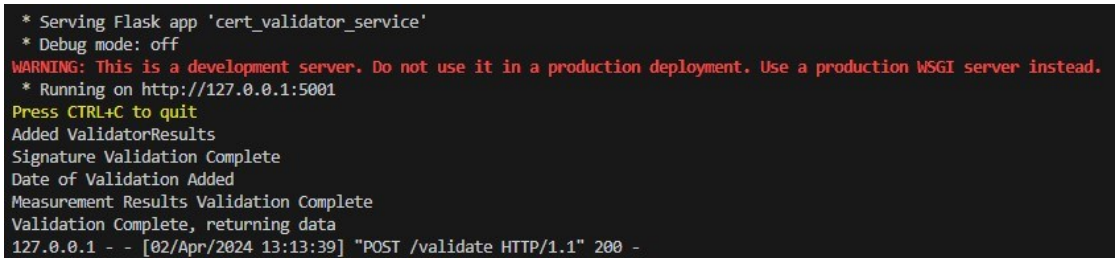


Figure 34: Screenshot of the Service While Running as A HTTP Request was Received, the Certificate Processed, and a Response Sent Back

Initial tests looked promising, as the service was able to quickly verify the certificate and send back the certificate with a validation section attached. The name of the technician who printed and signed the certificate was verified by only parsing the correct sections from the JSON and comparing the names, as they had to match while the measurement results were verified with one of the ETC model earlier created to verify that it was possible to load it with the service and have it verify new unseen data, similar to the tests in the evaluation phase.

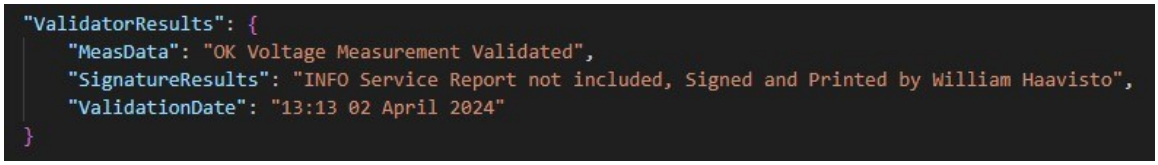


Figure 35: The Validation Results Added by the Service Before Returning the Certificate

By mixing rule-based approaches and ML approaches, it was possible to improve the speed and consistency of the verification process and, as simple checks and comparisons did not require anything else than simple logic to perform thus also saving on computational costs.

As there are several devices with several functions generating different types of measurement results, this would require models to be developed and deployed for each device. Hence, when the verification service receives the certificate to verify, it must first look which device the certificate belongs to before looping over all functions found in the certificate before it can start verifying the measurement results.

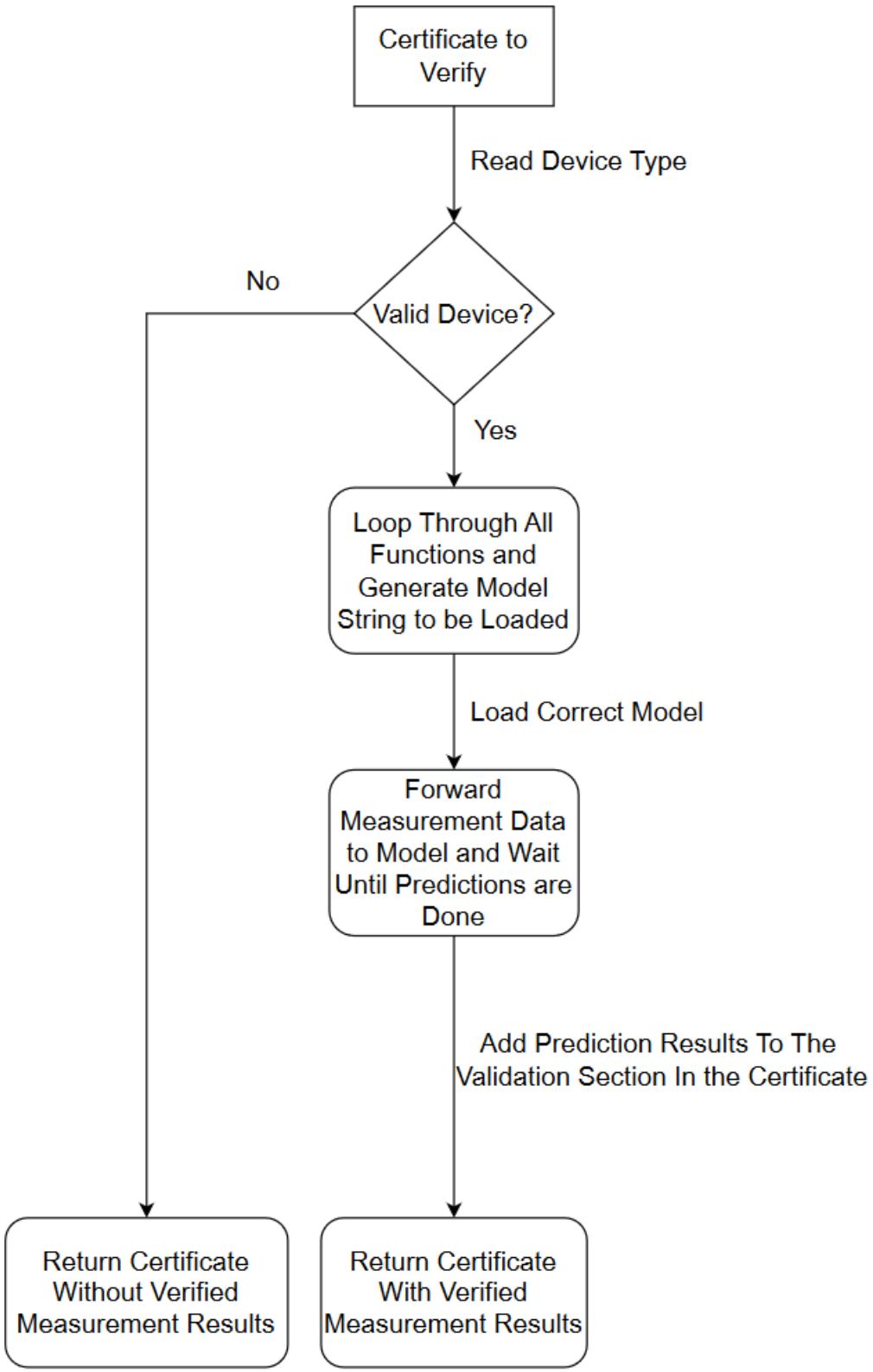


Figure 36: Sequence of how the Measurement Results would be Verified by Loading the Correct Model

As seen in the Figure 38, the device must be valid, otherwise the measurement results would not be verified. If it is valid, the functions would be looped through to build the model string i.e., the string which was used to search for and load the correct ML model. For example, if the device is MEASDEV1 and the functions found are VMEAS and VGEN, the strings that would be generated are MEASDEV1_VMEAS and MEASDEV1_VGEN which would have corresponding models with similar names. When all of the models have been loaded, the predictions were made, and the results attached to the verification section at the end of the certificate before returning it back to the CPT so the results could be analysed by the technician.

The proposed solution was presented to the PO who was satisfied with the proposal and agreed that it would most likely be able to assist in improving the certificate verification process. The PO preferred the alternative solution of presenting the results as a separate popup, as the verification results should not be included in the certificate, as it could potentially result in a technician accidentally including the section and sending it to the customer which was a potential risk as the page wasn't automatically removed by the CPT. This solution also fulfills the requirements determined from the focus group interview with the technicians as well as the COL as it would perform the verification the moment the certificate is generated, while not completely automating the process but simply assisting and suggesting the technician verify that the measurement is correct in case something was to be caught.

6 Conclusion

Lastly, the thesis was concluded with a retrospect to see if the RQs were able to be answered, what the different challenges and limitations of the thesis were and suggestions about future research made.

6.1 Retrospect

The aim of the thesis was to research if a manually performed process could be improved by implementing automation. The topic was suggested by the PO when a suitable project was searched for at the case company. No earlier research into automating the certificate verification process had been conducted by the case company before this thesis, hence most of the time went to familiarising with the current process, available data and planning the potential solution.

There were some challenges in the project, one being finding a suitable time to gather enough participants for the focus group interview session as both the COL and the technicians had heavy workloads during the time the thesis was conducted, hence the other data extraction methods might not have been explored and the proposed solution might have been developed further than what it was had the specifications and requirements been defined earlier. The focus group also showed the importance of getting multiple perspectives, as both the COL and the technician's input were required to create the requirements. Kontio et al., (2004) points out, that focus group can introduce bias to the results if conducted without proper planning or with an inappropriate group dynamic. A small sample size can also pose as challenges for focus group interviews, but as this was an in-house solution, this did not impact the results. Exploring the other methods were by no means a waste of time, as they could be used in other projects, for example in projects where the data wasn't stored in databases or without similar tools such as the CPT. Another challenge was, as earlier mentioned, no prior research had been done to automate the process by the case company, hence everything that could be used to automate the process had to be identified and evaluated.

The thesis also shows both the importance of a solid data collection culture, as carefully preparing the data before beginning the development of a ML model will result in much better performance as well as the importance of comparing different types of models, which again the *No Free Lunch theorem* proves. The data used in this thesis was low-dimensional and well-structured, making it easy to work with. There are also some managerial implications with implementing the proposed solution. These include:

- **Improved Efficiency** – By implementing the verification service, the verification process can help speed up the process and reduce errors, resulting in time being possible to allocate to other tasks and improved operational efficiency.

- **Cost Reduction** – By automating the process, the amount of incorrect certificates would be reduced as well as the cost of having to re-generate certificates and re-applying an electronic signature.

- **Continuous Improvement** – By encouraging the technicians to make improvement and additional feature suggestions to the service, continuous improvement of the service can be achieved.

- **Consistency** – By implementing the verification service, it would ensure that each certificate is verified using the same standards and logic, ensuring the results and the process itself would become more consistent.

Going back to the RQs, the question of *How should the process be structured to allow the certificate validation to be automated?* Is effectively answered by the focus group interview and the proposed solution, by modifying the CPT and implementing a separate service, the process can be structure in a way that allows an automated service to be implemented without disrupting or making major changes to the current process, but instead extending it. This however might not be true for every single process, hence each process should be carefully explored and evaluated before proposing improvements.

The second RQ of *How could the human intervention and automation be balanced in the certificate validation process?* is addressed by the proposed solution. First, the technicians would be allowed to either enable or disable the service. Second, the automated service would not change any data found in the certificate currently under verification, but instead would only return what it has verified and its results, acting as an assistant to the technician without making any of the decisions. The technician would then have to analyse the results and check the certificate in case any irregularities were detected. Hence, a balance between the technician's intervention and automation would be achieved. This is however limited to this specific case, and being able to achieve a balance might not be the same in similar projects and the type of balance would also depend on the project requirements and desired outcomes.

6.2 Limitations

The thesis was limited to using only one of the case company's devices and two of the device's function datasets, hence the results would most likely vary as every device has different setpoints, specifications and accuracy. Thus, as earlier discussed, every device would have to have their own models for each of their functions, which was not conducted during the thesis. Other limitations included the time and resources that had to be allocated to defining the requirements and experimenting with different tools and methods, which did not leave much time to develop the service itself. Thus, the service could only be proposed until developed to the stage of being able to be deployed as a minimum viable product. Further research and development have to be conducted before it can be verified if the proposed solution would be feasible in a production environment.

6.3 Future Research

Recommendations for future studies would be to research whether possible to improve or extend the CRISP-DM framework for ML projects. CRISP-DM is a simple and straightforward framework but does not consider the experimental and the constant required changes in ML projects, as the framework has not been maintained for a long time.
For example, Hayat & Widyani, (2023) proposed a guideline of how to combine Scrum concepts and principals with CRISP-DM. One justification presented was that as CRISP-DM divides the phases of the project into smaller steps, Scrum could potentially complement this decomposition as Scrum makes task management and tracking simpler through ceremonies and artifacts, where ceremonies refer to the daily standups and sprint planning and artifacts the product and sprint backlogs used in Scrum. The problem with Scrum, however, is that it requires at least four team members to be feasible, where one member is the PO and another the scrum master. Thus, for small scale project such as the one conducted in this thesis, Scrum wouldn't be viable hence the CRISP-DM works much better when the team is under four members and the project just needs a structure. But it does not mean Scrum concepts couldn't be incorporated, such as the weekly or daily standups to inform the PO of the current progress.

Another proposal to the framework was made by Studer et al., (2020) who proposed a process model which incorporates ML and quality assurance concepts to the CRISP-DM framework. One of the major challenges with effective ML project development seems to be the lack of guidance in the development process, poor data preparation and poor development execution, which the proposed CRISP-DM extension sought to address. The proposed model however couldn't consider each possible ML project, meaning the framework might not be viable for every project. Studer et al., (2020) points out, that there is a lack of standards within the ML industry, as the standards and project management methodologies has not grown as quickly as AI and ML development. Hence, a suggestion as a future study would be to research whether the CRISP-DM framework could be extended by incorporating concepts from Scrum while also assuring the framework would consider the challenges often encountered in ML projects.

Another potential study related to the certificate verification process could be to investigate the possibility of developing a DL model instead of several ML models to validate the certificate, either by extracting certain data of interest or by learning the structure of the certificate.

Clément, (2021) conducted a study where the purpose was to provide methods for ways to teach DL models to extract information from business documents. In Clément's study, the conclusion was that pre-trained models required very few training documents to be able to extract data at maximal performance, indicating that pre-trained models could be experimented with to evaluate if DL models could be used and whether or not these models would performed better than the ML models developed and make the proposed solution more time and cost-effective when implementing automation to the certificate verification process.

7 References

Ahmad, S. F., Han, H., Alam, M. M., Rehmat, M. K., Irshad, M., Arraño-Muñoz, M., & Ariza-Montes, A. (2023). Impact of artificial intelligence on human loss in decision making, laziness and safety in education. *Humanities and Social Sciences Communications 2023 10:1*, *10*(1), 1–14. https://doi.org/10.1057/s41599-023-01787-8

Alhamid, M. (2020, December 24). *What is Cross-Validation?* https://towardsdatascience.com/what-is-cross-validation-60c01f9d9e75

AlMahamid, F., & Grolinger, K. (2022). *Reinforcement Learning Algorithms: An Overview and Classification*. https://doi.org/10.1109/CCECE53047.2021.9569056

Alnoukari, M., & El Sheikh, A. (2011). *Knowledge Discovery Process Models*. 72–100. https://doi.org/10.4018/978-1-61350-050-7.CH004

Altman, N., & Krzywinski, M. (2018). The curse(s) of dimensionality. *NATURE METHODs |*, *15*. https://doi.org/10.1038/s41592-018-0019-x

Alvermann, D. (2019, October 26). *Word Error Rate & Character Error Rate – How to evaluate a model*. https://rechtsprechung-im-ostseeraum.archiv.uni-greifswald.de/word-error-rate-character-error-rate-how-to-evaluate-a-model/

Anand, M. V., Kiranbala, B., Srividhya, S. R., C., K., Younus, M., & Rahman, M. H. (2022). Gaussian Naïve Bayes Algorithm: A Reliable Technique Involved in the Assortment of the Segregation in Cancer. *Mobile Information Systems*, *2022*. https://doi.org/10.1155/2022/2436946

Anirudh, V. (2019, June 11). *How CRISP-DM Methodology Can Accelerate Data Science Projects*. https://analyticsindiamag.com/crisp-dm-data-science-project/

Aravind. (2023, August 9). *Exploring Clustering Algorithms*. https://neptune.ai/blog/clustering-algorithms

Arnold, C., Biedebach, L., Küpfer, A., & Neunhoeffer, M. (2024). The role of hyperparameters in machine learning models and how to tune them. *Political Science Research and Methods*, 1–8. https://doi.org/10.1017/PSRM.2023.61

Ayele, W. Y. (2020). Adapting CRISP-DM for Idea Mining A Data Mining Process for Generating Ideas Using a Textual Dataset. *IJACSA) International Journal of Advanced Computer Science and Applications*, *11*(6). www.ijacsa.thesai.org

Bartz-Beielstein, T. (2023). Hyperparameter Tuning and Optimization Applications. *Hyperparameter Tuning for Machine and Deep Learning with R*, 165–175. https://doi.org/10.1007/978-981-19-5170-1_6

Black, J. E., Kueper, J. K., & Williamson, T. S. (2023). An introduction to machine learning for classification and prediction. *Family Practice*, *40*(1), 200–204. https://doi.org/10.1093/FAMPRA/CMAC104

Brasjö, C., & Lindovsky, M. (2019). *Machine Learning Project Management A Study of Project Requirements and Processes in Early Adoption*.

Brownlee, J. (2019, September 24). *A Gentle Introduction to Model Selection for Machine Learning*. https://machinelearningmastery.com/a-gentle-introduction-to-model-selection-for-machine-learning/

Brownlee, J. (2020). Calculate Precision, Recall, and F-Measure for Imbalanced Classification. *Information Processing and Management*, *45*(4), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002

Brownlee, J. (2021a, March 17). *SMOTE for Imbalanced Classification with Python*. https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/

Brownlee, J. (2021b, April 27). *How to Develop an Extra Trees Ensemble with Python*. https://machinelearningmastery.com/extra-trees-ensemble-with-python/

Calders, T., & Custers, B. (2013). What is data mining and how does it work? *Studies in Applied Philosophy, Epistemology and Rational Ethics*, *3*, 27–42. https://doi.org/10.1007/978-3-642-30487-3_2/COVER

Casey, K. (2020). *Robotic Process Automation (RPA) in plain English*. https://enterprisersproject.com/article/2019/5/rpa-robotic-process-automation-how-explain

Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., & Lopez, A. (2020). *A comprehensive survey on support vector machine classification: Applications, challenges and trends*. https://doi.org/10.1016/j.neucom.2019.10.118

Clément, S. (2021). *Deep learning for information extraction from business documents*. https://theses.hal.science/tel-03521607

Clinton, J., Khabaza, T., Reinartz, T., & Wirth, R. (1999). *The CRISP-DM Process Model*.

Crisci, C., Ghattas, B., & Perera, G. (2012). A review of supervised machine learning algorithms and their applications to ecological data. *Ecological Modelling*, *240*, 113–122. https://doi.org/10.1016/j.ecolmodel.2012.03.001

Fawagreh, K., Gaber, M. M., & Elyan, E. (2014). Random forests: from early developments to recent advancements. *Systems Science & Control Engineering: An Open Access Journal*, *2*(1), 602–609. https://doi.org/10.1080/21642583.2014.956265

Flemisch, F., Heesen, M., Hesse, T., Kelsch, J., Schieben, A., & Beller, J. (2012). Towards a dynamic balance between humans and automation: Authority, ability, responsibility and control in shared and cooperative control situations. *Cognition, Technology and Work*, *14*(1), 3–18. https://doi.org/10.1007/S10111-011-0191-6/FIGURES/17

G. Shobana, K. P. D. S. (2018). *Harnessing the power of Machine Learning for Automating the Repetitive Tasks*. *06*(03), 108–112.

Galli, S. (2023, March 29). *Overcoming Class Imbalance with SMOTE: How to Tackle Imbalanced Datasets in Machine Learning*. https://www.blog.trainindata.com/overcoming-class-imbalance-with-smote/

Gholamy, A., Kreinovich, V., & Kosheleva, O. (2018). *A Pedagogical Explanation A Pedagogical Explanation Part of the Computer Sciences Commons*. https://scholarworks.utep.edu/cs_techrephttps://scholarworks.utep.edu/cs_techrep/1209

Guillen, M. D., Aparicio, J., & Esteve, M. (2022). *Gradient tree boosting and the estimation of production frontiers*. 957–4174. https://doi.org/10.1016/j.eswa.2022.119134

Hasanein, A. M., & Sobaih, A. E. E. (2023). Drivers and Consequences of ChatGPT Use in Higher Education: Key Stakeholder Perspectives. *European Journal of Investigation in Health, Psychology and Education*, *13*(11), 2599. https://doi.org/10.3390/EJIHPE13110181

Hayat Suhendar, M. T., & Widyani, Y. (2023). Machine Learning Application Development Guidelines Using CRISP-DM and Scrum Concept. *2023 IEEE International Conference on Data and Software Engineering (ICoDSE)*, 168–173. https://doi.org/10.1109/ICODSE59534.2023.10291438

Hedberg, N. (2020). *Automated invoice processing with machine learning: benefits, risks, and technical feasibility*.

Huilgol, P. (2023, December 21). *Precision and Recall | Essential Metrics for Machine Learning*. https://www.analyticsvidhya.com/blog/2020/09/precision-recall-machine-learning/

Islam Khan, A., & Al-Habsi, S. (2020). Machine Learning in Computer Vision. *Procedia Computer Science*, *167*, 1444–1451. https://doi.org/10.1016/j.procs.2020.03.355

Jierula, A., Wang, S., Oh, T. M., & Wang, P. (2021). Study on Accuracy Metrics for Evaluating the Predictions of Damage Locations in Deep Piles Using Artificial Neural Networks with Acoustic Emission Data. *Applied Sciences 2021, Vol. 11, Page 2314*, *11*(5), 2314. https://doi.org/10.3390/APP11052314

Johansson, S. (2022). *Classification of Purchase Invoices to Analytic Accounts with Machine Learning*. www.aalto.fi

Jones, T. (2017, December 3). *Unsupervised learning for data classification - IBM Developer*. https://developer.ibm.com/articles/cc-unsupervised-learning-data-classification/

Kontio, J., Lehtola, L., & Bragge, J. (2004). Using the focus group method in software engineering: Obtaining practitioner and user experiences. *Proceedings - 2004 International Symposium on Empirical Software Engineering, ISESE 2004*, 271–280. https://doi.org/10.1109/ISESE.2004.1334914

Kundu, R. (2022, December 16). *F1 Score in Machine Learning*. https://www.v7labs.com/blog/f1-score-guide

Kurama, V. (2023, February 28). *Regression in Machine Learning*. https://builtin.com/data-science/regression-machine-learning

Ligthart, A., Catal, C., & Tekinerdogan, B. (2021). Analyzing the effectiveness of semi-supervised learning approaches for opinion spam classification. *Applied Soft Computing Journal*, *101*, 107023. https://doi.org/10.1016/j.asoc.2020.107023

Lones, M. A. (2021). *How to avoid machine learning pitfalls: a guide for academic researchers*. http://www.macs.hw.ac.uk/

Madhumala, B. (2020). *Analysis of virtual Machine placement and optimization using Swarm intelligence Dayananda Sagar Institutions*. https://www.researchgate.net/publication/342183126

Martinez, J. (2024, February 26). *What Is The OCR Accuracy And How It Can Be Improved*. https://www.docuclipper.com/blog/ocr-accuracy/

Martinez-Plumed, F., Contreras-Ochando, L., Ferri, C., Hernandez-Orallo, J., Kull, M., Lachiche, N., Ramirez-Quintana, M. J., & Flach, P. (2021a). CRISP-DM Twenty Years Later: From Data Mining Processes to Data Science Trajectories. *IEEE Transactions on Knowledge and Data Engineering*, *33*(8). https://doi.org/10.1109/TKDE.2019.2962680

Martinez-Plumed, F., Contreras-Ochando, L., Ferri, C., Hernandez-Orallo, J., Kull, M., Lachiche, N., Ramirez-Quintana, M. J., & Flach, P. (2021b). CRISP-DM Twenty Years Later: From Data Mining Processes to Data Science Trajectories. *IEEE Transactions on Knowledge and Data Engineering*, *33*(8), 3048–3061. https://doi.org/10.1109/TKDE.2019.2962680

Mauro, M., Sivamayil, K., Rajasekar, E., Aljafari, B., Nikolovski, S., Vairavasundaram, S., & Vairavasundaram, I. (2023). *A Systematic Study on Reinforcement Learning Based Applications*. https://doi.org/10.3390/en16031512

Mehryar Mohri, Afshin Rostamizadeh, & Ameet Talwalkar. (2018). *Foundations of Machine Learning*. http://mitpress.mit.edu/9780262039406/

Mishra, S. (2017, May 20). *Unsupervised Learning and Data Clustering* . https://towardsdatascience.com/unsupervised-learning-and-data-clustering-eeecb78b422a

Mwiti, D. (2023, September 1). *Real-Life Applications of Reinforcement Learning*. https://neptune.ai/blog/reinforcement-learning-applications

Nadali, A., Kakhky, E. N., & Nosratabadi, H. E. (2011). Evaluating the success level of data mining projects based on CRISP-DM methodology by a Fuzzy expert system. *ICECT 2011 - 2011 3rd International Conference on Electronics Computer Technology*, *6*, 161–165. https://doi.org/10.1109/ICECTECH.2011.5942073

Naeem, S., Ali, A., Anam, S., & Ahmed, M. M. (2023). An Unsupervised Machine Learning Algorithms: Comprehensive Review. *International Journal of Computing and Digital Systems*, *13*(1), 911–921. https://doi.org/10.12785/IJCDS/130172

Nichols, J. A., Herbert Chan, H. W., & Baker, M. A. B. (2019). Machine learning: applications of artificial intelligence to imaging and diagnosis. *Biophysical Reviews*, *11*(1), 111. https://doi.org/10.1007/S12551-018-0449-9

Oyelekan, B. (2023, March 27). *Optimizing Database Interactions in Python: SQLAlchemy Best Practices — Soshace • Soshace*. https://soshace.com/optimizing-database-interactions-in-python-sqlalchemy-best-practices/

Pitafi, S., Anwar, T., & Sharif, Z. (2023). A Taxonomy of Machine Learning Clustering Algorithms, Challenges, and Future Realms. *Applied Sciences 2023, Vol. 13, Page 3529*, *13*(6), 3529. https://doi.org/10.3390/APP13063529

Rani, S., & Singh, J. (2018). Enhancing levenshtein's edit distance algorithm for evaluating document similarity. *Communications in Computer and Information Science*, *805*, 72–80. https://doi.org/10.1007/978-981-13-0755-3_6/FIGURES/10

Raschka, S., Patterson, J., & Nolet, C. (2020). Machine Learning in Python: Main Developments and Technology Trends in Data Science, Machine Learning, and Artificial Intelligence. *Information 2020, Vol. 11, Page 193*, *11*(4), 193. https://doi.org/10.3390/INFO11040193

Rayhan, A. (2023). *THE FUTURE OF WORK: HOW AI AND AUTOMATION WILL TRANSFORM INDUSTRIES*.

Ribeiro, J., Lima, R., Eckhardt, T., & Paiva, S. (2021). Robotic Process Automation and Artificial Intelligence in Industry 4.0 – A Literature review. *Procedia Computer Science*, *181*, 51–58. https://doi.org/10.1016/J.PROCS.2021.01.104

Rymarczyk, T., Kozłowski, E., Kłosowski, G., & Niderla, K. (2019). Logistic Regression for Machine Learning in Process Tomography. *Sensors (Basel, Switzerland)*, *19*(15). https://doi.org/10.3390/S19153400

Saltz, J. S. (2021). CRISP-DM for Data Science: Strengths, Weaknesses and Potential Next Steps. *Proceedings - 2021 IEEE International Conference on Big Data, Big Data 2021*, 2337–2344. https://doi.org/10.1109/BIGDATA52589.2021.9671634

Sarker, I. H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Computer Science*, *2*, 160. https://doi.org/10.1007/s42979-021-00592-x

Saxena, D. (2023, November 7). *Machine Learning vs Rule-Based Systems* . https://www.socure.com/blog/machine-learning-vs-rule-based-systems

Schonlau, M., & Zou, R. Y. (2020). The random forest algorithm for statistical learning. *Stata Journal*, *20*(1), 3–29. https://doi.org/10.1177/1536867X20909688

Scikit-Learn. (2024, March 21). *API Reference — scikit-learn 1.4.1 documentation*. https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics

Shahzadi, N. (2023, June 29). *Supervised Machine Learning: Classification and Regression | by Nimra Shahzadi | Medium*. https://medium.com/@nimrashahzadisa064/supervised-machine-learning-classification-and-regression-c145129225f8

Smith, C. S. (2023). *What Is OCR (Optical Character Recognition) Technology?* https://www.forbes.com/sites/technology/article/what-is-ocr-technology/

Studer, S., Bui, T. B., Drescher, C., Hanuschkin, A., Winkler, L., Peters, S., & Müller, K. R. (2020). Towards CRISP-ML(Q): A Machine Learning Process Model with

Quality Assurance Methodology. *Machine Learning and Knowledge Extraction*, *3*(2), 392–413. https://doi.org/10.3390/make3020020

Timalsina, A. (2023, September 6). *Analysis and Benchmarking of OCR Accuracy for Data Extraction Models*. https://www.docsumo.com/blog/ocr-accuracy

Ting, K. M. (2011). Confusion Matrix. *Encyclopedia of Machine Learning*, 209–209. https://doi.org/10.1007/978-0-387-30164-8_157

Tools, M., Rimal, Y., Sharma, N., & Alsadoon, A. (2023). *The accuracy of machine learning models relies on hyperparameter tuning: student result classification using random forest, randomized search, grid search, bayesian, genetic, and optuna algorithms Random forest · Educational data mining · Bayesian · Randomized search cross validation · Automate hyperparameter tuning · Tree-based pipeline optimization tool · Characteristic area under the curve*. https://doi.org/10.1007/s11042-024-18426-2

Tripathi, S., Muhr, D., Brunner, M., Jodlbauer, H., Dehmer, M., & Emmert-Streib, F. (2021). Ensuring the Robustness and Reliability of Data-Driven Knowledge Discovery Models in Production and Manufacturing. *Frontiers in Artificial Intelligence*, *4*. https://doi.org/10.3389/FRAI.2021.576892/FULL

van Engelen, J. E., & Hoos, H. H. (2020). A survey on semi-supervised learning. *Machine Learning*, *109*(2), 373–440. https://doi.org/10.1007/S10994-019-05855-6/FIGURES/5

van Engelen, J. E., Hoos, H. H., Fawcett Jesper E van Engelen, T. B., & Hoos hh, H. H. (2020). A survey on semi-supervised learning. *Machine Learning*, *109*, 373–440. https://doi.org/10.1007/s10994-019-05855-6

Warnell, G., Fitzgerald, T., Najar, A., & Chetouani, M. (2021). Reinforcement Learning With Human Advice: A Survey. *Frontiers in Robotics and AI | Www.Frontiersin.Org*, *1*, 584075. https://doi.org/10.3389/frobt.2021.584075

Werner de Vargas, V., Schneider Aranda, J. A., dos Santos Costa, R., da Silva Pereira, P. R., & Victória Barbosa, J. L. (2023). Imbalanced data preprocessing techniques for machine learning: a systematic mapping study. *Knowledge and Information Systems*, *65*(1), 31. https://doi.org/10.1007/S10115-022-01772-8

Wijaya, C. (2021, April 25). *CRISP-DM Methodology For Your First Data Science Project*. https://towardsdatascience.com/crisp-dm-methodology-for-your-first-data-science-project-769f35e0346c

Wirth, R., & Hipp, J. (2000). *CRISP-DM: Towards a Standard Process Model for Data Mining*. http://cs.unibo.it/~danilo.montesi/CBD/Beatriz/10.1.1.198.5133.pdf

Appendices

Appendix 1: Brief Summary of Different Classification Metrics

Metrics	Formula	Description	Example
Precision (Brownlee, 2020)	$\frac{True\ Positive}{True\ Positive + False\ Positive}$	Calculating by dividing the correctly predicted positive observations by the total predicted positive observations. Measured on a scale from 1 to 0 where 1 is a perfect score	How often the model predicted a observation to be PASS out of all the actual PASS observations in the dataset
Accuracy (Jierula et al., 2021)	$\frac{Correct\ Predictions}{Total\ Number\ of\ Predictions}$	Calculated by dividing the total number of correct predictions by the total number of predictions. Measured on a scale from 1 to 0 where 1 is a perfect score	How often the model was able to correctly predict which observations were PASS out of all the predictions made
Recall (Huilgol, 2023)	$\frac{True\ Positive}{True\ Positive + False\ Negative}$	Calculated by dividing the total number of true positives by all of the actual positive observations. Measured on a scale from 1 to 0 where 1 is a perfect score	For all the PASS observations, recall tells how many the model correctly identified
F1 Score (Kundu, 2022)	$2 * \frac{Precision * Recall}{Precision + Recall}$	The weighted average of Precision and Recall where both variables are important. Measured on a scale from 1 to 0 where 1 is a perfect score	In case a PASS observation is predicted to be UD it could be a sign that the observation might have passed when it shouldn't have

Appendix 2: Machine Learning Types and Usage Examples

Algorithm	Use Cases	Strength	Weaknesses
Classification (Black et al., 2023)	Profiling patients' disease risk, clinical support, fraud detection	Can outperform traditional statistical models when it comes to predictive performance and be more flexible when it comes to unstructured data.	Requires larger datasets compared to traditional statistical models e.g., while a ML model might require 200 events per candidate, a statistical model only requires 20. More often prone to bias due to errors in model design or biased data.
Regression (Kurama, 2023)	Temperature prediction, stock market prediction	Able to establish and determine relationships among different variables. Linear models can be easily updated with new data.	Poor performance when there are non-linear relationships in the dataset as the model is not as flexible in capturing complex patterns requiring additional polynomials or interaction terms which can be difficult to get correct and high variance can lead to poor prediction performance.
Clustering (Aravind, 2023; Pitafi et al., 2023)	Finding patterns and groups within areas such as marketing, politics, ecology, and genetics	Dataset does not have to be labeled and give an understanding of a dataset as clustering algorithms can group the observations in the dataset.	In cluster analysis, determining the correct number of clusters beforehand is difficult due to factors such as no prior experience with the data i.e., not knowing what kind of groups to be expected and the varying sizes, forms, and densities of groups inside a dataset also make it difficult to draw conclusions.

Appendix 3: Hyperparameters Tuned During the Development

Hyperparameter	Description
n_estimators	Defines the numbers of trees in the forest, while for GBC this defines the number of boosting stages. For GBC, a larger number usually results in better performance as GBC is fairly robust to overfitting.
max_depth	Defines the depth of the tree. If None is specified, it expands until all leaves contain less samples than min_samples_split or until leaves are pure. For GBC, this parameter defines the maximum depth limit the number of nodes in the tree and controls the tree complexity.
min_samples_split	The minimum numbers of samples required to split an internal node. If the number of samples in the node is less than specified, the node will be a leaf.
min_samples_leaf	The minimum numbers of samples required to create a leaf node. Before generating the node, this parameter will evaluate whether a potential split will create a child node with fewer samples than specified in the min_samples_split parameter. If it does, the split will be avoided.
max_features	Defines the number of features to consider when determining the best split. Can be used to control overfitting.
learning_rate	Defines how each tree impacts the final outcome. A smaller value might require more trees but can result in a better performance.
n_neighbours	Defines the K value i.e., the number of neighbours to use for the majority voting.
weights	Defines the weight function to decide an outcome. Can be either uniform (all points weigh equal) or distance (the closer the neighbour the heavier the weigh)
algorithm	Defines the algorithm to use when computing the nearest neighbours. Can be either 'auto', 'brute', 'kd_tree' or 'ball_tree'
leaf_size	If kd_tree or ball_tree is selected, this parameter can affect the speed of the prediction as well as memory requirements.
p	Defines the metric to use which is the Minkowski Distance by default. If p = 1, the Manhattan Distance will be used and if p = 2, the Euclidean Distance will be used.

Appendix 4: Summary of Different Types and their Characteristics

Learning Type	Features	Use Cases	Strengths	Weaknesses
Supervised (Crisci et al., 2012)	Uses labelled data to make predictions according to what is expected.	Spam detection, sales forecast, image recognition	Useful in a wide range of fields as it is possible to make accurate predictions and to generalize on patterns.	Requires sufficient amount of labelled and structured data to be able to accurately make predictions which can be time consuming and/or expensive to acquire.
Unsupervised (Naeem et al., 2023)	Can find patterns and group data previously unknown	Social network analysis, customer segmentation	Uses unlabelled data to make the predictions and groupings, useful when conducting an exploratory analysis	Results can be difficult to evaluate as results might not be accurate
Reinforcement (AlMahamid & Grolinger, 2022)	Decides the next action based on the policies made through trial and error	Game AI, robot navigation, self-driving cars	Learns from doing, has the capability of improving as time passes	Requires a logical reward and punishment system, requires more computational resources and time to achieve results.
Semi-Supervised (van Engelen & Hoos, 2020)	Uses both labelled and unlabelled data	Disease progress prediction, medical image analysis, web content classification	Useful when the amount of unlabelled data is abundant, but the labelled data is not	Unlabelled data potentially degrades the performance of the models.