UPTEC STS 22001
Examensarbete 30 hp
Januari 2022

UPPSALA
UNIVERSITET

An AI approach for quality improvement in heat treatment processing

Gustav Kruse
Lotta Åhag

Civilingenjörsprogrammet i system i teknik och samhälle

UPPSALA UNIVERSITET

An AI approach for quality improvement in heat treatment processing

Gustav Kruse
Lotta Åhag

Abstract

Export of heat treated steel goods has an important impact on the Swedish economy which brings performance demands and expectations on production to keep a competitive market position. Sustainability and efficiency are two important aspects in meeting these demands. This thesis studies how a data driven approach can be used to increase efficiency in manufacturing of rods produced for the mining industry.

The purpose of this thesis is to use a machine learning model suitable for classifying quality results for heat treated steel rods. This is done by comparing nine algorithms with the objective to tune and deploy the model best fitted while gaining insights in variables that have an impact on the quality output.

This thesis outset is a heat treatment process at Epirocs facility in Fagersta. Interviews are conducted to gain domain knowledge about important features and an AI pipeline is implemented to demonstrate its suitability for predicting quality given production and weather data in the form of time series and product-unique data points.

The result of the study shows that the machine learning algorithm random forest is indicated as most suitable among the analyzed. The study also shows that an AI pipeline with streaming data can be designed and efficiently implemented for quality improvement. Through this work, the authors have proved that machine learning can be used to improve the heat treatment process of rods, but the model still has room for improvement in feature selection and availability of larger and more detailed data at the facility.

Teknisk-naturvetenskapliga fakulteten

Uppsala universitet, Utgivningsort Uppsala

Handledare: Jonas Jern (Epiroc) Ämnesgranskare: Kaveh Amouzgar

Examinator: Elísabet Andrésdóttir

Populärvetenskaplig Sammanfattning

Användning av artificiell intelligens inom tillverkningsindustrin är ett relativt nytt fenomen då få företag har lyckats ta steget från konceptvalidering till produktionssättning. Det har dels att göra med att kunskapen om vilken data som är viktigt ännu inte fastställts då många tillverkningsprocesser är av hög komplexitet. Det innebär också att implementationen av en skalbar struktur inte haft ett investeringsbart underlag. Samtidigt genomgår Svensk industri en fjärde industriell revolution som strävar efter högre automationsgrad och utökad digitalisering. En utveckling som är vital för att möta global konkurrens.

Tidigare forskning har visat på att tidsserier och processvariabler kan analyseras på andra sätt än att se dem visuellt i tabell, graf eller jämföra dem en och en. Avvikelser kan förekomma i mer komplexa mönster där flertalet variabler i viss kombination har inverkan eller att en viss sekvens av data är mer betydande än en datapunkt. Med hjälp av maskininlärning kan man finna komplicerade mönster i data som avslöjar sambandet mellan ingående variabler i processen och hur utfallet utspelar sig. Detta är samband som inte en människa kan uppfatta. Underliggande mönster kan upptäckas med maskininlärning under förutsättning att rätt variabler används som indata, vilket också visar på vikten av domänkunskap om den specifika processen som undersöks.

I projekt som grundas på stora datamängder är bra orkestrering av arkitektur och datastruktur viktigt för att kunna utveckla, skala upp och produktionssätta modeller på ett effektivt sätt. Detta kan förverkligas genom en pipeline som kopplar ihop olika processteg och sparar en version av datats tillstånd mellan respektive steg. På så sätt kan en pipeline kontinuerligt matas med ny data till en maskininlärningsmodell. För utveckling eller forskning kring andra produktionsområden kan samma pipeline användas genom att koppla på en ny pipeline från valfritt processteg utan att behöva processa datat från början. Därav finns det stora fördelar i att kunna nyttja en pipeline för tillverkning då ny data kontinuerligt genereras.

Epiroc är ett verkstadsföretag som tillverkar utrustning för gruv- och infrastrukturindustrin. Denna utrustning innefattar bland annat borrkronor och stänger som fästs vid varandra och används under exempelvis malmbrytning. Genom att borra hål i berget ges plats åt dynamit som expanderar tomrummet. Dessa stålstänger tillverkas i Fagersta genom en mycket komplicerad process som ger stålet rätt egenskaper i form av seghet, hårdhet och kolhalt. Stängernas kärnhårdhet sätts i en värmebehandlingsprocess där stålet värms upp till cirka 1000 grader Celsius under en viss tidsperiod och kyls med hjälp av fläktar i en viss hastighet givet ett recept för den specifika produkten. Stängerna testas sedan av en kvalitétsavdelning som märker produkten som godkänd eller ej godkänd givet ett kvalitétsmått innan de skickas vidare till nästa operation eller kund. Idag tittar operatörer på avkylningskurvor i realtid som visas på en monitor för att se hur en värmebehandling ter sig. De kollar även på kvalitétsresultat och gör utifrån dessa en bedömning av hur produktrecept ska ändras för att optimera kvalitéten och minimera

skrotade produkter. Företagets ingenjörer försöker på så vis manuellt optimera kvalité genom att testa olika recept för körningarna. I detta examensarbete utforskas möjligheten att nyttja en pipeline för att genom maskininlärning utforska variabler och prediktera kvalitetsutfall på värmebehandlade stålstänger.

Resultatet visar på att pipelinen och de tjänster som nyttjas har möjliggjort värdeskapande insikter under kort implementationstid. Nio maskininlärningsalgoritmer har jämförts med sex olika prestandamått och detaljerade visualiseringar för respektive variabels inverkan på kvalitétprediktionen. Den slutgiltiga maskininlärningsalgoritmen uppvisade ett viktat AUC värde på 0.71 genom korsvalidering och predikterade 61 procent rätt vid test på osedd data. Modellen har utvecklingspotential om mer data med högre granularitet tillgängliggörs. Resultatet kan i huvudsyfte användas till att minska kostnader och effektivisera arbetet genom att minska antalet kvalitétstester. Den framtagna maskininlärningsmodellen visar också på att utomhustemperaturen, avkylningsfläkt och den totala värmeenergin i knippet av stänger är de viktigaste av de utvalda variablerna för att prediktera kvalitétsutfall. Det ger en intressant grund till vidare forskning och vilken typ av data som behövs för att kunna ta fram anpassade recept som minskar kassationer. Studien har även bekräftat att fläktar är individer som ger olika resultat på samma variabler, vilket medför att framtida forskning måste ta hänsyn till produktionspecifika variabler och de maskiner som används för respektive produktionslinje.

De slutsatser som tagits bidrar till en djupare förståelse för vilka faktorer som påverkar en värmebehandlingsprocess under blåshärdning och kan potentiellt bidra till en ökad produktivitet, högre kvalité och minskade kassationer. En effektivare produktionsmiljö bidrar till större ekonomiskt värdeskapande samtidigt som energin som processen kräver kan minskas. Studien har även undersökt huruvida en pipeline för artificiell intelligens kan utvecklas och implementeras för att prediktera kvalitétsutfall. Studiens resultat pekar på att så delvis är fallet hos fallföretaget. Att dokumentera vilken data som bidrar till stängers hårdhet samt att utröna huruvida data kan användas för att fatta AI-drivna beslut stärker Sveriges position på världsmarknaden.

Acknowledgements

This master thesis has been conducted for Uppsala University in collaboration with Epiroc, a steel manufacturing company in the mining and construction industry located in Fagersta.

We would like to give special thanks to our supervisor Jonas Jern, who has provided a significant network with all competencies and support needed for this project's progress. Without his vision of a data-driven factory, this thesis would not have been possible.

We also want to thank all co-workers at Epiroc who have taken their time to give us insight in the heat treatment process and participated in feedback sessions and interviews. The domain knowledge gained from these sessions is the reason for such interesting results. Also architectural implementation and weekly support from Joakim Åström at Microsoft has been vital for this thesis comprehensive scope.

Finally, we want to send our gratitude to our subject reader Kaveh Amouzgar, who contributed with valuable feedback and support along the course of this project.

Table of contents

1. Introduction

Sweden's manufacturing industry stands for roughly 40 percent of the country's total export [1] and around 16.3 million tons of CO_2 emissions a year. This makes the sector one of the biggest players in the shift towards a more sustainable future [2]. Steel is a vital component in many of Sweden's production sites and the steel manufacturing industry is an important key player in reducing emissions. Many products are made of steel, for example aircraft components and car engines [3]. Machined steel often undergoes some kind of property refining process which aims to give the right characteristics to the steel through heat treatment and cooling. It is important to have control over all production steps to maintain high quality, reduce costs and deliver the steel goods on time [4]. A refined manufacturing process contributes to better resource utilization and lower emissions [5].

Today's manufacturing industry is experiencing a rapid increase of available data. A vast amount of data is being collected through the whole production line from sensors, machines and other data collecting features [6]. Data related to quality output has potential to improve the monitoring of quality output [7]. According to the European commission, the "factories of the future" need to deal with a higher competition from global competitors and one strategy to do so is to incorporate new technologies, services and applications [8]. Extract, handle and analyze data are keys in this transformation. This means that not only the development of machine learning (ML) algorithms is important, but also implementing effective orchestration that handles the entire flow from raw process data to deployment of a model [9]. This orchestration is also called an artificial intelligence (AI) pipeline.

ML can be a part in solving today's manufacturing challenges with big and complex data as the raw process data does not provide any information itself [10]. The ground for solving these challenges is to use qualitative and quantitative approaches by using suitable tools for ingestion, storage and processing of data that enables ML and new insights. According to Wuest et al. [11], data driven solutions can identify nonlinear relations by "Transforming raw data to feature spaces, so called models". These models can then be applied to different problems within forecasting, regression, prediction, detection and classification. When manufacturing rock drilling equipment, the products undergo a certain predetermined process that immediately becomes complex. When the products are heat-treated, several variables are generated which become stochastic.

Variables can be collected from a process over time as a time series of sensor measurements. In such cases, data models and tailored algorithms for time series can be helpful to find causes of rejection and predict quality [12]. These algorithms can be tailored to quality targets and compared to find the most appropriate algorithm for the purpose [13]. When working with time series a set of variables can be extracted, preferably based on domain knowledge [10], [14]. The variables can then be categorized

into classes with the help of ML [15] and ergo be detected as anomalies in the process with regards to the quality output. The research area is unexplored within rock drilling equipment and lacks a deeper understanding of ML applications.

1.1 Purpose

This thesis aims to solve an issue connected to heat treatment of steel part manufacturing and quality outcome for rock drilling tools. The purpose is to transform raw process data and find the optimal ML algorithm that is best suited for the heat treatment process. Several varieties of algorithms are tested to find patterns between the process and its outcome, meanwhile scoring the model to prove its correctness.

Through this work, the authors want to prove that raw process data and quality results can be structured and combined in improving and predicting quality results through an AI pipeline. To do this, the right variables must be extracted from the time series in order to match other process data. The main goal is to first prove that ML can be used in a manufacturing environment and, if proven, use ML to indicate quality results and variables that have an impact on it.

The goal of this project is to answer the following questions in production of heat treated steel rods for rock drilling:

1) *What data has an impact on hardness results from the cooling process?*

2) *How can a tailored AI pipeline be used to predict quality output based on data related to the cooling process?*

1.2 Delimitations

In this study, several delimitations have been made with the purpose of making it more direct and concrete. The focus is to prove the potential of AI in rock tool manufacturing through a case study. The study does not go into details about measuring equipment in the lab and sensors during each process step. The data used only comes from three sources: a quality lab, a production database, and external temperature data. Due to limitations in time, the study does not go into details about the entire production flow, that is, before and after the heating and cooling process. Thus, this study is selecting only data for a part of the process, to investigate whether it has an impact on the quality or not.

2. Background

Section 2.1 gives a brief introduction to heat treatment of steel and mandatory process steps for steel to gain certain characteristics. This is followed by an introduction to ML and its potential in steel manufacturing procedures in Section 2.2 and how to utilize time series for anomaly detection in Section 2.3 and 2.4. The advantages of visualizing machine learning results are discussed in Section 2.5.

2.1 Heat treatment of metal

Heat treatment is a process that is defined by heating metal to a critical temperature and in most cases rapidly cool it by gas, air, oil, or water. The process transforms the internal structure in the material and mechanical characteristics. Hardening is one form of heat treatment which sets a material's hardness, wear resistance and hot-working ability. The process of heat treatment involves three steps to acquire the right characteristics. The metal must first be heated to a certain temperature and then kept in a certain range of temperature for a specified amount of time. Lastly the metal is carefully cooled at a specified rate to get a stabilized grain structure. This is a vital step in all heat treatment processes since unsuccessful cooling can cause cracks and chips in the material [16].

2.2 ML potential in manufacturing

ML is an advanced way of processing data to get deeper insights. Various kinds of ML techniques can uncover non-linear and overly complex patterns in several types of data [11]. One general possibility of ML techniques is the ability of handling advanced problems that often occur in modern production environments [17]. These problems can be solved with troubleshooting, control and optimization where the ML models play a huge role in finding solutions [18]. ML is applicable in several perspectives of manufacturing which all play a significant role in daily business operations. It can result in a competitive position on the market, reducing production costs and limiting environmental impacts [18]], [[19]. Companies can innovate in manufacturing efficiency by more advanced process control and forecasting maintenance. By enabling better data insights through ML, industries can reduce waste, energy usage and carbon emissions. Products can also be more reliable manufactured and sold with increased quality [19].

To increase ML's potential in manufacturing, the flow of data can be orchestrated in an AI pipeline, which accelerates the process of taking raw data to tuned ML models. An AI pipeline automates the process to both save time, money and get more consistent and reliable models. The pipeline can be seen as a recurring cycle that begins with data ingestion, data validation and data preprocessing. The purpose of the ingestion is to transform the data into a uniform format that works with all following components in the process. The data is then validated by examining its statistics and distribution. This includes investigating any abnormal values or imbalance between class labels and applying suitable actions for it. The data preprocessing step is cleaning the data to be

ready for model training and tuning. After the model training, the AI pipeline cycle ends with model analysis and validation, model deployment and model feedback. Model analysis is when a data scientist carries out a deeper investigation of the model performance with different performance metrics to make sure that the model makes fair predictions. When the model is trained and tuned, it is deployed to be used on fresh and unseen data. The last step of the AI pipeline, model feedback, is meant to check the real effectiveness and performance of the model and add more data or update the model if improvements are needed. The advantage of an AI pipeline is that the entire pipeline can be automated except for the analysis and feedback step. This makes room for data scientists to focus on development instead of maintenance [9].

2.3 Feature extraction from time series data

Time series is a common way of collecting data over time [20] and is a crucial part of production follow up and control. Time series has for example been used to improve manufacturing control system performances [21]. Using ML for quality improvement in manufacturing is a common way to utilize time series data. These techniques can be devoted to find the relations between the input parameters and outcome [22]. ML algorithms are divided into two kinds of groups, supervised and unsupervised [23]. If instances are given with known labels, the learning is supervised [24]. Supervised algorithms aim to create a model from a known dataset [25] which is used to improve production output and quality control [26].

Supervised learning is used on classification problems, as the output is to be defined by a class label. When handling time series data, or any other type of classification data, feature extraction or feature selection is conducted to reduce the number of dimensions as a first step before modeling a classification algorithm. Feature extraction is the process of constructing new feature spaces out of the initial feature spaces to reduce dimensionality. Feature selection is selecting a subset of features that are of relevance for the analysis and minimizes redundancy. The feature selection is important as irrelevant features result in poor models when training the model [27].

Most literature that proposes time series classification is application dependent, which indicates that algorithms developed for the widespread use case also come with poorer performance. Thus, a few features extracted based on domain knowledge are preferred above hundreds of random features as it can cause overfitting on a small dataset. Time series are defined by a continuous series consisting of a contextual attribute, time, and a behavioral attribute which is the corresponding value at the given time. Time series data can either be classified by a specific time-instant or as a part of the whole series and both types of measures can be equally important in a classification model. Measurement errors can easily cause worse classification and although some algorithms take outliers or anomalies into account, they can still cause effects on accuracy. The effect of this must be considered when analyzing data from time series [15].

2.4 Anomaly detection in time series

For detection of anomalies in time series data, assumption must be made that the normal behavior of a process is stationary to find the occasions when the process generating data is abnormal. This means that the process must be stationary also in the future if prediction is going to be applied. For manufacturing, anomaly detection is of significant importance as anomalies need to be found at an early stage for expensive operations. Manufacturing industries have for a long time checked the quality by using algorithms for change detection from sensor data, which is considered a straightforward way of detecting anomalies. A lot of time series are generated from manufacturing processes, which makes it interesting to compare how they differ from each other instead of changes within each time series. Time series for the same type of process may also vary depending on what is produced and thus follow different target values [14].

Research on anomaly detection in time series has increased in the last decade for a diverse set of fields. Anomalies are commonly either a point anomaly or a structured anomaly. Point anomaly is when a single point deviates from the other in a series and thus becomes isolated. A structured anomaly is described by a set of points that differs in a comparison with another set of points and is often more complex to handle. It is important to reveal the data structure when solving structured anomaly problems and a lot of today's algorithms within the field are focusing on exactly that part for the analysis. One popular method is to use interval sets theory, meaning to divide the time series into segments according to upper and lower bounds for qualitative information analysis. Previous research has divided the time series into equal sized subsequences and then explored the bounds of each interval to extract distribution information from a single point. These points can then be used in finding the largest distances which are to be counted as anomalies [28]. Two other common ways of anomaly detection for multiple time series are point-to-point distances and variations over time. Point-to-point distances are comparing the distance of one point to the corresponding point in another time series. Variations over time take the gradient of a sequence into account, comparing the derivative for a certain time period with another time series. In this way many values can be generated from local gradients. Anomalies can occur in a specific sequence of a time series, or affect the whole process. Thus it is important to find the sequences that are of importance for the quality output [14].

2.5 Interpretation of ML insights by visualizations

A frequent problem with proof of scientific concepts is that they rarely make it to a production environment with the whole pipeline from data extraction to visualization. The visualizations are important to interpret scientific findings and explain the added value of the model [29]. Thus, visualizations are knowledge generators as it helps the user to find hypotheses about a ML model's output. The ML domain has three main purposes that characterize the benefits gained from visualizations. The first one is eventual incompleteness in understanding the problem itself. By using visualizations, data can be understood in a more holistic way and reveal other perspectives of a problem. The second one is the diagnosis of the result, which either corresponds to expectations or not. Through diagnosis of the result, the user can compare visualizations containing already gained knowledge. The third characteristic is refinement, by which better models can be designed. By understanding the output and compare it with the optimal output, the model can be iterated and become better [30].

3. Related work

This section gives a summary of related work and previous research within the area of heat treatment in combination with AI and quality improvement in manufacturing. Section 3.1 presents the results on the applicability of AI in steel manufacturing processes followed by examples of types of research done in the area for quality improvement in Section 3.2. Research that is closely related to this report's analytical outset is presented in Section 3.3 and ends with a discussion on how this report contributes to current research in the mining and manufacturing field in Section 3.4.

3.1 AI in steel manufacturing

The complexity of steelmaking and the multitude of production chains that generate process data makes the industry a perfect candidate for AI research and implementation gains [31]. The data in combination with the latest information technologies is the core part of future smart factories, thus a lot of research has been done in the fields of steel manufacturing and process improvements [10].

Wuest et al. [11] means that neither steelmaking nor the general manufacturing industry has yet embraced, and accepted architectures and applications based on cloud computing, partly because of difficulties to take it to a production state. Pellegrini et al. [10] conducted research based on implementation of a pipeline concept on different processes that are applicable for AI and the next generation of manufacturing. The research is based on a ML adoptable architecture that supports cloud modules to extract features from various sources of raw data and store them in a uniform format in a standardized way for horizontal and vertical scalability. The architecture also supports data mining and visualization for predictive and monitoring purposes. Throughout the research Pellegrini et al. presents three different use cases of the architecture to prove its value for steel manufacturing. First it is used as a decision support tool for operators based on binary classification prediction of clogging probability in continuous casting. It is also used to show real time steel temperature during a degassing process and lastly for detecting surface defects with deep learning for image recognition. As a conclusion, Pellegrini et al. concludes that one of the main advantages of the cloud-based architecture is that it supports heavy workloads of for example image processing and reduces the initial costs of hardware. The results show that the architecture has many more application areas and can give results that have an immediate impact on the industry in quality precision and operational costs.

Cemernek et al. [32] has investigated current ML techniques for the continuous casting process of steel with a vast review of existing publications. The findings show that prediction of steel quality and defects needs consideration to the full process and that decision trees and neural networks make the ground for most applicable algorithms. When predicting quality, a more diverse set of target variables can be involved as quality is an umbrella term referring to distinctive characteristics such as hardness or tensile

strength. Due to this, quality prediction research is much more heterogenic in its models and applications as different variables are useful for different quality measures. The research concludes that supervised and active learning would be of beneficial use in the steel industry if new techniques are implemented to handle imbalanced data [32].

3.2 Quality improvement in steel manufacturing

A lot of research has been done in the area of quality improvement and surface defects is one of the most common use cases of ML in the steel industry [10]. Published studies that research on AI for heat treatment processes mostly aim to develop systems that use AI in real time to simulate human behavior or develop decision support functions in processes that involve human interaction of detecting defects [34]. Many of these studies are based on images processing algorithms [39] while others are based on mathematical co-relations between ingoing parameters and known output quality parameters gathered in a knowledge base [34]. For example, Mitra et al. [34] investigates furnace temperature, material thickness, weight and steel grade to predict furnace temperature for optimal final carbon content, hardness, ductility, formability and tensile strength. Other research done by, for example Tsutsui et al. [42], Panda et al. [43] and DeCost et al. [44] are looking at the physical characteristics of the steel and using control parameters extracted from sensors such as images or processing data for temperature and time in the furnace. Previous studies take the materials composition and its predicted mechanical properties into account to find optimal recipes for the heat treatment process. Variables for these types of studies are thus not only based on collected process data, but also data that specifies how the characteristics of the steel should be worked with scientifically [42].

3.3 Heat treatment analysis for quality improvement

The common denominator of research focused on heat treatment is to use deep neural networks or linear regression to develop prediction models or methodologies for the general steel product [45]. One example is Carneiro et al. research which has, likewise this thesis, investigated how to predict quality outcomes to minimize production line bottlenecks such as quality tests. Carneiro et al. investigates steel tubes with neural network and tree ensemble methods on water quenched steel with an unsupervised approach. The research differs from previous research by examining a process that involves data from a quenching tank and looking at how the water flow and pressure impacts the quality. The results show that ML techniques must be investigated in conjunction with variable selection for each use case. This is because the different quality parameters such as tensile strength, hardness and yield strength are affected by different input parameters and are ultimately predicted by different algorithms [37].

Another study that predicts quality on yield strength and tensile strength is Xie et al. [48] who uses deep learning on raw steel parameters and process data from the reheat furnace process, rolling data and water-cooling data at a steel plant. The cooling data is from average cooling rate, start and finish cooling temperature out of which all measurements

between 200 and 900 degrees are included. The research includes 27 in parameters and the model reaches an accuracy of 0.907 with deep learning. The research results in an online deployment of the model at the industry site with a graphical user interface to help operators manage the hot roll process parameters through predictive analysis [48].

Hanza et al. [49] predicts total hardness after continuous cooling of steel based on Artificial Neural Networks. The research is investigating whether chemical composition can be replaced as input variables by the Jominy distance. The Jominy distance value is correlated to a materials composition and defines its ability to harden. Values can be calculated through a formula that determines the distance based on hardness of steel with a microstructure of 50 percent martensite. Two tests are conducted where one includes chemical compositions and the other the Jominy distance value. The research results in that input data for the heat treatment temperature, heating time, cooling time down to 500 degrees Celcius and the Jominy distance gives almost as successful results predicting total hardness compared to the models with chemical composition included. Based on this, Hanza et al. draw the conclusion that only 4 input variables can predict the hardness which reduces the complexity of the model.

3.4 Knowledge gap

This study differs from existing publications in multiple ways. No studies are found within the area that also investigate the individual machines and variables that can be batch dependent, such as which cooler and furnace the product has passed in the production line. Focus lies on the process itself to prove how manufacturing industries can use ML to find flaws in their specific processes, but also investigate a subset of features and their impact on the quality. This study aims to classify results from air cooled steel rods for rock drilling, which has not been found in previous research. No previous studies have investigated the cooling process in terms of recipe dependent sequences from time series and the impact of the weather conditions on cooling. The focus for this thesis is on a specific part of the process to verify whether the chosen input parameters have an impact on certain product quality features or not. The research is delimited to mining and rock drill manufacturing's air-cooling process and more specifically to pit furnaces**.** No published research has investigated the correlation between input parameters and product quality with ML in the previously described way which indicates a scientific knowledge gap in the field. Thus, this is a contribution to the existing research of optimizing quality in heat treatment processes for the specific field of rock drilling manufacturing. Through this work a real-world application is found for the obtained model. The interest in understanding the process by using previously acquired knowledge in ML is tremendous in the research world. The question is not only how variables affect quality, but also how to implement solutions in a scalable way that supports further development efficiently. Through this work, the authors see a clear path to contribute to a deeper understanding in the area of rock drilling manufacturing and investigate the applicability of an AI pipeline to ease the work of gaining insights and putting them into production.

4. Methodology

Section 4.1 accounts for the chosen methodology and why it will enable a critical examination of the final results and the study's credibility. Section 4.1 will also describe how the research has been conducted to collect valid process data and assumptions about the case study and its process in Section 4.2.

4.1 Research strategy

The thesis aims to predict quality results for a specific heat treatment process by performing ML on production and quality data. Based on theories from time series and anomaly detection, feature extraction should be conducted in consideration to many factors. This pleads for the importance of domain knowledge to reach a result with a credible algorithm. With the right features from time series and knowledge about the process characteristics, one can get a total overview of the entire process from heat treatment to quality results and look at anomalies from a greater perspective.

The authors consider a case study to be the most appropriate way to explore the right data and extract the right features. Furthermore, a qualitative approach is suggested as the theory and decisions in data analysis is built upon individuals' domain knowledge about the process [50]. This thesis will investigate a steel manufacturing company, Epiroc, and their heat treatment process. The main purpose of a qualitative approach is to get domain knowledge about the processes and machines which are locally designed in order to extract the right features and train an appropriate ML algorithm.

To gain enough domain knowledge for making accurate decisions in data analysis, interviews are conducted on site. Semi-structured interviews are recommended by [50] to let the interview cover aspects and arguments that one might not have thought of beforehand. All selected respondents have a key relationship with the operative process of heat treatment or quality testing. The first interviews are focused on getting to know the production process by following the operators and obtaining an introduction to the characteristics of steel. Over time, interviews are more focused on the raw process data and possible variables to extract from it. See Appendix A for more information on the respondents.

4.2 Assumptions

It is both a strength and weakness that the choice of methodology is steered by the problem and specific process on site [51]. On one hand, conducting case studies comes with a risk of not getting to the depth of the problem due to secrecy or lack of trust from respondents. On the other hand, a complex and technical area that involves so many process steps and products require on site information to take as many aspects of the process into consideration as possible. In this thesis, assumptions are made that communication with stakeholders involved in the heat treatment process will be sufficient

to get enough understanding of the local heat treatment process for steel and the data collected to extract the most interesting features for the case study.

Some general assumptions are made in the investigation of the heat treatment process at the company. This is because the process is knowledge intensive, and a correct use case is crucial. This implies that the company should have a deeper understanding of heat treatment and know this process better than the average heat treatment company that serves the mining business with rod products. This will ease the work to find interesting and valuable variables for the study. Assumptions are also made that the heat treatment process and its generated data is representative for the business area.

5. Theory

This section presents relevant concepts and theories used for this project's implementation. In Section 5.1, nine ML algorithms for binary classification are defined followed by popular metrics in Section 5.2 for evaluating performance of models that are trained on imbalanced data. Section 5.2 also ends with two methods for visual model performance. Section 5.3 covers a discussion about imbalanced datasets and how to ensure fair partitions of train and test data.

5.1 Binary classification algorithms in AutoML

5.1.1 Bernoulli Naive Bayes

Naive Bayes is an algorithm that uses a classification approach which adopts the principles of Bayes theorem $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$ [52]. The theorem means that the presence of one variable does not affect the presence of another in the probability of an already given output. $P(B)$ is the probability of the evidence. $P(A)$ is the probability of hypothesis H and is known as the prior probability. $P(A|B)$ is the probability of the evidence given that the hypothesis is true. $P(B|A)$ is the probability of the given hypothesis that the evidence is true. The Bernoulli version is an effective algorithm for binary values [53].

5.1.2 Gradient boosting

Gradient descent is a method that tries to find the minimum point by checking the gradient step by step. When the gradient stops being negative, the minimum point is found. Gradient boosting is a sequential ensemble method that based on the gradient descent method finds the minimum errors of the residuals. The sequential learning makes the model smarter by not repeating a mistake twice and the model gets boosted by combining all weak trained models into one strong one [54].

5.1.3 Stochastic Gradient Descent

Stochastic Gradient Descent is a stochastic variant of the gradient descent model where a stochastic random variable is used to find the gradient's minimum point. The algorithm computes the gradient of the function with respect to each feature, then picking a random initial value for the parameters. The gradient function updates after inserting the parameter values and then adding parameters with respect to learning rate and step size repeatedly until the minimum value of the gradient is reached [55].

5.1.4 Linear support vector machine for binary classification

Support vector machines for binary classification are explained by a hyperplane that linearly separates two classes. The algorithm is optimized in separating classes by finding

the maximum distance between the closest training points of both classes. The points closest to the separating line are called the support vectors [56].

5.1.5 Decision Trees

Decision Trees is a rule-based method that partition values in the columns into disjoint regions, or branches. Thus, a certain set of feature values leads to certain branches and finally to a leaf node in the tree, which tells the predicted output class. The regions are decided with respect to a Gini index, which indicates the split with lowest error. A Gini index close to 0 tells that there are no errors, while a Gini index 0.5 is no better than a random guess [54].

5.1.6 Random forest

Random forest is an ensemble model that divides the training dataset into random subsets and fits a decision tree classifier on each part before aggregating them, which results in less overfitting and better accuracy for prediction [15]. Random forest effectively divides the input variables into multiple disjointed regions and give each one of them a set value for the prediction. The algorithm is a flexible supervised ML solution and used both for regression and classification [57].

5.1.7 Extremely Randomized Trees

Extremely randomized trees is similar to random forest, training trees on random subsets of features and forming a resulting ensemble [58]. However, it differs by drawing random thresholds for each feature and using the best as a splitting rule instead of the most discriminative threshold. One effect of using Extremely Randomized Trees instead of random forest is reduced variance and greater bias [59].

5.1.8 Extreme Gradient Boosting

Extreme Gradient Boosting (XGBoost) is an ensemble method that builds on the decision trees and gradient boosting methodology. XGBoost iteratively learns from previous trees by optimizing the weight for an observation based on results from a previous classification tree. This is also called boosting, which differs from bagging techniques by taking previous trees into consideration [54].

5.1.9 Light Gradient Boosting Machine

Light Gradient Boosting Machine (LightGBM) is a tree based algorithm that is used for fast gradient boosting. To reduce memory usage and computation costs that come with XGBoost, LightGBM uses histograms to gather continuous features into separate bins. LightGBM grows its trees leaf-wise, meaning that one entire branch is developed at a time instead of an entire level for multiple branches. This results in lower losses as it converges quicker [60].

5.2 Evaluation metrics

Looking at anomalies in real data domains, datasets are often imbalanced, meaning there are more occurrences of one class than another. When training classification models on imbalanced data, accuracy cannot be used as an indication of how well the model performs as the misclassification of positive instances is not as meaningful as for misclassification of negative ones. Also, models tend to give a robust prediction for the common class but overfit for the class with a smaller set of examples [15]. To evaluate performance of respective models, a vast amount of scoring models can be used and compared. The scoring output is a way to rank different algorithms to scientifically prove and compare the different algorithms. There are several ways of doing this, and it is usually correlated with what kind of ML approach is being used, the target of the investigation and how imbalanced the dataset is.

5.2.1 Precision, Recall and Accuracy

Precision is a performance measure that only takes the true positive (TP) predictions and false positive (FP) predictions into account. Precision measures the ratio of correctly positive observations out of the total predicted positive observations by $\frac{TP}{TP+FP}$. A high precision thus indicates few false positive ratings [61].

Recall is used to understand how complete the results are. The recall is defined by also including false negatives (FN) in the equation $\frac{TP}{TP+FN}$ where the number of true positives is divided by the predicted results. The recall score is a way of telling the percentage of all actual positive cases that the prediction predicted right [54].

Accuracy tells how close to reality the algorithm performs by computing the ratio of correctly predicted observations, true positive and true negative (TN), of the total number of observations following the equation $\frac{TP+TN}{Total}$. Accuracy can also be computed for imbalanced datasets, called balanced accuracy score, which calculates each class recall value and returns the average of those [61].

Some of the previous mentioned methods can be extended with averaging methods such as macro, micro and weighted. Macro takes the unweighted average from the metrics that are calculated from each class while micro class independently calculates the true positives, false negatives, and false positives. The weighted metric takes the weighted average based on the class distribution. Both precision and recall have methods for macro and micro scoring [62].

5.2.2 F1-score

F1-score is based on a combination of precision and recall where

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} = \frac{TP}{TP + \frac{1}{2}(FP+FN)}.$$

The F1-score is particularly good for binary classification problems. F1-scoring is the weighted average of recall and precision and is preferred to use when precision and recall are equally important, and the datasets have imbalanced class distributions. F1-scoring can also be extended to other scenarios that not only focus on true positive predictions. Macro-averaged F1-score is used for tasks with a single label and multiple classes. It takes the unweighted average of the F1-score for each class, which gives all classes an equal contribution to the result no matter the ratio of each class. Weighted F1-score also calculates the F1-score for each class but takes the weighted average based on the class distribution ratio [63].

5.2.3 Matthews correlation

Matthews's correlation coefficient is a measure that can be applied on very imbalanced datasets to measure the quality of the classification [62]. It is used for binary classification problems and uses both true and false positives and negatives and is defined as $MCC = \frac{TP \cdot TN - FN \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$. Thus, it is able to calculate balanced performance measures as all predictions of classes are counted. Matthews's correlation gives a value between -1 and +1, where 0 is an average random prediction and 1 a perfect prediction [64].

5.2.4 AUC and ROC

The receiver operating characteristic (ROC) curve is a model for classifying how well an algorithm corresponds to reality. ROC is a probability curve. Area under the curve (AUC) represents the degree or measurement of separability. Separability is how good the model is at separating the distinctive features between the classes [15]. In other words, the AUC tells the proportion of correctly classified samples, which for imbalanced datasets can be very misleading [62]. The ROC curve plots the relationship between false positive rate and true positive rate as the threshold of the decision changes. The AUC describes the proportion of samples that are correctly classified. The curved shape indicates the relation between FP and TP as a function of the classification. Thus, a high AUC speaks for a better algorithm than a small AUC [15]. Weighted AUC may be interpreted as the weighted average of sensitivity with weights emphasizing the class distribution [65]. Macro AUC gives equal weight to each classified label and displays the study as a whole [15]. In other words, the difference of macro and weighted AUC is that the macro takes the unweighted average for each class independent of the class distribution while the weighted metric takes the class distribution into account. In case of an imbalanced dataset,

macro average is recommended as it can be more informative when including equal weight of the minority class [62].

5.2.5 Confusion matrix

Confusion matrix is a visualization that shows the mislabeled samples for a classification model where one axis marks the predicted labels, and the other axis marks the actual true label. Thus, a perfect model has all samples along the diagonal of the matrix as it tells that the predicted label also is the true label. The confusion matrix is a good evaluation metric as it quickly gives a visual overview of the mislabeled amount for the minor class, which is a common scenario when predicting on imbalanced datasets [62].

5.3 Dealing with class imbalance

There are many established procedures for how to deal with imbalanced data and typical use cases are implementing over- or under-sampling on the rare or common class label. An additional way is to use ensemble-based classifiers that handle the imbalances itself and do not require any preprocessing of the data [66]. Another method to handle imbalanced datasets for ML is to make sure that they are trained and validated fairly. As different subsets, or folds, of data are used to train, validate and test algorithms, the subsets should have representatives from all classes for a fair validation [67]. One method to prevent sampling bias for binary classification is stratified sampling [68]. Stratified sampling is a method that splits each class and then assigns one part to a fold together with a part from another class. In this way both classes are represented with the same ratio in all folds after the split as the ratio in the total dataset [15].

6. Case study

Epiroc manufactures products for the mining and construction industry and is divided into several sub areas. Epirocs production site in Fagersta is in the tools and attachment division and is Sweden's biggest manufacturer in terms of the amount of pit ovens and the throughput of heated goods that is cooled by air [Respondent 1]. This thesis is conducted in the rock drilling operative area within manufacturing of steel shank adapters, couplings, hex rods and drill bits. The analytical outset is the heat treatment process of long shank adapters or long drill rods that can be coupled to each other. These rods are called Male-Female (MF), visualized in Figure 1. Manufacturing of these products is challenging since several dependent processes are involved. This section describes how heat treatment is conducted at Epirocs site in Fagersta and how process data is collected. Interviewed respondent thoughts on important variables for quality output are also summarized to make a ground on important features that should be included in the research.

Figure 1. Epiroc steel rods, also called MF rods. The male part can be coupled to another rod's female part.

6.1 Heat treatment process on site

The production of hardened steel is a complicated process with a plurality of operational steps. These steps are shown in Figure 2. The process starts with pre-process loading for which the rods are mounted on a plate together with one or several test pieces. The batch is lowered into a pit furnace with the right depth adapted to the length of the product [Respondent 1]. The pit furnaces have a carbon environment which hardens the surface of the steel [Respondent 2]. The batch is then transported to a cooler where the temperature of the rods is lowered by fans. At the post-process unload, the order is unloaded from the cooling tubes and moved to the next operation. At the same time as the order is on post-process unload, a sample is sent to the quality department for a deeper

examination of the material's characteristics. If the quality is approved, the order is released for the next operation [Respondent 3]. The process is exceedingly difficult to perform, and a small error can give a large negative quality outcome with a total batch rejection as a result [Respondent 1].

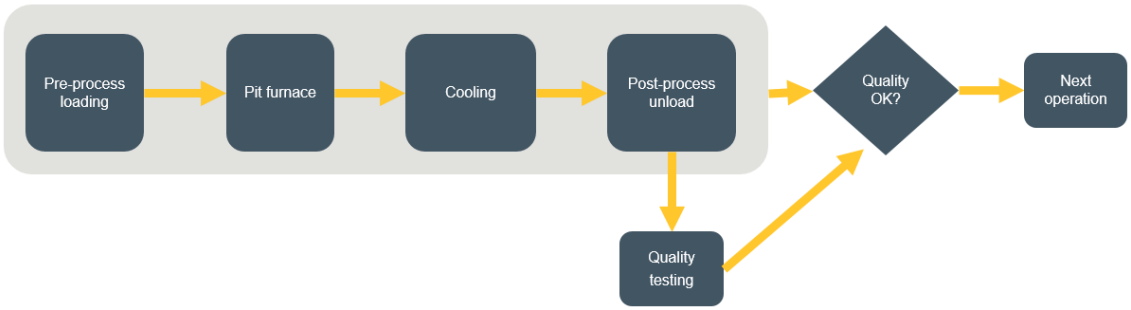


Figure 2. Process map of the heat treatment flow and quality testing which results in a rejected or approved batch.

6.2 Quality process and data collection

Each batch has either one or two test pieces or a product that is quality tested. Each control tests three quality parameters; hardness, case depth and carbon rate. If the measured result for one of the tests is outside the tolerance interval, a reassessment is done on another product from the batch. For borderline cases, the technicians discuss whether the order has the quality that the customer asks for to decide on a full batch rejection or not. The quality tests are always performed the same way for each product. The point of measurement on the test piece depends on the product type. All results are manually documented in a physical form that is archived in files for each respective furnace. The same data is also manually filled in an Excel file where all test data are documented in the same sheet. Each batch has a status column that indicates whether the product is approved or not [Respondent 4].

6.2.1 The flow between heat treatment and quality control

Data is gathered in different systems and databases during the heat treatment process. Figure 3 gives an overview of the data collection from the different production steps. Manufacturing orders follow a certain routing of operations. The route is logged into an Enterprise Resource Planning (ERP) system that was implemented in October 2019. The heat treatment operation process is adapted with recipes for different products to gain the right characteristics of the steel. As the order is picked up by the heat treatment division, the appropriate recipe starts and data variables from the pit furnace and quenching process starts to log. Process data is stored in a control system in the format of a Manufacturing Execution Systems (MES) data lake in the heat treatment system [Respondent 3].

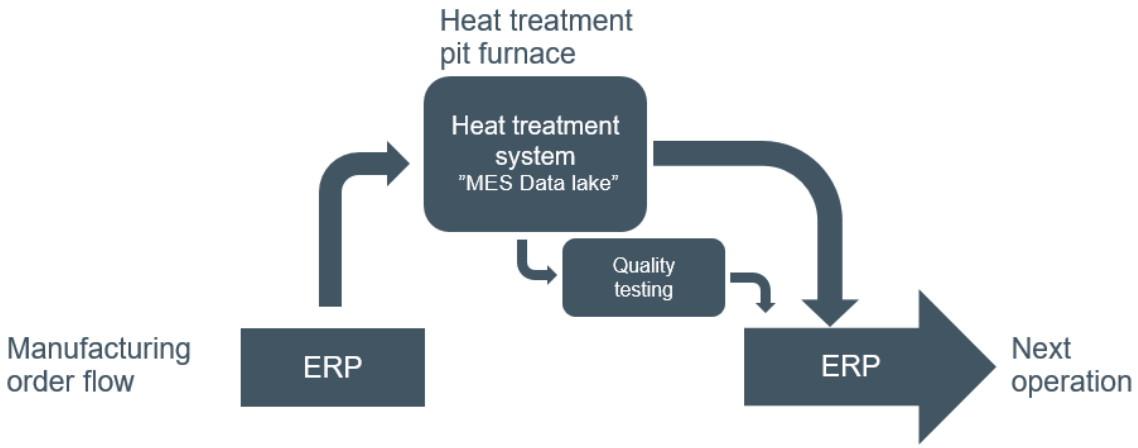


Figure 3. The manufacturing order flow for heat treatment in pit furnaces.

6.2.2 Process data

Each heat treatment process follows a recipe that is predefined for the product. The desired values are plotted on a monitor during the process together with data for how the operation performs. Variables for the heat treatment process are logged real-time. [Respondent 5] Two automatic collections of data are performed during the furnace and cooling process. These are in form of time series data that measures the temperature every 52 seconds [Respondent 1].

6.3 Domain knowledge of efficiency gaps

If the heat treatment process for one product batch fails, it means that all previous processes for that batch have been in vain unless it can be reheated. Production failures in the heat treatment process are thus very expensive given time, material and resources as it affects the whole production chain [Respondent 6]. The connection between logged process data and the quality outcome is not fully documented and correlations between certain properties and hardening are not established [Respondent 3].

The pit furnaces have a constant temperature of almost 1000 degrees Celsius which is very expensive in terms of costs and environmental impact. The material needs to be streamlined with a low rejection rate in order to use the pit furnace to its full potential and minimize its impact on costs and environment [Respondent 2].

There is a mutual understanding about the weak links in the manufacturing lines among the know-how people in production and quality testing [Respondent 1, Respondent 5, Respondent 7]. Firstly, the heat treatment process is broadly automatic, which minimizes the risk of rejections due to human error. The recipes for the pit furnaces are fixed and preselected, which makes manual involvement less frequent than 1 out of 1000 operations. Secondly, compared to the pit furnaces, the cooling process is much less stable. The fans run on less generalized programs compared to the pit furnaces and take more consideration to weight, type of product and number of articles. The fan's

environment is also more insecure as the surrounding air in production fluctuates due to season changes.

Another established fact is that both the furnaces and the cooling fans are individuals and behave differently for unknown reasons. Different fans have different quality output [Respondent 1]. A pre-study has been conducted by Epiroc to consolidate the idea of bottlenecks in the heat treatment process. The study resulted in an analysis of faulty products caused by specific process steps together with a recommendation of what areas in the heat treatment process to investigate further. A specific fan is overrepresented in causing rejections as well as a specific drill rod product which are rejected more frequently than other products [Respondent 3].

7. Tools and architecture

The project is using Microsoft Azure resources to test the applicability of implementing an AI pipeline to store, manage and perform ML on data from the heat treatment process. The pipeline in Figure 4 is optimized by Microsoft for Epiroc to enable data engineering pipelines and take ML to the production environment.

Raw data is written to a binary large object (blob) storage which is used to store big amounts of unstructured data in an Azure cloud environment [69]. The unstructured data is used as input to a bronze, silver and gold architecture powered by Databricks which is further described in Section 7.1. The reason for using a bronze, silver and gold architecture is to save different states of data to be reused for other projects, reducing duplication of work and computation costs. Bronze silver and gold are destination sources for different levels of refined data. Bronze contains raw data in a uniform format, for example Parquet files. Silver is filtered, cleaned and augmented data. Gold contains business level aggregated data, ready to be used in visualizations and reports [70].

The process that is conducted in between the destinations bronze, silver and gold is the extract, transform and load (ETL) work and it is always conducted in the same order of steps. First, data is extracted from a source, then transformed, and lastly loaded into a destination source [71]. The gold state of data is used to find a suitable ML model for the problem. For this project Azure Machine Learning, which is further described in Section 7.2, is used for the modeling and the results are visualized in Power BI, described in Section 7.3.

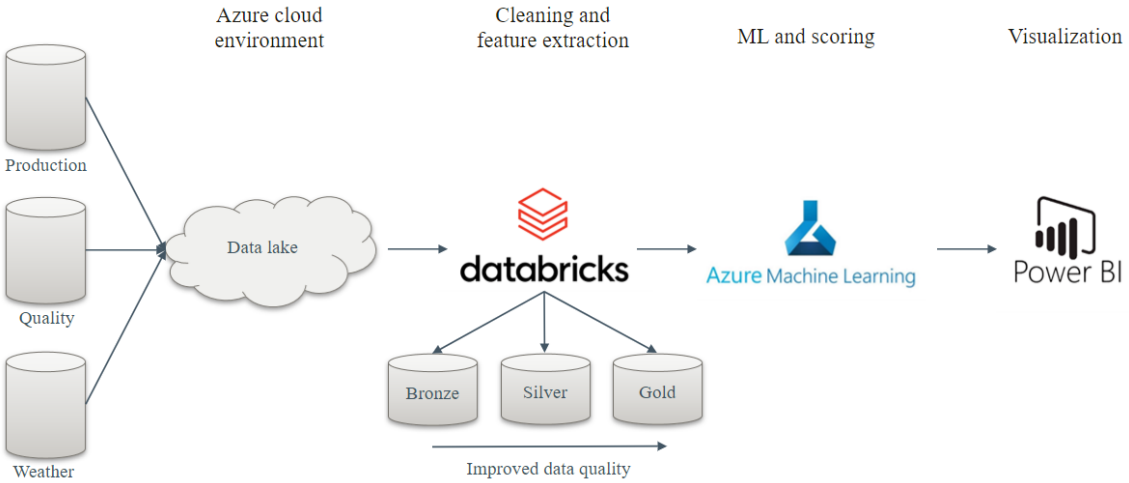


Figure 4. AI pipeline that takes raw process data to the cloud, cleans it, extracts features, performs ML and visualizes the results for people to gain insights of the process.

7.1 Databricks

Databricks is a cloud platform providing workspaces for notebooks, datasets, storage and compute clusters. Apache Spark is pre-imported in Databricks, and is a parallel processing framework for big data and scalable cluster computing [69]. It supports multiple programming languages and offers libraries to handle SQL, dataframes and ML. PySpark is a Python API based on the Apache Spark framework and enables the user to work across different programming models such as SparkSQL and Pandas [72]. In this project, Databricks platform is used running an Apache Spark cluster with 3 worker nodes. PySpark and Python are chosen as programming languages for all types of transformation of the data as it supports SQL-like commands, dataframes and relevant libraries for data modeling [73].

7.2 Azure Machine Learning

Azure ML is a tool to collaborate on notebooks, share compute resources and trace changes while accelerating, automating and deploying ML models. The automated ML service is called AutoML and makes it possible to featurize and train algorithms on data using Python Software Development Kit (SDK) [74]. AutoML trains and tunes models for classification, regression and forecasting through Python SDK which provides open source code and functions easily accessible for the user to train models.

When training, AutoML trains models given features and one or several ML algorithms in parallel through a set of pipelines [75]. The user specifies parameters such as number of iterations, algorithms, dataset, source format, computational target and type of problem to be solved. Each iteration results in a trained model, a training score and a ranking based on the score seen in the leaderboard to the right in Figure 5. AutoML also enables automatic featurization and hyperparameter tuning. The featurization is scaling or normalizing the data with methods found in Scikit-learn libraries. Automatic hyperparameter tuning means that AutoML automates the process of configuring the optimal set of hyperparameters that gives the best performance [76]. Thus, each iteration has its own unique set of parameters which are automatically tuned for each iteration based on previous ones. The type of hyperparameters depends on the algorithm trained.

The algorithms are trained on the training sets of the data and evaluated by using cross-validation that takes a part of the training data for each iteration as validation of the algorithm. The variation of features, algorithm and parameters are resulting in a training score which can be presented in a range of different metrics, discussed in Section 5.2. The test set is used when the best-performing model has been tuned to conduct a final evaluation of the model on unseen data [77].

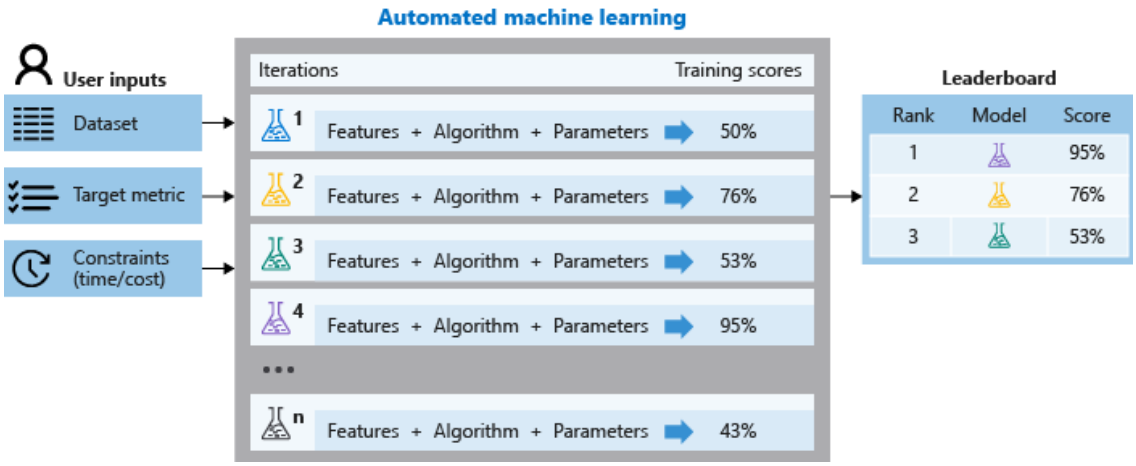


Figure 5. The automated ML flow when using AzureML. Each iteration results in a training score given the user input and the best models get ranked in a leaderboard.[1]

To make calculations more robust, a Data Science Virtual Linux Machine is used in this project, which retrieves the finished gold data set and coordinates the calculations against a calculation cluster in Microsoft Azure Machine Learning studio. The cluster type is a Standard_D14 with 16 cores, 112 GB RAM and 800 GB disk that is optimized for memory usage. The amount of compute nodes varies between one and four depending on what algorithm is being used.

7.3 Power BI

Power BI is a set of tools used for business intelligence operations. One of the tools is used to visualize results for analysis and discovery from data achieved in models. Visualizations can be used for both batch and real-time analysis and shared with all stakeholders involved. When a model has been deployed, for example through AzureML, Power BI can automatically update the data in the model and display the latest findings [78].

[1] Figure from https://docs.microsoft.com/en-us/azure/machine-learning/concept-automated-ml

8. Experimental Procedure

Data quality is crucial when analyzing data. The data process described in this section tells where the data comes from and why it is chosen. Section 8.1 and 8.2 describes how and where data is retrieved. Section 8.3 describes how the tables are joined and how the data is processed to features that will be used for the ML. The datasets used for training and testing the model are described in Section 8.2 and 8.3 and the methodology on how to select and validate the ML model is described in Section 8.5.

8.1 Data sources

Production data is mainly gathered from a production MES. The production MES is a system that is developed for controlling heating and cooling processes and is crucial for the operation stages to work. Data is gathered under the processes and is being stored in a HeidiSQL database. All process data that is used in the project is fetched from HeidiSQL except from the recipe data that is fetched directly from the MES, recipe by recipe, and manually translated into separate CSV files. Temperature data is pulled from a site with local measurements [79] and quality data is obtained from an Excel file directly from the quality test lab.

8.2 Data selection

To get a dataset that is valid and reliable for the ML algorithm, some delimitations and selections in data are required. Time series data is gathered from the cooling stage as it is a known difficult process to control and causes many of the rejections in comparison to furnaces. The temperature data is also represented by time series. Only data from MF rods are included, as they are the hardest products to cool, which contributes to a less imbalanced dataset with a higher rate of rejected products. In October 2019, a new business system was implemented that forces the research study to add a historical time limitation. To only investigate a stationary process, data points after that event are selected.

Rehardened products and those products that have not passed the pit furnaces are excluded to get as similar production lines for all batches as possible. Recipes are every now and then updated. The MES only shows the last version of each recipe, which means that time series from the time before the recipe was updated latest, are not relevant for the study and would result in misleading values from the feature extraction. Thus all batches that are run before the latest update for the corresponding recipe, are excluded from the study.

8.3 Data preprocessing

In order to get the right data at the ML stage, the data is pre-processed. This step is conducted in close collaboration with domain expertise that provides recommendations on what kind of data is important and can create value with a higher reliability. Obvious errors are handled on the authors own initiative.

8.3.1 Raw ingestion

MES data from October 2019 to October 2021 is extracted in a CSV file from HeidiSQL and the Excel file with quality data is converted into CSV format. As the project is considered to be in a research and development phase and only a few years of stationary data is going to be analyzed, raw data files from the MES, quality data and external files are manually loaded to a blob container in an Azure Storage account instead of being extracted from a Data Factory [Respondent 8]. First, a raw ingestion is performed by extracting data from the blob container. All column names are lower-cased, stripped to not contain any Swedish characters and special signs are dropped. The data frames are then flattened out into a column oriented data storage format called Parquet and loaded to bronze state.

8.3.2 Joining data

The data lake consists of a collection of different data sources. These come from external variables such as weather data with associated temperature as well as more detailed descriptions of different sub-production steps and processes. The tables in the entity relation diagram in Figure 6 are describing, among other things, different types of production data with regards to purpose and detail. The A_value table describes the measured temperature one meter below the top of the bars with the associated unit of time in seconds. A_value also links this value to a specific fan. A_id describes which operation is performed between which times and B_wp describes the type of operation being performed. A_cont_pos is used to link quality data and operation data as well as the number of products in a batch and its unit weight.

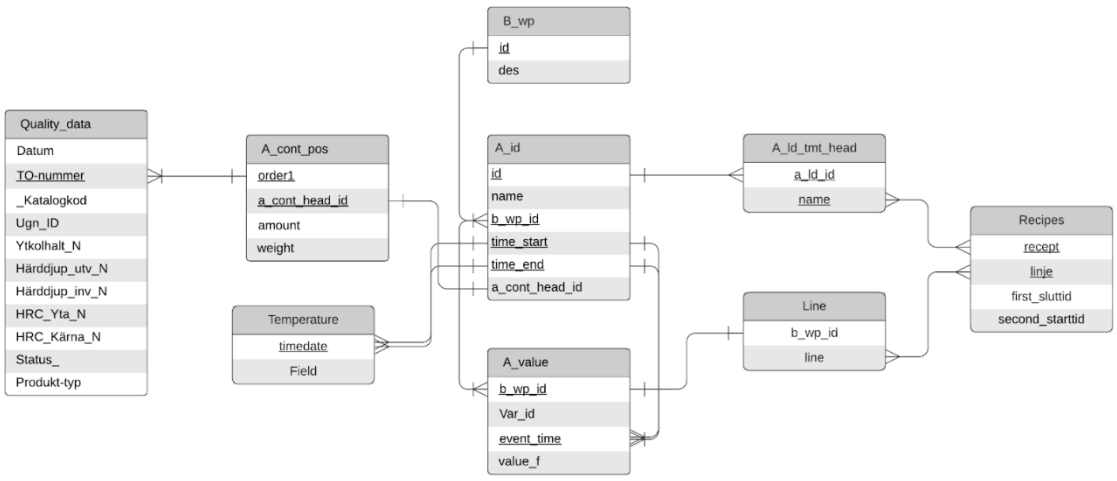


Figure 6. Entity relationship diagram of the datasets used.

The quality data contains various measured variables linked to the properties of the steel. It has also a status field telling whether it was approved or not. The table Temperature contains temperature data in degrees Celsius in the near area of the production site and the temperature is keyed and divided by A_id to get the right mean temperature for each batch. The recipe used for the product differs according to what production line it passes and is determined by the table Line. The recipe names are stored in the table A_ld_id_head and are keyed with A_id and its id. More detailed recipe steps are stored in the table Recipes. The tables A_id and A_value are keyed on multiple columns, using both time_start and time_end to key all event_time for each operation. The same type of scenario occurs when joining tables A_ld_tmt_head to Recipe and Line, as both columns for b_wp_id and name must respectively match each recipe.

8.3.3 Data cleaning

Cleaning of the production data is performed on each of the tables separately before it is merged with the quality data. First, all irrelevant columns for the analysis or keying of the tables are dropped. All files go through several cleaning procedures to make the data easy to handle. For example, all time columns have a Unix timestamp, which needs to be converted into DateTime format. The original data sets are filtered and the columns that are kept are shown in the entity relationship diagram in Figure 6. The most common operations in the cleaning process on row-level are summarized in Figure 7. Also external data, such as weather data, are cleaned by for example backfilling NaN temperatures. Recipes are not filtered or cleaned before the joining stage.

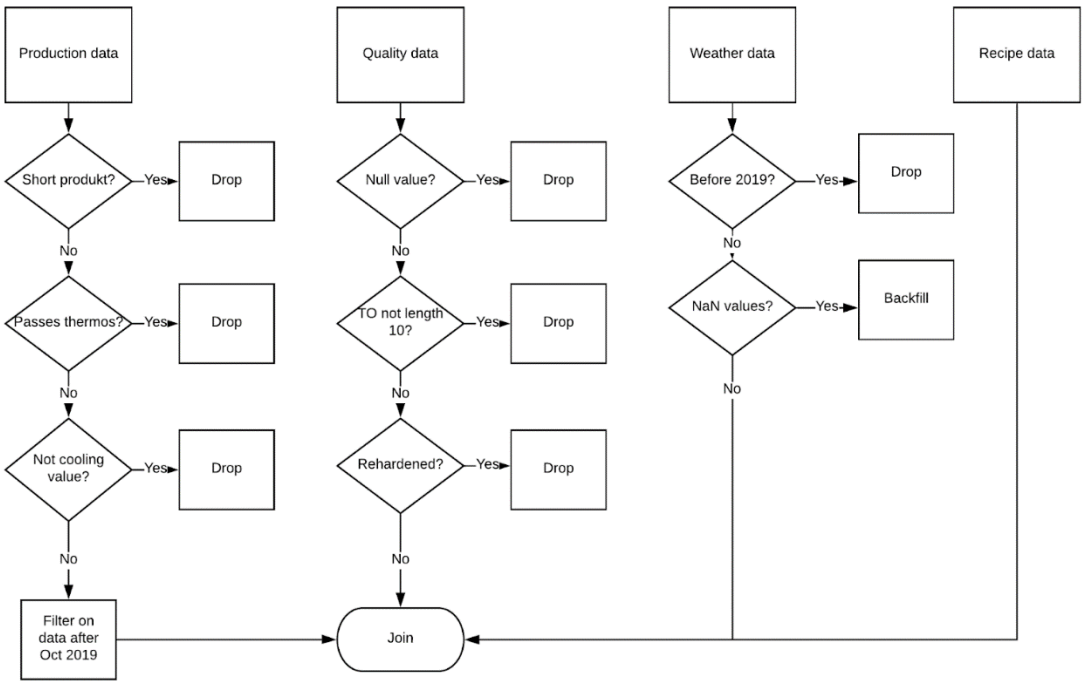


Figure 7. Flowchart of some of the data preprocessing steps for the different sources of data.

8.3.4 Feature Extraction

Figure 8 gives an example of a typical cooling curve. The x axis represents the time and the y axis represents the air temperature one meter underneath the batch. Point 1 marks where the recipe starts as the temperature has passed 65 degrees. That is also why the temperature starts to decrease as the recipe starts with fans cooling down the product. Point 2 marks where the fans let the material rest, which is indicated by the rising temperature. Point 3 marks the maximum temperature, from where the most important part of the cooling process begins. Between point 3 and 4, most of the characteristics of the steel is set. At point 4, the fans are in full effect again to cool down the material as quickly as possible to make room for new batches and transport the current batch to the next operation. The curves vary depending on the recipe.

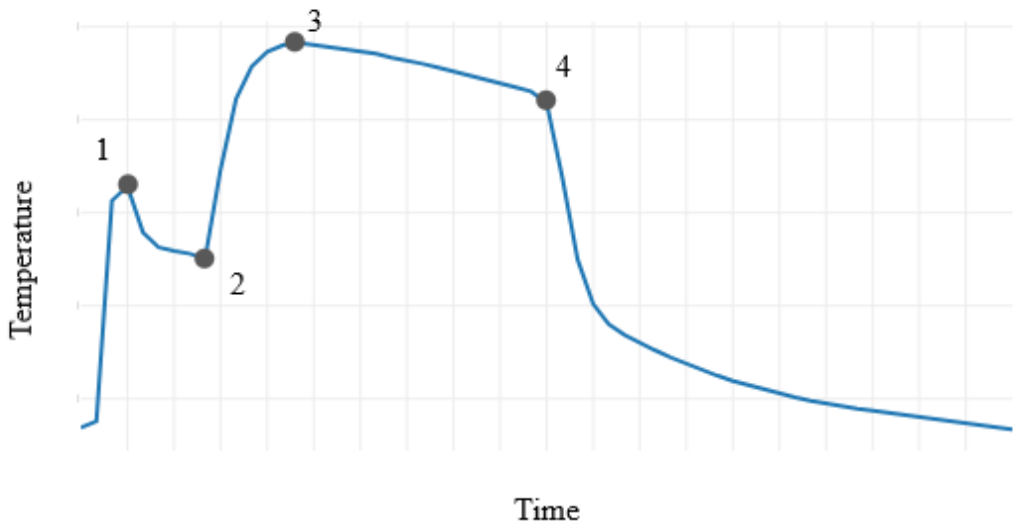


Figure 8. Example of a time series from the cooling process with marked points of interest.

Variables calculated for each time series are described in Table 1. All features have been summoned and extracted according to local recommendations.

Table 1. Features extracted from each time series.

Feature	Explanation	Data type
Max	The maximum temperature measurement of the entire process, at Point 3.	Float
Knee	The temperature of the knee, located at Point 4, is calculated by looping over a window which filters on temperature, group by TO number and order by event time based on the recipe time for Point 4.	Float
Difference	The time difference in seconds from the max value Point 3 to Point 4.	Long
Integral	The integral of the curve is calculated using the composite trapezoidal rule from the SciPy library. [80] Trapezoidal rule integrates over each time event and upcoming temperature value. The result gives the area under the curve. Two integrals with different time intervals are calculated. The first one is based on the recipe from Point 3 to Point 4, from the maximum point to the knee. The other one is calculated on all temperature measurements above 100 degrees. The second integral is calculated as an extra security measure, as recipes and real output are not always consistent.	Float
Standard deviation	The standard temperature deviation from start of recipe until event time for Point 4 is calculated with Pandas .std() function.	Float
Hypotenuse	The length of the hypotenuse between the temperature measurement for Point 3 and Point 4 is calculated using Pyspark .hypot() function.	Float
Slope	The slope from the maximum until the transformation of the steel has ended. Two slopes are calculated. One using the Point 3 and Point 4 values together with their time difference. The other one is using Point 3 and a temperature measurement 15 minutes after the max point in time. The second slope is also calculated as an extra security measure, as recipes and real output are not always consistent.	Float

There are some features extracted from other variables that are believed to have an impact on the product quality which are described in Table 2.

Table 2. Features calculated from the raw data.

Feature	Explanation	Data type
Total weight	The total batch weight is calculated by multiplying the amount of goods with the individual product weight.	Float
Length	The final length of the products is given in the product name.	Double
Mean temperature	The mean temperature of the outdoor weather during the operation. The outdoor temperature is generated as a time series by a sensor placed in Fagersta.	Float

The remaining variables, described in Table 3, are already given in the dataset.

Table 3. Features accessible in the raw data.

Feature	Explanation	Data type
Amount	The amount of goods in the batch.	Integer
Weight	The weight of each rod in the batch.	Float
Furnace id	The name of the furnace that the batch has passed.	String
Recipe name	Name of the cooling recipe.	String
Fan	Name of the fan that the batch has passed.	String

The Status of the product, approved or rejected, is the label that will be predicted. The Status label is received from the quality output file and is described in Table 4.

Table 4. Label from the quality results that tells the status output of the batch.

Label	Explanation	Data type
Status	The status, or result, from the quality testing that is either rejected or approved. 0 is rejected and 1 is approved.	$N \in [0, 1]$

8.3.5 Silver to gold data

The process of joining silver tables and extracting the features needed for the gold table are summarized in the flow in Appendix B. As the rows of all summarized data are increasing fast with more unique batch numbers (TO_numbers), the flow is divided into three paths to minimize computational costs. After joining workplace data (B_wp) to the dataset, the data follows two different paths. In one path, the data is joined with time series data, from which time series features are to be extracted. The other path is joined with temperature to extract features that are not connected to the time series data. After all features are extracted, they are joined with the initial dataset that was used as input to the two paths. Together, it forms the gold dataset that contains one row per TO_number and one column for each feature and label. The gold data that is created when joining the silver datasets consist of seventeen different columns containing extracted features, process information and the status label that tells whether the batch is rejected or not. The total number of rows is 410, making the dataset quite small in terms of ML analysis.

In this project, the specific use case and the degree of rejection makes it possible to balance the data. The imbalanced data is easily adjusted by further tightening the tolerance level which generates more non-approved batches. As a lot of the results from quality tests lie on the very margin of approval, no drastic changes need to be done to balance the data. However, tightening the tolerance level will also mean that there will be no or very small distinction between rejected and approved products which can be hard for a model to interpret given the small amount of total data and uncertainty in correlation to extracted features. As a compromise to both of these issues, the dataset will have a somewhat tightened tolerance level, where the most optimal hardness measures will pass, like visualized in Figure 9. In addition, ensemble algorithms are included in the set of algorithms that will be trained and evaluation metrics that are adopted for imbalanced data are used to evaluate the best algorithms.

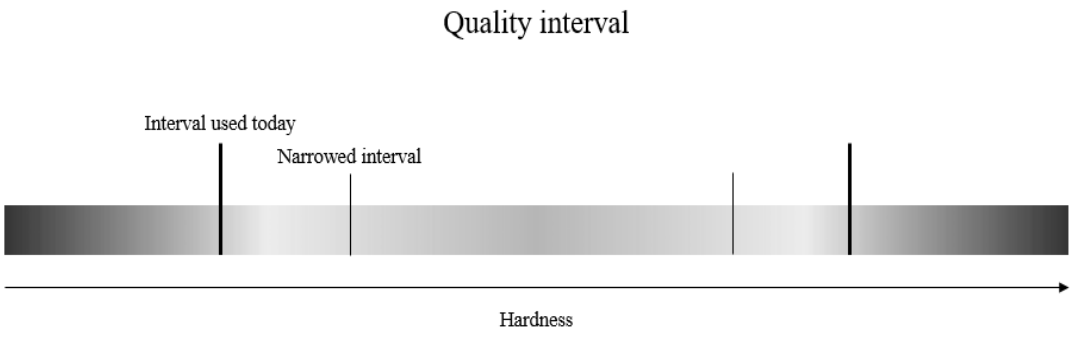


Figure 9. The tolerance level used is narrowed to only pass the best batches.

8.4 Data limitations

The data and heat treatment process has limitations that are important to be aware of for the limitations of the study. As there are no logs on past recipes, only the most recent recipe and the batches that have passed that program version can be included in the research. Unfortunately, this means that the majority of all batch data must be excluded from the analysis, and that only a few programs from 2020 are included. This can cause imbalance also in the labels and there are no possibilities to look closer at a specific product catalog or recipe without getting a too small dataset. The limitation in the amount of data might result in more biased data and less well-grounded results.

Other errors in the data, such as products that get another product's cooling program, may cause poorer performance in the ML models. These errors are hard to make a generalized solution for, thus only a part of them are found and marked. The obvious errors result in null values and are in this research replaced with a 0 and the recipe name is renamed to "Wrong recipe". Given that the dataset already is limited in size, this can have a severe impact on how well the ML model performs and interprets the other features for those rows.

Balancing the dataset also means that the focus of the investigation shifts from finding the extreme anomalies, to finding those who have less perfect results. The extreme anomalies do not get the same weight in the more balanced dataset which might lead to greater difficulties finding the significant connection between input and quality.

The authors, accompanied by local engineers with extensive domain knowledge, suspect the material itself being an important cause of different quality results. The materials in the steel products are different and originate from two different main suppliers. The thesis is not aiming to analyze the differences of the steel manufacturers but rather focusing on Epirocs internal processes; thus the steel itself is handled as one single constant and the differences are treated as such.

8.5 Model selection

The different ML algorithms described in Section 5.1 are compared to get good candidates for a future model promotion. When comparing the nine algorithms, the best performing model determines which type of algorithm is promoted to be optimized by feature selection and lastly parameter tuning. The tuning is increasing the algorithms correctness and accuracy to get a better classification result. This section defines how the models will be trained, selected and tuned.

8.5.1 Train and test datasets

Stratified Shuffle Split is an object provided by scikit-learn that is used to stratify sampling in this project. The Stratified Shuffle Split preserves the percentage of samples for each class as it splits datasets into stratified and randomized folds [81]. Stratified

Shuffle Split will be used with 80 percent of the data for training, 10 percent for validation and 10 percent for testing. Validation will be performed with cross-validation due to the restricted amount of samples. Thus the validation set will not be used in this research investigation, but for upcoming experiments when more data is available. The gold dataset consists of 410 rows, out of which 39.8 percent are rejected. This results in the same proportion for the train, validation and test set, specified in Table 5.

Table 5. Distribution of class labels across gold, train, validation and test set.

	Gold	Train	Validate	Test
Total	410	328	41	41
Rejected	163	130	17	16
Approved	247	198	24	25

8.5.2 Evaluate models

Accuracy is an inappropriate scoring method to use for imbalanced data as it will give a high accuracy score given the true positives of the majority class. Thus, it is not a preferred measure for this case study [54], [82]. Even though different measures are good for different purposes, depending on what is important to predict from the data, averaging methods are generally most suitable to use for class imbalance as the minority class gets the same weight as the majority class [62]. Matthews correlation coefficient is said to be able to give an equitable validation even if the dataset has some severe imbalance although studies have proved the opposite, concluding that it is just as accurate as F1-scoring [83].

As this thesis aims for a balance in classifying true positives and true negatives, evaluation metrics that take class imbalance into account are used to verify the most appropriate algorithm [15]. There are multiple metrics that are valid candidates for the study, and there is no measure that is given to be the best one, as it depends on the studys character. The goal for this study is to foremost choose an algorithm that can separate the class labels as well as possible. No matter the distribution, each class should have an equal weight. This motivates metrics adapted for imbalanced datasets and using such metrics given the class imbalance presented in Table 5. Both Debón et al. [84] and El-Bannas [85] has researched on AUC as a measure for binary classification in metal manufacturing. Both investigations proved AUC to be useful in comparing models for the industry. This is also confirmed in a research by Starovoitov et al. [86] that proves AUC being the best estimation function on both balanced and imbalanced datasets.

As AUC has been used and proved to be at least equally accurate as other measures in previously presented research, its macro metric is used as the primary metric when selecting an algorithm and parameter tuning it. To support the AUC, five other metrics

for imbalanced datasets will also be considered to both validate and back up the macro AUC score. Along with the macro AUC is Matthews correlation coefficient macro F1, weighted AUC, macro recall and balanced accuracy score investigated. The ROC curve and confusion matrix are also plotted to better visualize how well the model separates the classes.

An overview of the process in choosing the best algorithm is seen in Figure 10. First, nine algorithms are trained, out of which the algorithm for the model that performs the best considering its performance metrics will be chosen for feature selection. Only one algorithm is considered for the feature selection. The feature selection is based on the ranked feature importance of the best performing model in the first round. This tells how each feature and label influences the model prediction. The algorithm is trained with different subsets of the most important features and performance metrics are compared to find the optimal set of features. The different subsets of features that are chosen for each run are presented in Appendix C. As there are two similar measures for integral and slope, only the most important feature of each will be included in some of the runs. Lastly, the algorithm chosen in the first round and the optimal subset of features are going to be hyper parameter tuned with a longer threshold time to get a final model to deploy to Power BI.

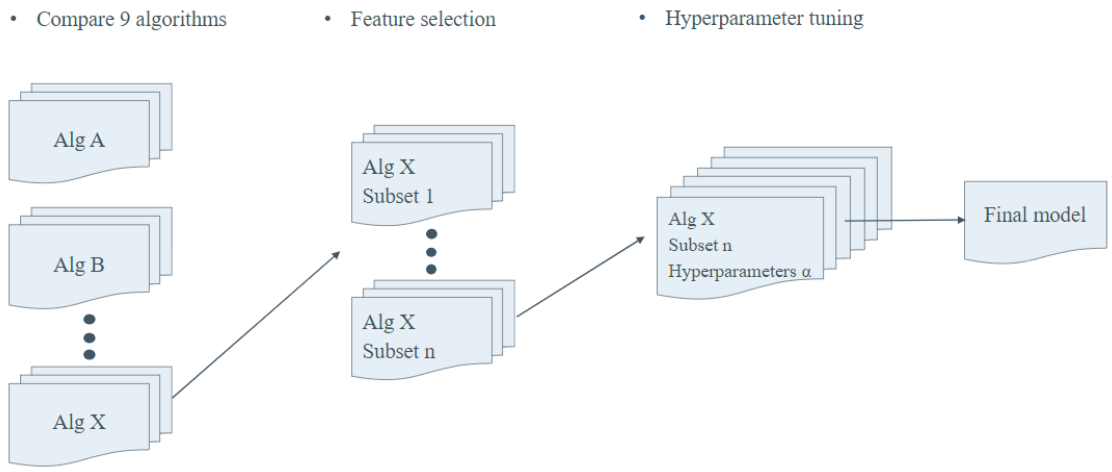


Figure 10. Overview of methodology used when choosing an algorithm, selecting features and tuning a final model.

Some of the parameters for the model are set before each algorithm is trained. For the first and second round, in algorithm and feature selection, the threshold is set to one hour. Each model is trained with 300 iterations and validated with five fold cross-validation. Early stopping is also enabled, which terminates the training if the latest twenty iterations have not converged. For the last round the algorithm is trained with 300 iterations and a threshold of one or five hours.

9. Result

This section presents the results for the models trained based on the algorithms presented in Section 5.1 and the metrics described in Section 5.2. First, all the compared algorithms are presented in Table 6. More details on the feature selection can be found in Appendix C. This is followed by the result from feature selection for the best performing algorithm. The section ends with parameter tuning where the number of folds, run time and iterations are adjusted. The best resulting model is visualized in Power BI, seen in Figure 28. The featurization and hyperparameters used in the final model are presented in Appendix D.

9.1 Best models for each algorithm

The resulting metrics from the best trained model of each type of ML algorithm are shown in Table 6. Random Forest reaches the best macro AUC score, which is the primary metric. Each algorithm has several iterations and only the iteration with the highest macro AUC score is presented in the table.

Table 6. Metrics for the models with highest macro AUC score for each algorithm

Algorithm	Matthews correlation coefficient	F1 macro	AUC weighted	AUC macro	Recall macro	Balanced accuracy
Light GBM	0.291	0.637	0.701	0.701	0.64	0.64
Random forest	0.284	0.626	**0.713**	**0.713**	0.63	0.63
XGBoost Classifier	0.276	0.631	0.688	0.688	0.632	0.632
Stochastic Gradient Descent	0.093	0.472	0.607	0.607	0.539	0.539
Linear Support Vector Machine	0.109	0.437	0.604	0.604	0.527	0.527
Gradient boosting	**0.3**	**0.646**	0.675	0.675	**0.646**	**0.646**
Decision Tree	0.248	0.62	0.642	0.642	0.624	0.624
Extremely Randomized Trees	0.291	0.643	0.704	0.704	**0.646**	**0.646**
Bernoulli Naïve Bayes	0.175	0.573	0.566	0.566	0.579	0.579

9.2 Best performing algorithm

The feature importance for the random forest model with highest macro AUC score is presented in Figure 11. The features are sorted by their model importance for the random forest algorithm. Total weight is the most important feature followed by the average outside temperature during the operation, the integral above 100 degrees Celsius and the fan that the batch has passed.

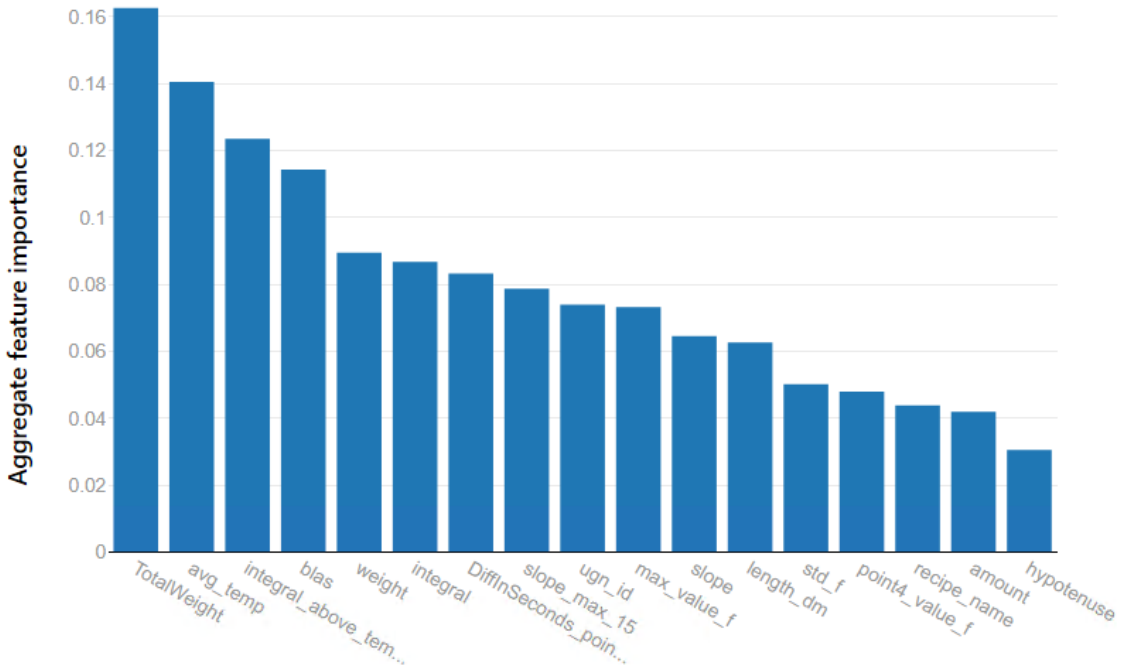


Figure 11. The seventeen features sorted by their model importance for the random forest model with highest macro AUC score.

The ROC curve for the random forest model is presented in Figure 12. True positive rate is on the y-axis and false positive rate on the x-axis. The dark blue line shows the macro average and the light blue the weighted average. The spotted dark purple line is the ideal line. If the average follows the diagonal light purple line, the classifier is as good as a random guess. The average weighted and macro have a similar line and have an AUC greater than 0.5, with other words better than the randomized diagonal line.

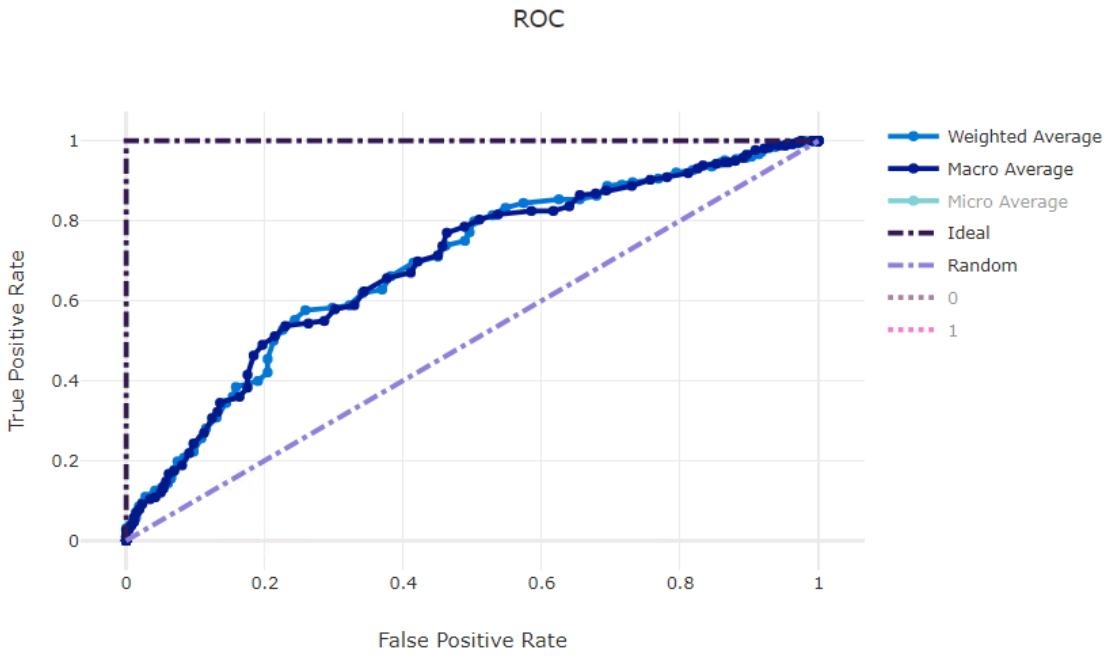


Figure 12. ROC curve for the random forest model with highest macro AUC score after first iteration.

The confusion matrix for the first iteration of random forest with the highest macro AUC score is presented in Figure 13. False negative is 73, false positive is 36, True negative is 57 and true positive is 162. The confusion matrix is based on cross-validation on the training dataset and tells that the model is better at classifying approved batches, 81 percent, than it is at predicting rejected batches, 43 percent.

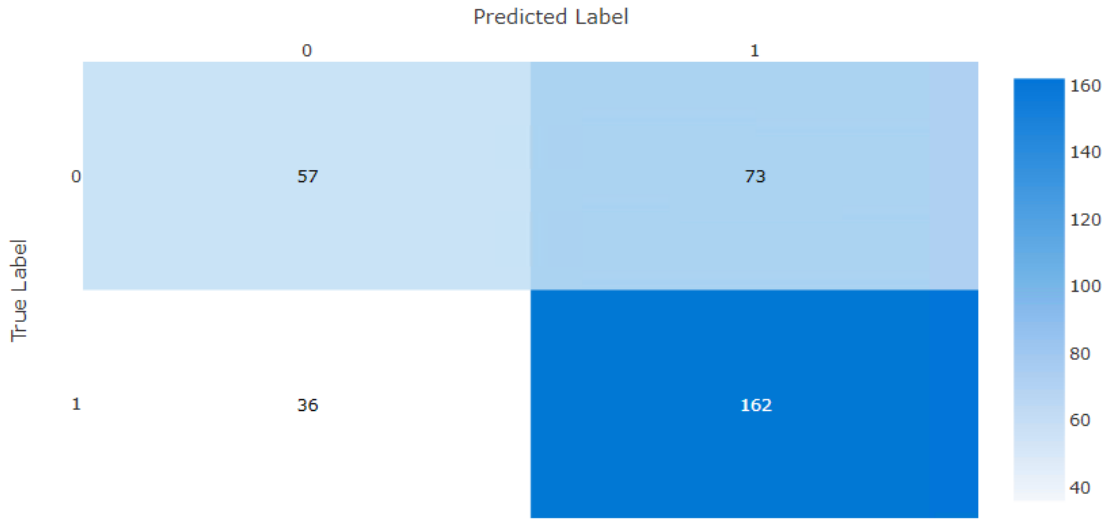


Figure 13. Confusion matrix of the random forest model performance in classifying batches within the given interval.

9.3 Feature selection

Table 7 presents the scoring of random forest with different sets of features which are prioritized based on the feature importance given in Figure 11. Appendix C gives a specification on feature subsets included in each run. Feature subset 1 in Table 7 is the same run as for the first iteration, where random forest got the highest macro AUC score. The scoring for the different feature subsets shows that random forest keeps having the highest macro AUC and weighted AUC. However, feature subset 10 outperform feature subset 1 in all other metrics at the same time as it gets the second best macro AUC.

Table 7. The metrics for Random forest with different subsets of features.

Feature subset	Matthews correlation coefficient	F1 macro	AUC weighted	AUC macro	Recall macro	Balanced accuracy
1	0.284	0.626	**0.713**	**0.713**	0.63	0.63
2	0.287	0.641	0.703	0.703	0.642	0.642
3	0.289	0.643	0.696	0.696	0.644	0.644
4	0.224	0.61	0.648	0.648	0.614	0.614
5	0.227	0.612	0.674	0.674	0.614	0.614
6	0.227	0.593	0.674	0.674	0.601	0.601
7	0.225	0.611	0.675	0.675	0.611	0.634
8	0.301	0.638	0.682	0.682	0.39	0.639
9	0.324	0.66	0.696	0.696	0.661	0.661
10	0.327	**0.662**	0.705	0.705	**0.663**	**0.663**
11	**0.334**	**0.662**	0.704	0.704	**0.663**	**0.663**

9.4 Best model with feature subset 10

The feature subset 10, in Table 7 is trained with the features displayed in Figure 14. The most important feature for the model is the integral of the time series calculated for all values greater than 100 degrees. Second most important is the fan that the batch passes followed by weight, average temperature outdoor, integral, total weight and lastly seconds from Point 3 to Point 4.

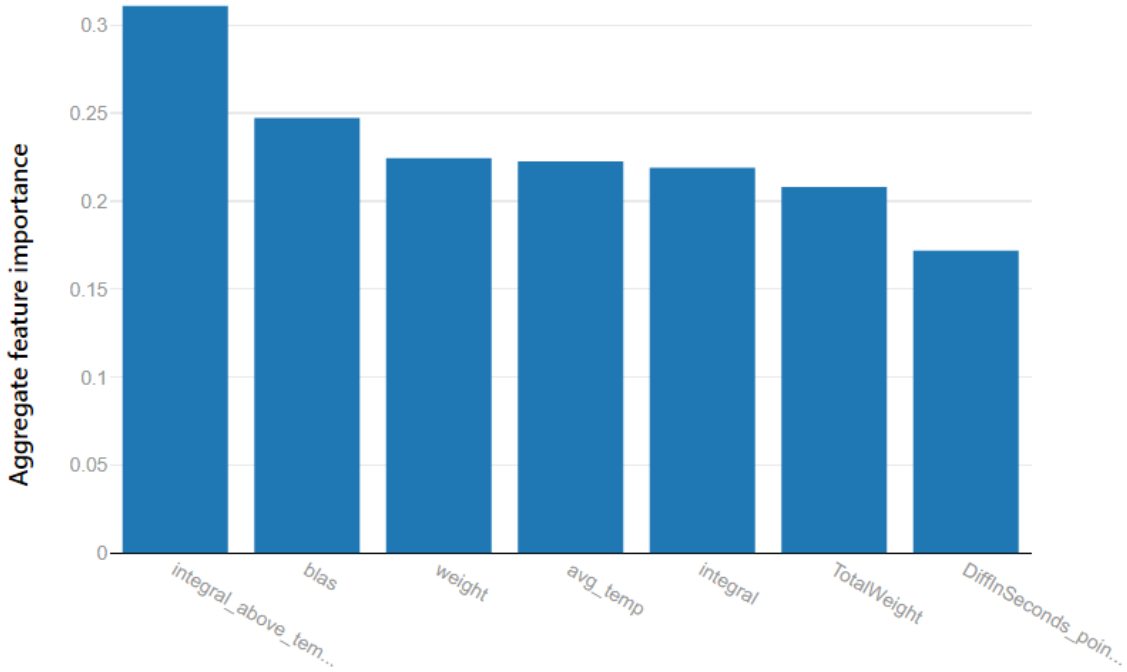


Figure 14. Feature importance for the model with best performing feature subset 10 for random forest algorithm.

The ROC curve for the second iteration of random forest with the best feature subset is presented in Figure 15. The macro and weighted average follows a similar line in between the ideal and random line.

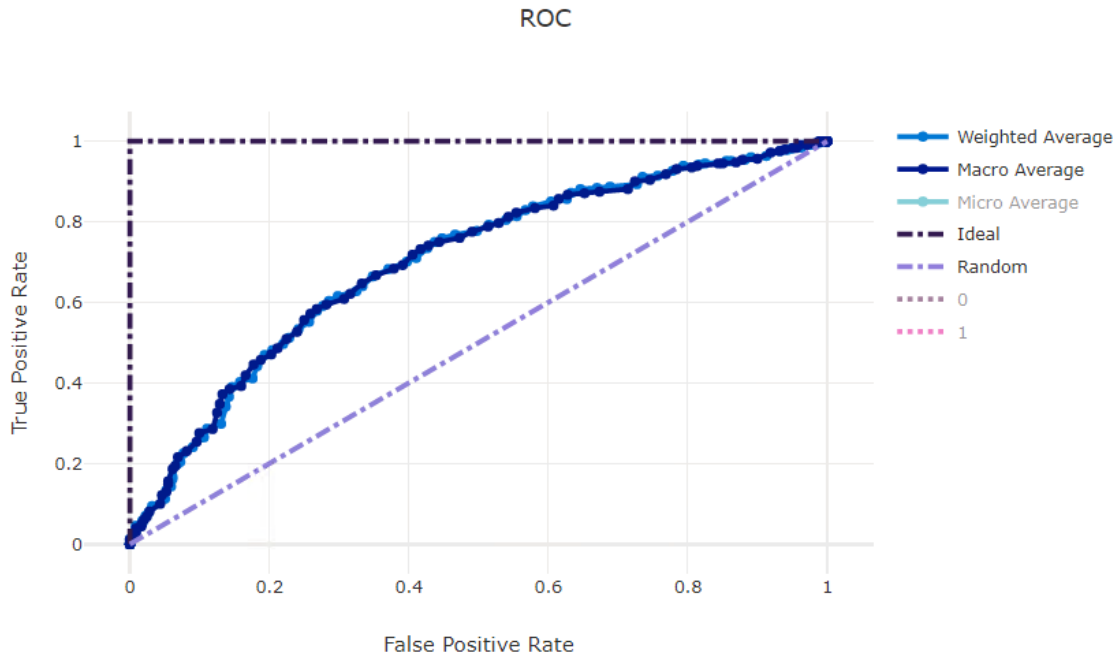


Figure 15. The ROC curve for feature subset 10 showing that the average performs better than random.

The confusion matrix for the second iteration for random forest with the best feature subset is presented in Figure 16. False negative is 54, false positive is 52, True negative is 76 and true positive is 146. The confusion matrix tells that the model is getting better at classifying rejected batches and that it to a majority classifies the approved and rejected batches right.

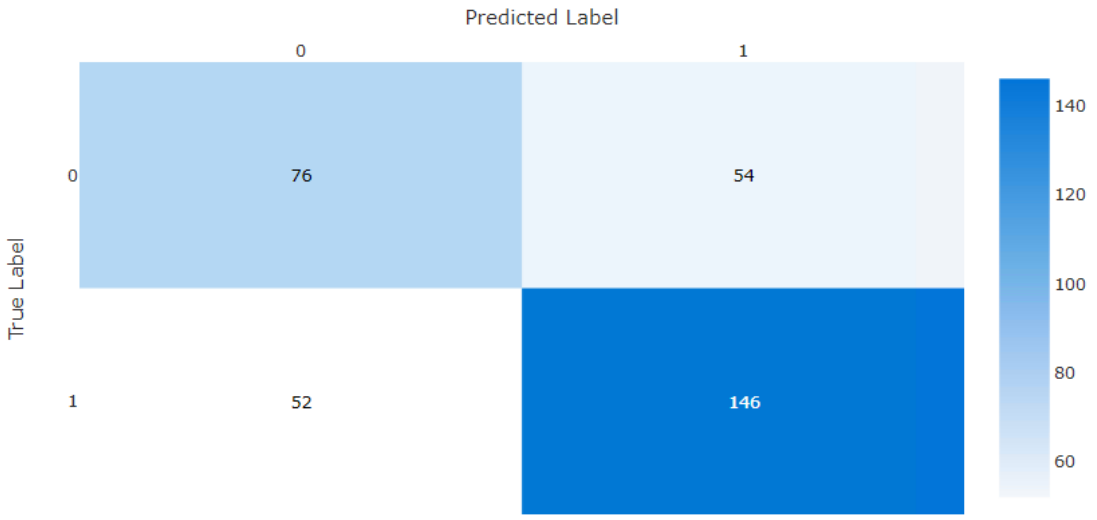


Figure 16. Confusion matrix for the random forest model with an optimal set of features.

9.5 Parameter tuning

Random forest with feature subset 10, is trained with 300 iterations during one or five hours with five folds for cross-validation. The number of iterations is not changed as the training in all previous runs has stopped before reaching 300. The summary of how the model performs is presented in Table 8. The 2nd run receives higher scoring for Matthews's correlation coefficient, recall macro and balanced accuracy and close to macro and weighted AUC score for the 1st run.

Table 8. The metrics of final tuning of random forest with feature subset 10.

Run	Time	Matthews correlation coefficient	F1 macro	AUC weighted	AUC macro	Recall macro	Balanced accuracy
1st	1h	0.327	0.662	0.705	0.705	0.663	0.663
2nd	5h	**0.346**	**0.669**	**0.71**	**0.71**	**0.671**	**0.671**

9.6 Final model

9.6.1 Feature importance

The 2nd run in Table 8's feature importance is presented in Figure 17. The 3 most important features are the same as for the feature for run 10 in Figure 14 and the rest are on a similar level as before.

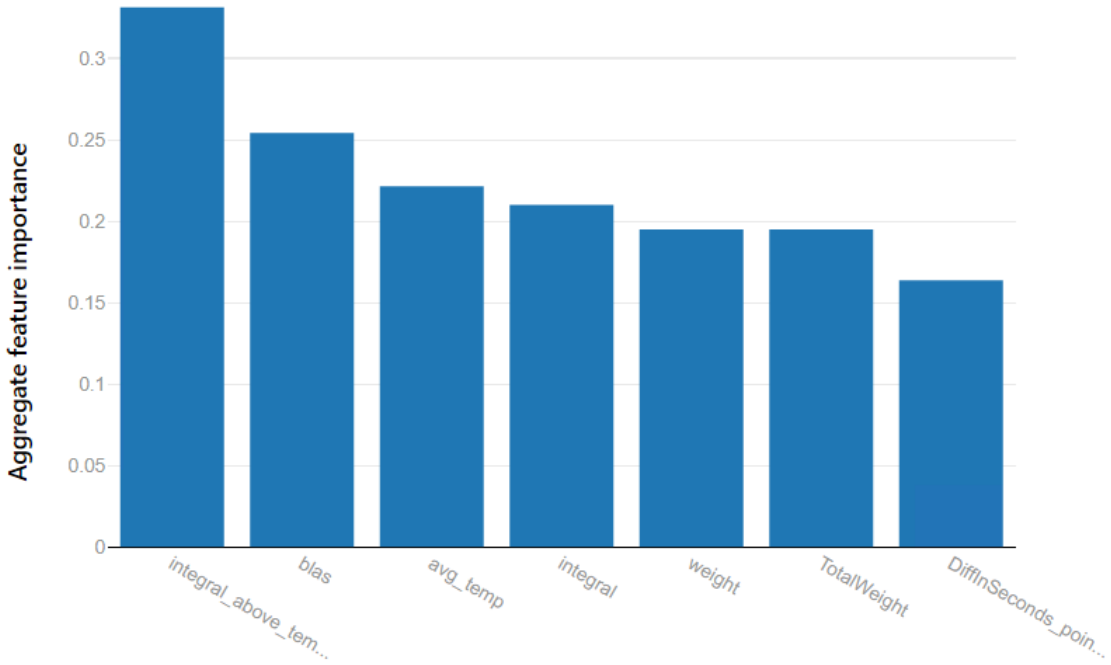


Figure 17. Feature importance for the model from 3rd round with best performing feature subset given features, time and folds for cross-validation.

The AUC in the ROC curve for the 2nd run can be seen in Figure 18. The macro and weighted lines are in between the random and ideal lines, just like the ROC curves in Figure 12 and Figure 15.

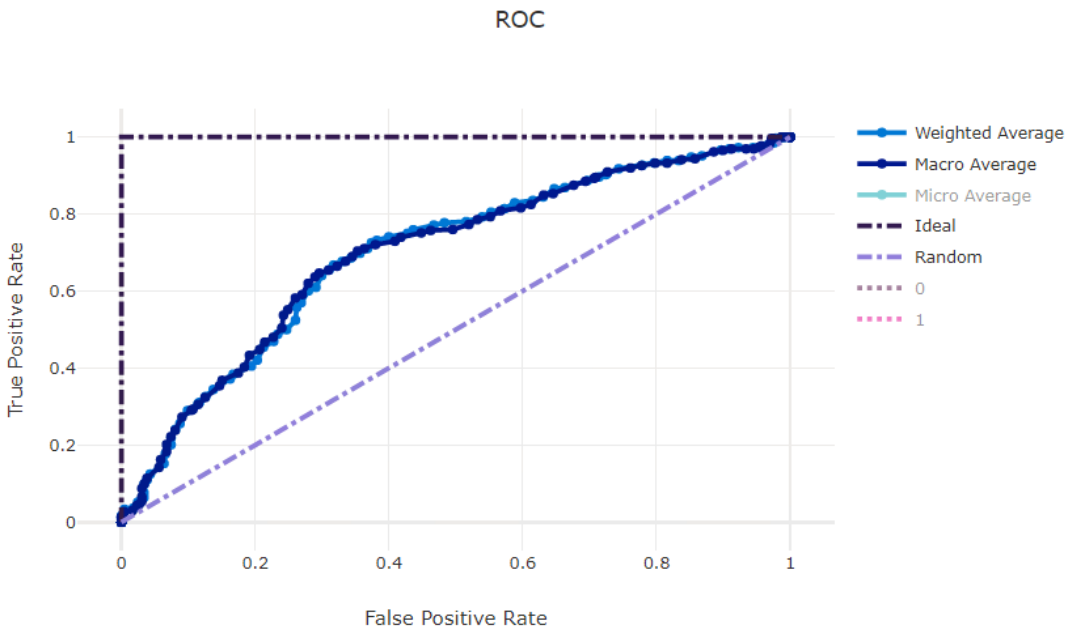


Figure 18. The ROC curve for the 2rd run showing that the average performs better than random.

The confusion matrix for the 2nd run in Figure 19 have similar results as previous trained model with feature subset 10 in Figure 16. The 2nd run is better at predicting approved batches which has increased from 146 to 154. It is also somewhat worse in predicting rejected batches which has decreased from 76 to 73. Overall, the model predicts the majority of all test batches correctly.

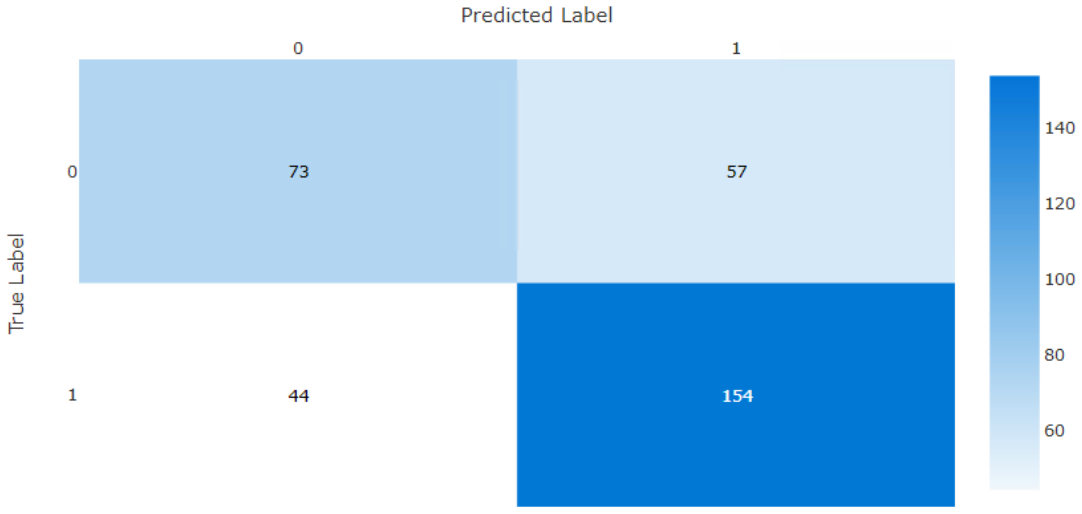


Figure 19. Confusion matrix for the random forest model with an optimal set of features.

9.6.2 Feature impact on predicted quality

Figure 20-26 shows the separate feature impact on the model prediction. The scale goes from -1 to 1. The closer -1 the points are, the greater importance it has for the model to predict an approved batch. The closer to 1 the points are, the more likely the model is going to predict a rejected batch. The feature importance for the integral of cooling measures above 100 degrees Celsius in Figure 20 shows that an integral below 300 000 area units is more likely to be approved. An integral between 322 000 and 480 000 area units makes more batches rejected. Area units above 600 000 have less impact on the quality output.

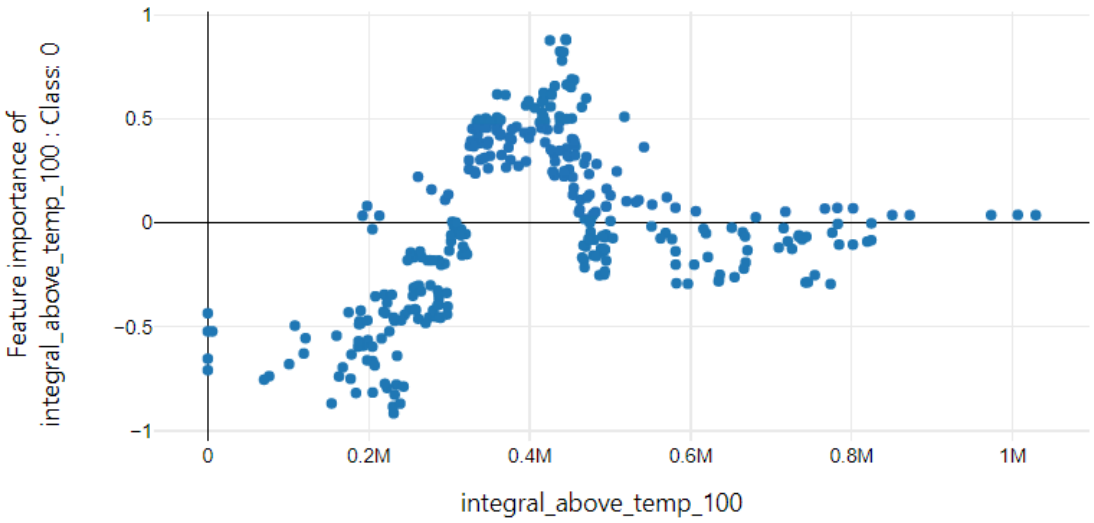


Figure 20. The impact of the cooling curves integral above the temperature 100 degrees Celsius, where the integral between 322 000 and 480 000 area units have a larger rejection importance.

The feature importance of which fan the batch passes can be seen in Figure 21. Fan 2, in the second column, has an important impact on predicting rejected batches. Fan 7 clearly has more approved. Fan 4, 6 and 8 are just under the line and indicate a small acceptance rate. The rest of the fans have a neutral impact on the hardness.

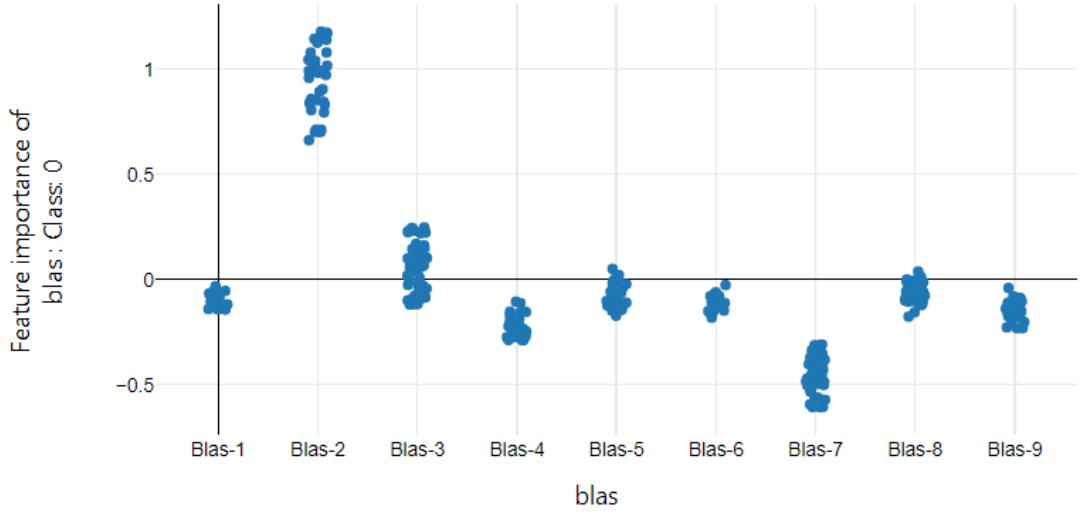


Figure 21. The importance of what fan the batches pass. Fan two has a larger impact on the quality output where the rejection probability is higher.

The average temperature model importance is shown in Figure 22. The temperature outdoors during the operation has an important model impact with a break around +6 degrees. Batches that are run when the outdoor temperature is below +6 degree Celsius are more likely to be rejected.

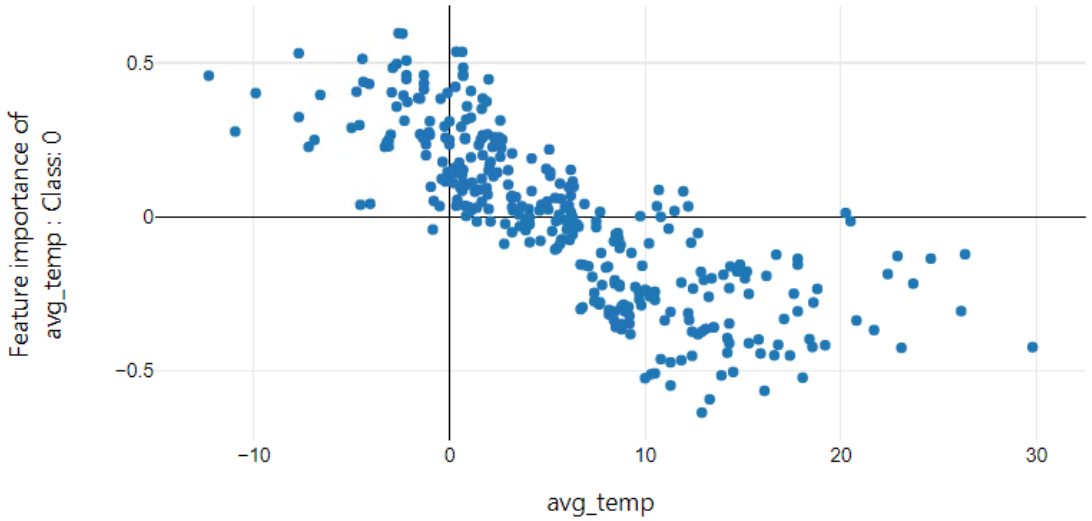


Figure 22. The average temperature and its quality importance where an outdoor temperature below +6 degrees has a negative impact on the quality output.

The recipe dependent integral for each cooling curve is visualized in Figure 23. There are two outstanding areas where an integral below 200 000 area units is more probable to result in rejection and an area in between 200 and 300 000 results in an approved batch.

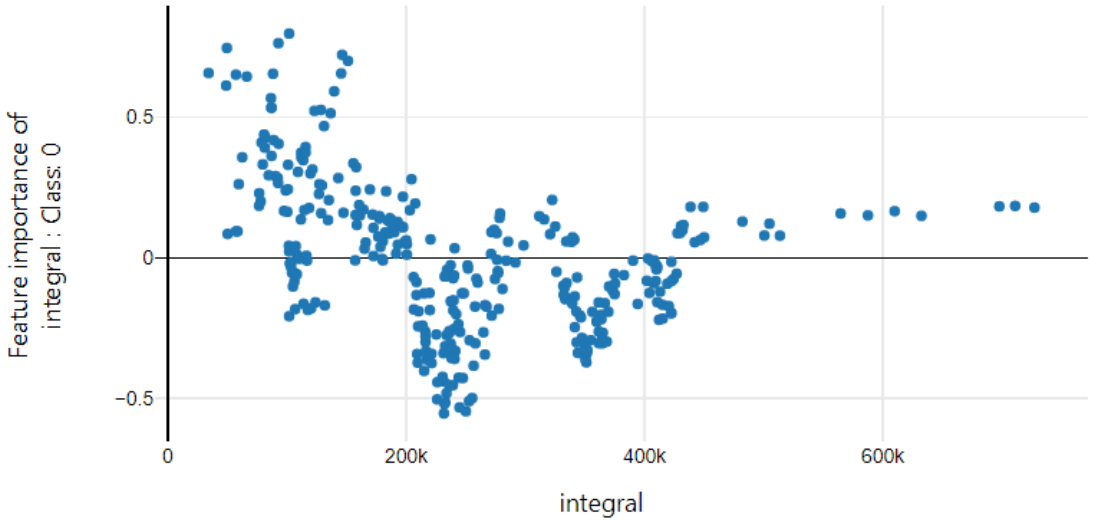


Figure 23. The integral of the cooling curve and its importance on the quality output. The batches have an impact cluster wise.

The individual piece weights model importance are shown in Figure 24 and tells that pieces with a weight above 20 kg are more likely to result in a rejected batch while pieces less than 22 generally result in approved bathes.

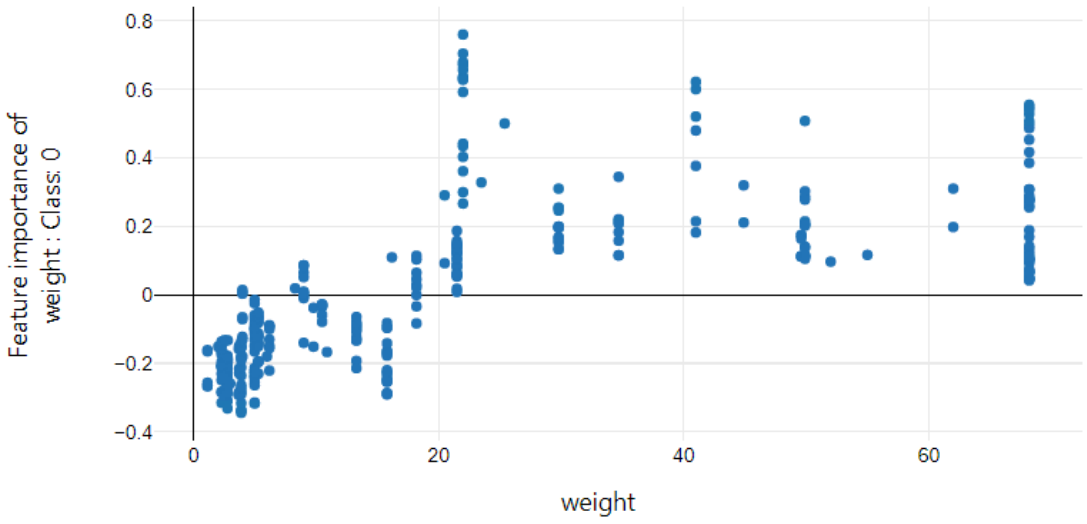


Figure 24. The individual piece weight and its importance of quality output, where in general a smaller weight gives a better quality output.

The feature importance for total batch weight is seen in Figure 25. A batch weight less than 900 kg is more likely to return products that will not reach the quality standards.

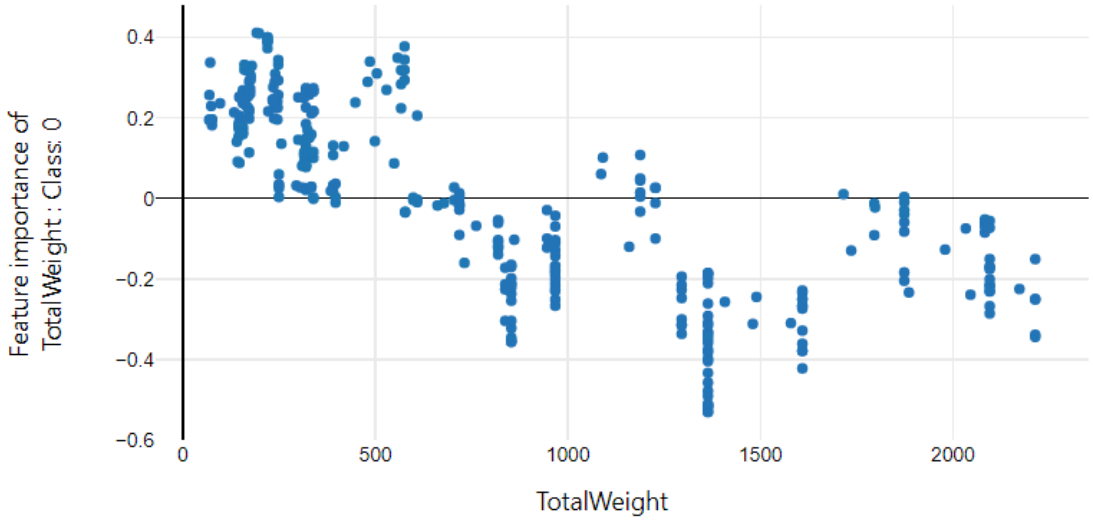


Figure 25. The importance of the total batch weight in kg where a smaller weight has a negative impact on the quality.

Figure 26 visualizes the feature importance for the difference in seconds when the material rests between point 3 to point 4 in Figure 8. When the difference is less than 300 seconds, the material has either not rested long enough or the wrong recipe has been used for the specific fan, which causes rejection. A clear trend is seen from 400 to 1300 seconds for which batches in the interval between 754 and 1157 seconds are more likely to be rejected. For longer differences, the probability of rejection is unclear.

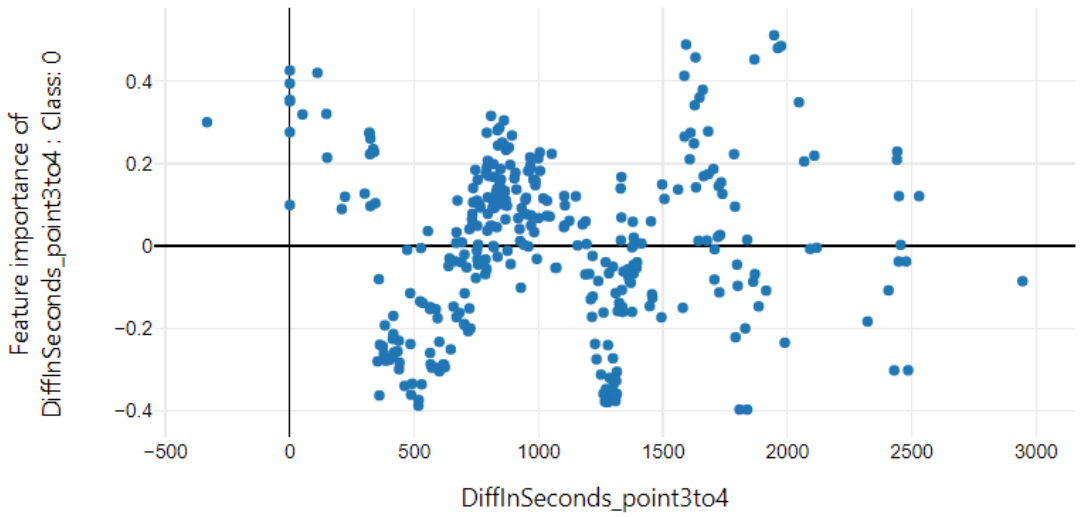


Figure 26. The importance of the difference in seconds from point 3 to 4. A clear trend is seen in the interval between 400 and 1300 seconds.

9.6.3 Testing the model

The final model is tested with the test dataset and the results can be seen in Figure 27. The test results show that out of the 25 approved batches, 20 are correctly classified, which corresponds to a marksmanship of 80 percent. The model is not as good in predicting the rejected batches, which are correctly classified in 5 out of 16 samples which corresponds to 31 percent. In total the models predict 61 percent correctly classified quality results.

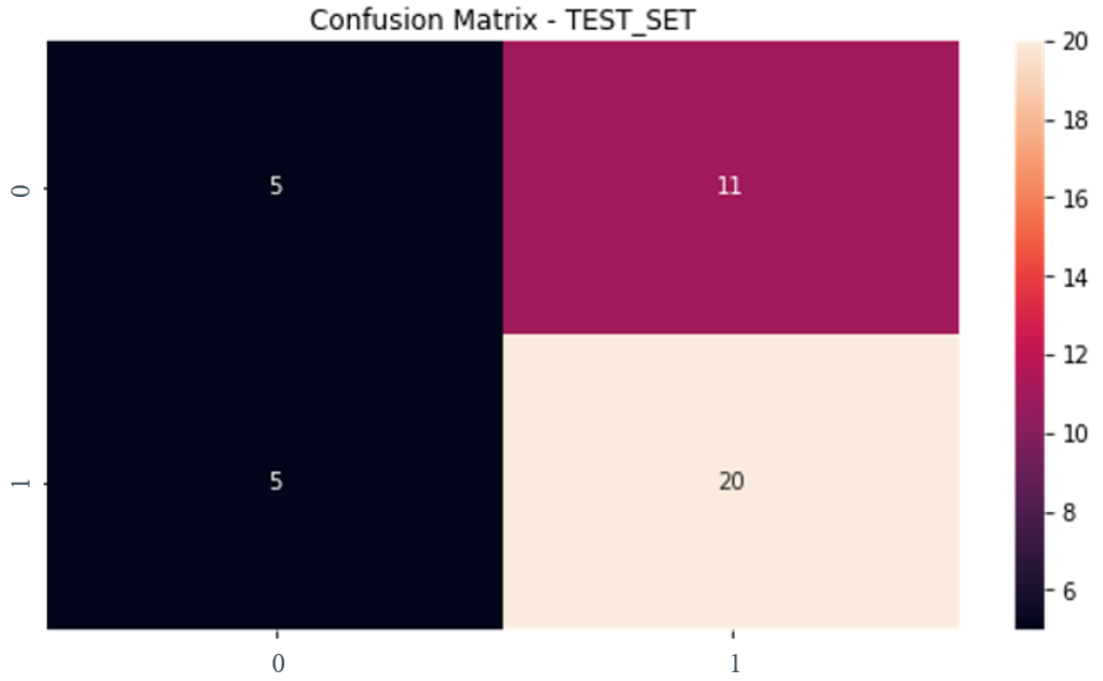


Figure 27. Confusion matrix for the random forest model with an optimal set of features, tested with test data, which the model has never seen before. The y axis is the true label and the x axis is the predicted label.

9.7 Visualization on a BI platform

The model output that performs the best is deployed to the project data lake. The test results dataset is then fed to Power BI desktop which enables interactive visualization of the new data. The visualizations are structured in a report with plots to ease interpretation of data. The platform can be reached on the facility's premises on the local intranet and is visualizing the predicted quality for the validate dataset and its feature values for each individual batch. Figure 28 shows a page in the Power BI report. There are five slicers on the top left for recipe, fan, product code, predicted quality and batch weight. The top left visualization is a bar chart visualizing the ratio of each fan and predicted label. The top right visualization is a line chart visualizing the mean integral above 100 degree Celsius for each batch weight for approved and rejected batches. The pie chart to the bottom left visualizes the amount of batches that have passed each fan. The bottom table is the summary of all features and predicted status. The predicted data is coupled with the raw process data for traceability of batch number, weight and physical measure results.

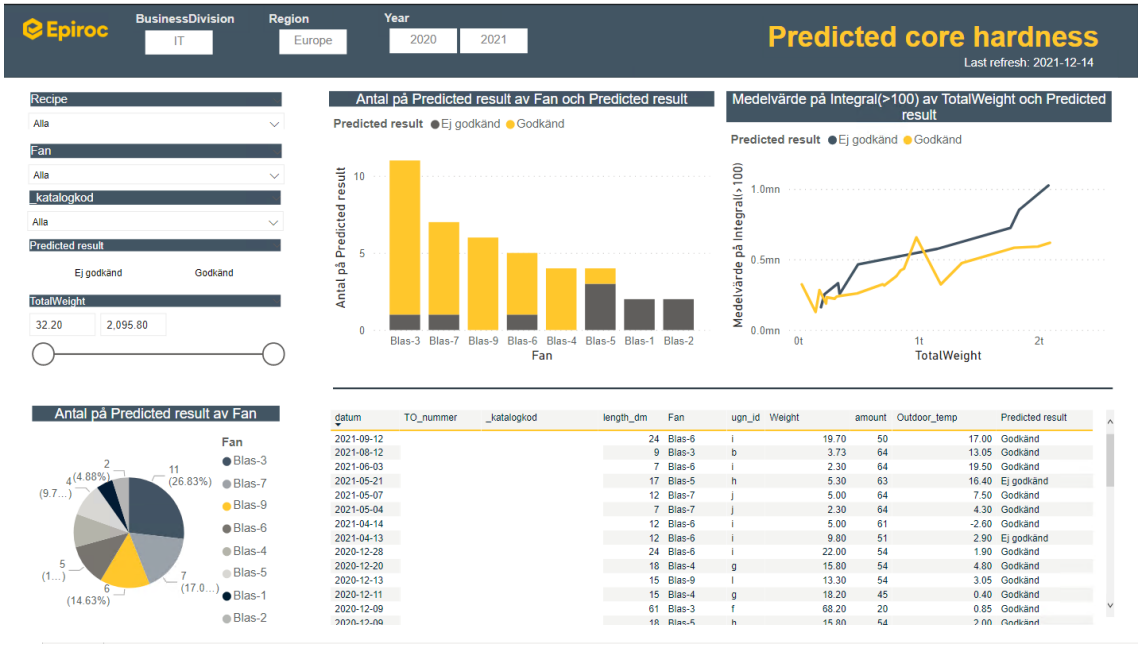


Figure 28. Power BI page with visualizations of variable measures divided in predicted, approved and rejected batches.

10. Discussion

This section will discuss the results from the AI pipeline which is divided into two parts discussing each of the research questions. Section 10.1 discusses the first research question; data and related features that are important for the quality output. Section 10.2 will discuss the second research question; the AI-pipeline and the use of its implementation. Lastly, Section 10.3 will clearly state how the result answers each of the research questions.

10.1 Feature impact on quality prediction

In the investigation of which data has an impact on core hardness, domain knowledge from Respondent 1, 3, 5 and 7 are consistent with the result for feature importance in Figure 17. Heat energy in the fan, outdoor temperature, total and unit weight have been shown to be important while factors such as length, oven ID and recipe name have less impact. This is in line with the theory from [15], that anomalies can be distinguished from raw data points, but also sequences of data. As [14] describes, time series data can be used to find anomalies in an industrial process. From the examined data, insights have been gained based on calculations and sequences of the time series from the cooling process together with information about the batch itself. Thus, both process-external information and calculated values from time series are important factors in addition to raw process data when predicting quality results in heat treatment of steel rods for rock drilling.

The integral of the cooling curve above the temperature 100 degrees Celsius is the most important model feature of all variables. This is plausible as the integral represents the heat energy in each heat-treated batch. According to [28], data points extracted from time series represented by a given interval, may be the foundation of finding anomalies, which the results of this report demonstrates. The fan that the batch passes is, according to the model, the second most important model feature. As Respondent 1 stated, the fans are individuals and the quality output is behaving as such. From the results, a clear pattern can be seen which tells that one fan has a higher rejection rate than others, which confirms the outcome of the pre-study. The mean outdoor temperature for each batch seems to have a relation to the core hardness. The temperature has a clear limit at +6 degrees Celsius, below which, the different batches tend to have a large probability of being discarded. This is in line with what Respondent 1 mentioned about unverified seasonality. The integral of the cooling curves between points 3 and 4, in Figure 23, generates a clear difference between the integral below 200 000 area units and above. An integral below this impacts the model by promoting poor quality outcomes. The two subsequent clusters that are mostly below the y-axis give a positive quality outcome. The result from this feature indicates that the optimal integral is above 200 000 and below 420 000 area units. The individual piece weight is the fifth most important feature for the model. A piece weight of less than 20 kg gives a good quality result and the cooling process for these seems to generate a high degree of approval. Given recommendations from Respondent

1, these rods are easier to produce. Thus, the model's interpretation of the individual piece weight matches the expectations. Larger unit weight gives a wider outcome. The total batch weight has an impact on the predicted quality outcome, which also was confirmed by Respondent 1. Total batch weights below 600 kg tend to result in rejections while heavier batch weights more likely result in approved batches. This can be a result of unfilled batches where a smaller amount of rods are heat treated than usual and the standard recipe for each product is only intended for a certain amount of rods. The result obtained from the difference in seconds between Point 3 and 4 indicates that there are two optimal time intervals for the model to approve a batch.

10.2 The ML process and pipeline

10.2.1 Choosing the best ML algorithm

The results in Table 6 shows that there are several trained algorithms that are candidates for predicting hardness. Light GBD, extremely randomized trees and random forest receive similar scoring. However, random trees receive the highest macro AUC and get high scores also on other metrics, which makes it the strongest candidate to look further into for feature selection and parameter tuning. In the confusion matrix for random forest in Figure 13, there are more falsely classified positives than correctly classified negatives. However, it can classify most of the positives correctly. The ROC curve in Figure 15 also implies that the algorithm has found patterns from the features that have an impact on the data, as the AUC is greater than 70 percent and thus forecast better than a random guess. Based on these facts, the random forest algorithm is selected as a starting point in finding the optimal set of features and parameter tuning.

A similar approach is used to decide the optimal set of features for the algorithm. Based on the results in Table 7, feature subset 10 receives second highest scoring on macro AUC and significantly higher scoring on the other performance metrics. The features for 10 and their order of importance makes sense as Respondent 1 suspected the difference in the fans, the impact of outdoor temperature and amount of energy in the cooling process which is measured by the integral of the curve. Because of the realistic feature importance, overall high scoring and improved results from the confusion matrix, feature subset 10 is chosen to go forward with for the tuning. As [27] stated, the ML model got a better score for a specific set of the features while irrelevant features resulted in a poorer model during the training process.

The final tuning of random forest with the feature subset 10 trained in one and five hours respectively is shown in Table 8. The 2nd run shows a strong performance improvement for all metrics. The confusion matrix for the second run in Figure 19 is better at classifying the approved batches and predicts a few more falsely approved batches than for the previous confusion matrix in Figure 19. Overall, the 2nd run is preferred as it is superior in performance metrics and capacity of predicting approved batches.

10.2.2 Test results of the model

Training the model on unseen data from the test dataset, seen in confusion matrix in Figure 27, gives a realistic result. The previous confusion matrix on the test data has hinted that the model is better at predicting approved batches than rejected ones. Due to the limited amount of data and delimited research area, the model is predicting as good can be expected. The model needs overall further improvement, especially in detecting rejected batches. In a production point of view a falsely classified approved batch would lead to greater harm, as the predicted rejected ones probably would be prioritized for extra quality checks. The results shows that the model is not yet ready to be put in production, but rather be seen as a good starting point regarding what type of data, features and algorithms that can be used in both understanding the cooling process scientifically and streamline further research. A model trained on more granular data, such as the temperature of the air in production, or more samples of data would make an even better basis for pointing out batches that do not need to be tested. In a perfect scenario, only batches on the border of the quality tolerance interval would need to be tested, which would increase effectiveness and reduce lead times at the quality department.

10.2.3 The AI pipeline

The AI pipeline that has been built for the project has shown to be useful in predicting quality output. Fetching data from cloud, cleaning and extracting features from bronze to silver and joining them to gold has been useful in analyzing the problem with ML given the important insights presented in the previous section. The pipeline saves a lot of time and effort, as data is easily accessible in any of the pipeline steps. If any of the steps need to be updated with data, the following steps in the pipeline do not need to be rewritten. This makes the pipeline very easy to use for further development and scaling out the research with more data or variables. The architecture simply reduces the distance of taking a research proof of concept to an integrated operationalized state.

This thesis has been able to use the AI pipeline to predict quality output based on data related to the cooling process. As [19] stated, ML can in this case be used for advanced process control and innovation in manufacturing efficiency. Creating a model to improve production and quality control, as [25] describes, has also shown to be a suitable strategy for use cases within heat treatment of rods. A full AI pipeline has been implemented. Raw process data has been transformed to valuable output within the project time which includes training a great number of hyperparameters to answer the research questions. This proves the AI pipelines value in decreasing time and effort in predicting quality output for the cooling process of heat treated rods.

10.2.4 Visualized prediction

The use of a business intelligence tool to visualize the predicted status of each batch is found to be a crucial part of understanding the ML predicted test result. Just like [29] stated, a visualization platform is important in order to interpret scientific findings and to explain the added value of the model. Vellido [30] states that the visualization itself is a knowledge generator and is a crucial part in finding incompleteness, diagnosis of the result and the ML model refinement. In this thesis the plots of the ML output have been important in these three cases. Power BI can be used in an AI pipeline to fill a visualization gap and help the end user to interpret the model output.

10.3 Answering research questions

The first research question, given the previous discussion in Section 10.1, is answered by concluding that three types of data are important. Single data points about the process, external variables about the weather and process time series have been shown to have an impact on the quality. The total heat energy in the oven, characteristics of the fan and the temperature of the air that it cools the material with has a greater impact on the final model compared to other investigated variables. This is also applicable to the size of the rod and the total amount of rods in the batch.

The second research question is answered by the extent of work that has been conducted during a five months' time and the resulting value for the heat treatment and cooling of rods that the AI pipeline has made possible. An AI pipeline is applicable for the rock drilling tool manufacturing industry and can be efficient with the services used in this research for further development, scalability and deployment for production.

11. Conclusion

This study aims to find and test variables that can have an impact on hardness quality results of heat treated steel rods in mining manufacturing. Concurrently, an AI pipeline's applicability and ability to speed up the process is tested in finding interrelated variables affecting the quality. Previous studies investigating the general applicability of AI in the steel manufacturing industry show that the area is a perfect candidate due to its complex interrelations of steel composition and process steps. A lot of different chemical steel components and process variables have been used in combination with deep learning for unsupervised problems while less studies have been conducted to deeply investigate the cooling process to classify output quality with a clustering approach.

As no studies have been found for air cooled steel rods manufactured for mining, or even the steel industry, this thesis fills a knowledge gap both as proof of using an AI pipeline and testing new variables from cooling to improve quality or rods. The heat treatment process is expensive and consumes a lot of energy. The process of manufacturing perfect quality is not fully understood due to its complexity. Thus, this investigation is needed for minimization of resource use, streamline production and find the right variables and models to be industrialized.

The investigation is based on a case study where knowledge from interviews builds the ground for feature extraction. The AI pipeline is powered by Microsoft services through which raw process data is collected, cleaned, feature engineered, predicted on and visualized. Through the experimental procedure nine algorithms are compared, out of which the best performing algorithm is trained with 11 different feature subsets. The best performing subset is hyperparameter tuned to get a model as good as possible. Six performance metrics are used to evaluate the models, out of which macro AUC is the primary one.

In summary, the random forest algorithm is a suitable algorithm to predict quality results from the cooling process in manufacturing of rods for the mining industry. This report also shows that creating an AI pipeline has been efficient in the designed way that has been adopted for this use case. Data can be stored and fetched in a cloud instance using the Databricks platform to engineer data for use in ML. The tuned model can be deployed and utilized for further insights on a business intelligence platform like Power BI. Process, weather and production time series data have been found to make a good ground in finding correlations between the heat treatment process and quality output. Domain knowledge about the process and steel characteristics is crucial when finding the right features to choose an algorithm. The amount of features has an effect on the model performance and can be optimized in order to improve the algorithm.

As a concluding remark, the authors have proved that ML can be used to improve the quality output from heat treatment rods and that an AI pipeline can be used although improvement can be made in feature selection and availability of larger and more detailed data on site.

12. Future work

Based on the results in this report, further work can be done to improve industrial processes in mining industry manufacturing. We see potential for further implementation of AI in several areas on site, such as sintering of cemented carbide and carbonization during hardening of steel. We also see that other potential data sources and features can be added to the existing AI pipeline, such as composition of the steel, other products and routes between ovens and fans. Given that temperature and its seasonality have an impact on quality, additional weather data such as humidity and air pressure would be interesting to add. There is great potential for further research in the field in combination with ML that is unexplored which gives many possible outlines for future projects.

References

[1] Ledarna, Tillverkningsindustrin ur ett chefsperspektiv. https://www.ledarna.se/branschforeningar/ledarna-teknik-motor/dokument/ (2020-09-22).

[2] Naturvardsverket, Territoriella utsläpp och upptag av växthusgaser. https://www.naturvardsverket.se/data-och-statistik/klimat/vaxthusgaser-territoriella-utslapp-och-upptag (202-09-22).

[3] Jernkontoret (2018), End products of steel. https://www.jernkontoret.se/en/the-steel-industry/production-utilisation-recycling/end-products-of-steel/ (2020-09-22).

[4] National Research Council (2000), Surviving Supply Chain Integration: Strategies for Small Manufacturers. Washington, DC: The National Academies Press.

[5] Wang, J., Wang, Y., Zhang, H., Zhang, Z. (2015), Development of an Evaluating Method for Carbon Emissions of Manufacturing Process Plans, Discrete Dynamics in Nature and Society, 2015.

[6] Chand, S., & Davis, J. F. (2010). What is smart manufacturing? Time Magazine.

[7] Elangovan, M., Sakthivel, N. R., Saravanamurugan, S., Nair, B. B., Sugumaran, V. (2015). Machine learning approach to the prediction of surface roughness using statistical features of vibration signal acquired in turning. Procedia Computer Science, 50, 282–288.

[8] Smart Manufacturing - Shaping Europe's digital future - European Commission [Internet]. Shaping Europe's digital future - European Commission. 2021 [cited 10 September 2021]. Available from: https://wayback.archive-it.org/12090/20210727024631/https://ec.europa.eu/digital-single-market/en/policies/smart-manufacturing

[9] Hapke, H., & Nelson, C. (2020). Building machine learning pipelines. O'Reilly Media, Inc.

[10] Pellegrini, G., Sandri, M., Villagrossi, E., Challapalli, S., Cestari, L., Polo, A., & Ometto, M. (2019). Successful use case applications of artificial intelligence in the steel industry. Paper presented at the AISTech - Iron and Steel Technology Conference Proceedings.

[11] Wuest, T., Weimer, D., Irgens, C., & Thoben, K. (2016). Machine learning in manufacturing: Advantages, challenges, and applications. Production & Manufacturing Research, 4(1), 23-45.

[12] Janning, R., Schmidt-Thieme, L., Spiliopoulou, M. & Gesellschaft für Klassifikation. Jahrestagung University of Hildesheim) 2012 : (36th 2014;2013;, Data analysis, machine learning and knowledge discovery) Springer, Cham; New York.

[13] Sarker, I.H. (2021) Machine Learning: Algorithms, Real-World Applications and Research Directions. SN Computer Science 2, Article number 160.

[14] Mehrotra, K.G., Mohan, C.K., Huang, H. & SpringerLink(Online service) (2017), Anomaly Detection Principles and Algorithms, Springer International Publishing, Cham.

[15] Aggarwal, C. C. (2014;2015;). In Aggarwal C. C. (Ed.), Data classification: Algorithms and applications. CRC Press, Taylor & Francis Group.

[16] Bryson, W. E. (2015). Heat treatment: Master control manual. Hanser Publishers.

[17] Monostori, L., Hornyák, J., Egresits, C., & Viharos, Z. J. (1998). Soft computing and hybrid AI approaches to intelligent manufacturing. Tasks and Methods in Applied Artificial Intelligence Lecture Notes in Computer Science, 1416, 765–774.

[18] Alpaydin, E. (2014). Introduction to machine learning (Third ed.). The MIT Press.

[19] Guillen, D. P. (2020). Machine learning applications in advanced manufacturing processes. Jom 72(11), 3906-3907.

[20] Velicer, W. F., Fava, J. L. (2003). Time Series Analysis. Handbook of psychology: Research methods in psychology, 2, 581–606, John Wiley & Sons Inc.

[21] Liu, H., Jia, Z., Wang, F. J., Zong, F., (2013) Study on a giant magnetostrictive actuator with constant output force. International Journal of Industrial and Systems Engineering 13:2, 197-218.

[22] Köksal, G., Batmaz, İ., & Testik, M. C. (2011). A review of data mining applications for quality improvement in manufacturing industry. Expert Systems with Applications, 38(10), 13448-13467.

[23] Sathya, R., Abraham, A., (2013) Comparison of supervised and unsupervised learning algorithms for pattern classification. International Journal of Advanced Research in Artificial Intelligence; 2(2):34-8.

[24] Maglogiannis, I. G. (2007). Emerging artificial intelligence applications in computer engineering: Real word AI systems with applications in eHealth, HCI, information retrieval and pervasive technologies. IOS Press.

[25] Love, B. C. (2002). Comparing supervised and unsupervised category learning. Psychonomic Bulletin & Review, 9(4), 829-835.

[26] Wang, Y., Martonosi, M., & Peh, L. (2007). Predicting link quality using supervised learning in wireless sensor networks. Mobile Computing and Communications Review, 11(3), 71-83.

[27] Tang J, Alelyani S, Liu H. Feature selection for classification: A review. In Data Classification: Algorithms and Applications. CRC Press. 2014. p. 37-64.

[28] Ren, H., Liu, M., Liao, X., Liang, L., Ye, Z., & Li, Z. (2018). Anomaly detection in time series based on interval sets. IEEJ Transactions on Electrical and Electronic Engineering, 13(5), 757-762.

[29] Rodriguez, M., J., C. (2019). Section 4. data visualization and advanced machine learning. (pp. 1-2). Packt Publishing.

[30] Vellido, A. (2019;2020;). The importance of interpretability and visualization in machine learning for applications in medicine and health care. Neural Computing & Applications, 32(24), 18069-18083.

[31] Ruiz, E., Ferreño, D., Cuartas, M., Lloret, L., Ruiz del Árbol, Pablo M, López, A., Esteve, F., & Gutiérrez-Solana, F. (2021). Machine learning methods for the prediction of the inclusion content of clean steel fabricated by electric arc furnace and rolling. Metals (Basel), 11(6), 914.

[32] Cemernek, D., Cemernek, S., Gursch, H., Pandeshwar, A., Leitner, T., Berger, M., Klösch, G., & Kern, R. (2021). Machine learning in continuous casting of steel: A state-of-the-art survey. Journal of Intelligent Manufacturing

[33] Clarke, C. J., & Carswell, B. (2007). Principles of astrophysical fluid dynamics. Cambridge University Press.

[34] Mitra, S., Singh, R. K., & Mondal, A. K. (2014). An expert system based process control system for silicon steel mill furnace of rourkela steel plant. Paper presented at the 29-33.

[35] Mitra, S., Saha, S. K., & Ghosh, N. K. (2012). Application of expert system for heating control of annealing furnaces in a cold rolling mill - A case study at bokaro steel plant. Paper presented at the 245-249.

[36] Li, F., Wu, J., Dong, F., Lin, J., Sun, G., Chen, H., & Shen, J. (2018). Ensemble machine learning systems for the estimation of steel quality control. Paper presented at the 2245-2252.

[37] Carneiro, M. V., Salis, T. T., Almeida, G. M., & Braga, A. P. (2021). Prediction of mechanical properties of steel tubes using a machine learning approach. Journal of Materials Engineering and Performance, 30(1), 434-443.

[38] Varfolomeev, I. A., Ershov, E. V., & Vinogradova, L. N. (2018). Statistical control of defects in a continuously cast billet based on machine learning and data analysis methods. Automation and Remote Control, 79(8), 1450-1457.

[39] Gong, R., Wu, C., & Chu, M. (2018). Steel surface defect classification using multiple hyper-spheres support vector machine with additional information. Chemometrics and Intelligent Laboratory Systems, 172, 109-117.

[40] Zheng, X., Zheng, S., Kong, Y., & Chen, J. (2021). Recent advances in surface defect inspection of industrial products using deep learning techniques. International Journal of Advanced Manufacturing Technology, 113(1-2), 35-58.

[41] Damacharla, P., M. V, A. R., Ringenberg, J., & Javaid, A. Y. (2021). TLU-net: A deep learning approach for automatic steel surface defect detection. Paper presented at the 1-6.

[42] Tsutsui, K., Terasaki, H., Uto, K., Maemura, T., Hiramatsu, S., Hayashi, K., Moriguchi, K., & Morito, S. (2020). A methodology of steel microstructure recognition using SEM images by machine learning based on textural analysis. Materials Today Communications, 25, 101514.

[43] Panda, A., Naskar, R., & Pal, S. (2019). Deep learning approach for segmentation of plain carbon steel microstructure images. IET Image Processing, 13(9), 1516-1524.

[44] DeCost, B. L., Francis, T., & Holm, E. A. (2017). Exploring the microstructure manifold: Image texture representations applied to ultrahigh carbon steel microstructures. Acta Materialia, 133, 30-40.

[45] Pattanayak, S., Dey, S., Chatterjee, S., Chowdhury, S. G., & Datta, S. (2015). Computational intelligence based designing of microalloyed pipeline steel. Computational Materials Science

[46] Agrawal, A., Deshpande, P. D., Cecen, A., Basavarsu, G. P., Choudhary, A. N., & Kalidindi, S. R. (2014). Exploration of data science techniques to predict fatigue strength of steel from composition and processing parameters. Integrating Materials and Manufacturing Innovation, 3(1), 90-108

[47] Jones, D. M., Watton, J., & Brown, K. J. (2005). Comparison of hot rolled steel mechanical property prediction models using linear multiple regression, non-linear multiple regression and non-linear artificial neural networks. Ironmaking & Steelmaking, 32(5), 435-442.

[48] Xie, Q., Suvarna, M., Li, J., Zhu, X., Cai, J., & Wang, X. (2021). Online prediction of mechanical properties of hot rolled steel plate using machine learning. Materials & Design, 197, 109201

[49] Hanza, S. S., Marohnić, T., Iljkić, D., & Basan, R. (2021). Artificial neural networks-based prediction of hardness of low-alloy steels using specific jominy distance. Metals (Basel), 11(714), 714

[50] Bryman, A., Bell, E., & Nilsson, B. (2017). Företagsekonomiska forskningsmetoder (3 ed.). Liber.

[51] Christensen, L., Engdahl, N., Grääs, C., & Haglund, L. (2016). Marknadsundersökning: En handbok (4 ed.). Studentlitteratur.

[52] Kendall, M. G., Sir. (1994). Kendall's advanced theory of statistics: Vol. 1, distribution theory (6.th ed.). Edward Arnold.

[53] Clarke, C. J., & Carswell, B. (2007). Principles of astrophysical fluid dynamics. Cambridge University Press.

[54] Glynn, K., & Wade, C. (2020). Hands-on gradient boosting with XGBoost and scikit-learn. Packt Publishing.

[55] Brownlee, J. (2016) Master Machine Learning Algorithms: Discover How They Work and Implement Them From Scratch, Jason Brownlee

[56] Géron, A. (2017). Understanding support vector machines (First ed.). O'Reilly Media.

[57] Lindholm, A., Wahlström, N., Lindsten, F., Schön, T. B., (2021) Machine Learning - A First Course for Engineers and Scientists, http://smlbook.org/.

[58] Criminisi, A., Konukoglu, E., & Shotton, J. (2011). Decision Forests for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning. Microsoft Academic.

[59] Scikit-learn, Ensemble methods, https://scikit-learn.org/stable/modules/ensemble.html (2021-10-17)

[60] Quinto, B., & SpringerLink (Online service). (2020). Next-generation machine learning with spark: Covers XGBoost, LightGBM, spark NLP, distributed deep learning with keras, and more (1st 2020. ed.). Apress.

[61] Sammut, C., & Webb, G. I. (2010). Encyclopedia of machine learning. Springer.

[62] Microsoft Docs (2021), Evaluate automated machine learning experiment result, https://docs.microsoft.com/en-us/azure/machine-learning/how-to-understand-automated-ml (2021-10-28)

[63] Hand, D., & Christen, P. (2017;2018;). A note on using the F-measure for evaluating record linkage algorithms. Statistics and Computing, 28(3), 539-547. https://doi.org/10.1007/s11222-017-9746-6

[64] Scikit-learn, Metrics and scoring: quantifying the quality of predictions, https://scikit-learn.org/0.22/modules/model_evaluation.html#matthews-corrcoef (2021-12-06)

[65] Li, J., & Fine, J. P. (2010). Weighted area under the receiver operating characteristic curve and its application to gene selection: Weighted area under receiver operating characteristic curve. Journal of the Royal Statistical Society: Series C (Applied Statistics)

[66] Herrera, F., Charte, F., Rivera, A. J., del Jesus, M. J., & SpringerLink (Online service). (2016). Multilabel classification: Problem analysis, metrics and techniques. Springer International Publishing.

[67] James, G. D., & Liebeck, M. W. (2001;2012;). Representations and characters of groups (2nd ed.). Cambridge University Press.

[68] Burez, J., & Van den Poel, D. (2009). Handling class imbalance in customer churn prediction. Expert Systems with Applications, 36(3), 4626-4636.

[69] Soh, J., Singh, P., & SpringerLink (Online service). (2020;2021;). Data science solutions on azure: Tools and techniques using databricks and MLOps (1st 2020. ed.). Apress.

[70] Microsoft Docs, Describe bronze, silver, and gold architecture, https://docs.microsoft.com/en-us/learn/modules/describe-azure-databricks-delta-lake-architecture/2-describe-bronze-silver-gold-architecture (2021-09-22)

[71] Johnson, E. (2017). Introduction to SQL server 2016 integration services (SSIS): Getting started with extract, transform, and load (ETL) using SSIS. Sams.

[72] Alla, S., Amirghodsi, S., Karim, M. R., & Kienzler, R. (2018). Apache spark 2: Data processing and real-time analytics. Packt Publishing.

[73] Laserson, U. (2022). Advanced analytics with PySpark. O'Reilly Media, Inc.

[74] Microsoft Docs, What is Azure Machine Learning?, https://docs.microsoft.com/en-us/azure/machine-learning/overview-what-is-azure-machine-learning (2021-10-08)

[75] Microsoft Docs, What is automated machine learning(AutoML)?, https://docs.microsoft.com/en-us/azure/machine-learning/concept-automated-ml (2021-10-08)

[76] Microsoft Docs, Tune model hyperparameters, https://docs.microsoft.com/en-us/azure/machine-learning/component-reference/tune-model-hyperparameters (2021-09-12)

[77] Microsoft Docs, Set up AutoML training with Python, https://docs.microsoft.com/en-us/azure/machine-learning/how-to-configure-auto-train (2021-10-29)

[78] Larson, B. (2020;2019;). Data analysis with microsoft power BI. McGraw-Hill.

[79] Frännfors, S., https://www.temperatur.nu/fagersta2 (2021-10-14)

[80] SciPy, (2016) scipy.integrate.trapz, https://docs.scipy.org/doc/scipy-0.18.1/reference/generated/scipy.integrate.trapz.html (2021-10-04)

[81] Scikit-learn, sklearn.model_selection.StratifiedShuffleSplit, https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedShuffleSplit.html (2021-10-26)

[82] Vuttipittayamongkol, P., Elyan, E., & Petrovski, A. (2021). On the class overlap problem in imbalanced data classification. Knowledge-Based Systems, 212

[83] Zhu, Q. (2020). On the performance of matthews correlation coefficient (MCC) for imbalanced dataset. Pattern Recognition Letters, 136, 71-80.

[84] Debón, A., & Carlos Garcia-Díaz, J. (2012). Fault diagnosis and comparing risk for the steel coil manufacturing process using statistical models for binary data. Reliability Engineering & System Safety, 100, 102-114.

[85] El-Banna, M. (2017). Modified mahalanobis taguchi system for imbalance data classification. Computational Intelligence and Neuroscience, 2017, 5874896-15.

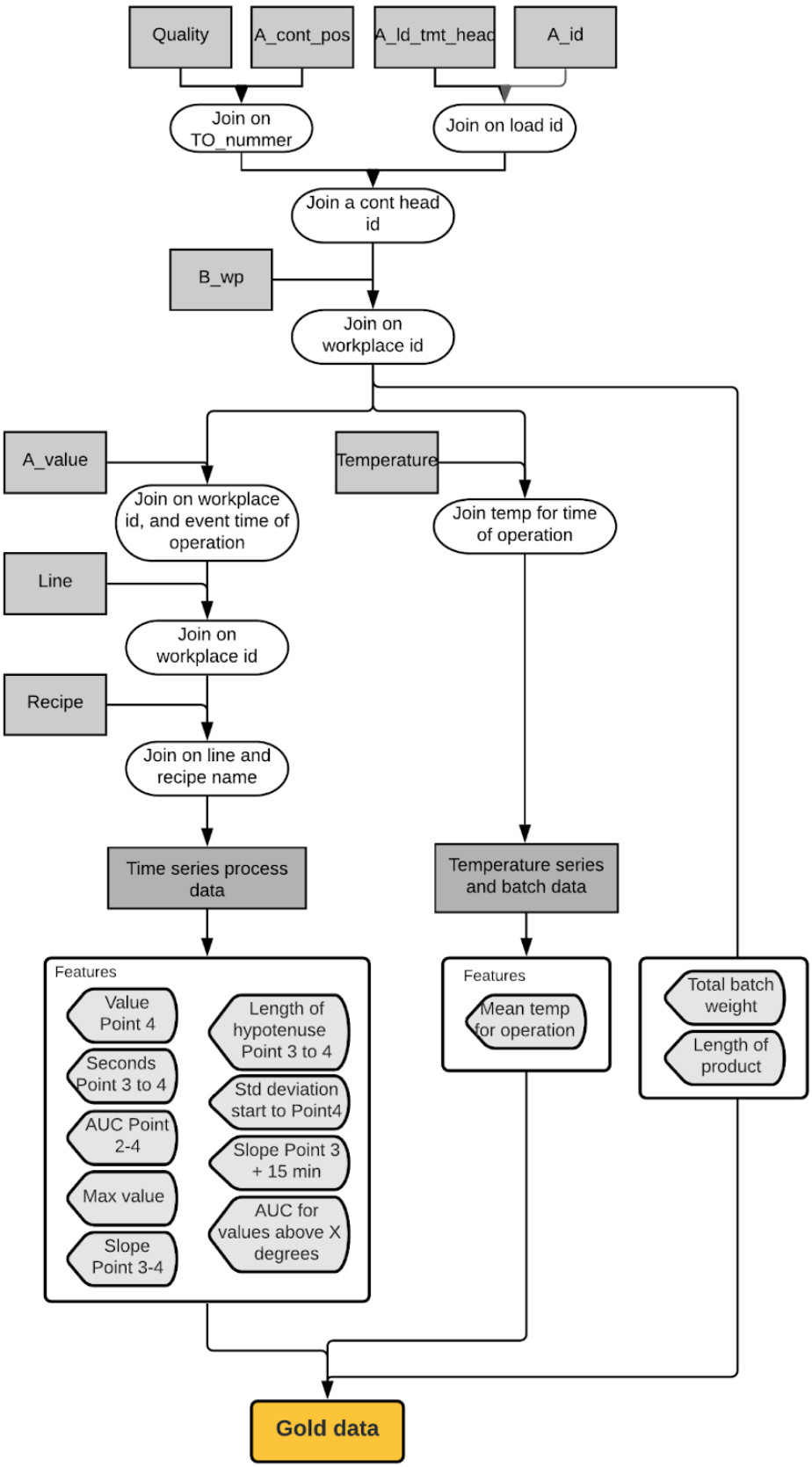
[86] Starovoitov, V. V., & Golub, Y. I. (2020). Comparative study of quality estimation of binary classification. Informatika (Minsk, Belarus), 17(1), 87-101.

Appendix

A: Respondents

Respondent	Role	Time
1	Technician	1 h
2	R&D Materials Manager	2 h
3	Digital Innovation Manager	2 h
4	Quality technician	2 h
5	Operator	2 h
6	Heat treatment researcher	1 h
7	Quality Engineer	30 min + 45 min
8	Microsoft advisory consultant	1 h

B: Joining datasets and feature extraction



C: Feature subset selection for random forest

Feature	1	2	3	4	5	6	7	8	9	**10**	11
TotalWeight	x	x	x	x	x	x	x	x	x	**x**	x
avg_temp	x	x	x	x	x	x	x	x	x	**x**	x
integral_above_temp_100	x	x	x	x	x	x	x	x	x	**x**	x
blas	x	x	x	x	x	x	x	x	x	**x**	x
weight	x	x	x	x	x	x	x	x	x	**x**	x
integral	x	x							x	**x**	
DiffInSeconds_point3to4	x	x	x	x	x	x	x	x	x	**x**	x
slope_max_15	x	x	x	x	x	x	x	x	x		x
ugn_id	x	x	x		x	x	x	x	x		x
max_value_f	x	x	x		x	x	x	x	x		x
slope	x		x								x
length_dm	x		x		x	x	x	x			x
std_f	x		x			x	x	x			x
point4_value_f	x					x	x				
recipe_name	x						x				
amount	x						x				
hypotenuse	x										

D: Hyperparameters for random forest

Hyperparameters

Data transformation:

```
{
    "class_name": "SparseNormalizer",
    "module": "automl.client.core.common.model_wrappers",
    "param_args": [],
    "param_kwargs": {
        "norm": "l2"
    },
    "prepared_kwargs": {},
    "spec_class": "preproc"
}
```

Training algorithm:

```
{
    "class_name": "RandomForestClassifier",
    "module": "sklearn.ensemble",
    "param_args": [],
    "param_kwargs": {
        "bootstrap": true,
        "class_weight": "balanced",
        "criterion": "gini",
        "max_features": "sqrt",
        "min_samples_leaf": 0.01,
        "min_samples_split": 0.01,
        "n_estimators": 100,
        "oob_score": true
    },
    "prepared_kwargs": {},
    "spec_class": "sklearn"
}
```