

**IMPLEMENTATION OF RANDOM FOREST REGRESSION WITH
HYPERPARAMETER TUNING AND SHAP ANALYSIS TO PREDICT
PULL TEST RESULT AT SOFT LAMINATING MACHINE**

UNDERGRADUATE THESIS

**Submitted to the International Undergraduate Program in Industrial Engineering in
Partial Fulfillment of Requirement for the Degree of Sarjana Teknik at the Faculty of**

**Industrial Technology
Universitas Islam Indonesia**



Name : Ar Royyan Utama Tanjung

Student Number : 21522312

**UNDERGRADUATE PROGRAM IN INDUSTRIAL ENGINEERING
FACULTY OF INDUSTRIAL TECHNOLOGY
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2025

AUTHENTICITY STATEMENT

In the name of Allah SWT, I declare this work is the result of my own, except for the those which I have explained the source. If, in the future, it turns out that my declaration is proven to be untrue and violates the legal regulations in the paper and intellectual property rights, in that case, I am willing to get diploma that I have received to be withdrawn by the Universitas Islam Indonesia.

Sleman, 16th May, 2025



A handwritten signature in black ink, appearing to read "Ar Royyan Utama T".

Ar Royyan Utama T
21522312

RESEARCH COMPLETION LETTER



FAKULTAS
TEKNOLOGI INDUSTRI

Gedung KH. Mas Mansur
Kampus Terpadu Universitas Islam Indonesia
Jl. Kaliurang km 14,5 Yogyakarta 55584
T. (0274) 898444 ext. 4100, 4101
F. (0274) 895007
E. fti@uii.ac.id
W. fti.uii.ac.id

SURAT KETERANGAN PENELITIAN

Nomor: 02/Ka.Lab.Datmin/70/Lab.Datmin/V/2025

Assalamu'alaikum Warahmatullahi Wabarakatuh

Kami yang bertanda tangan dibawah ini, menerangkan bahwa mahasiswa dengan keterangan sebagai berikut :

Nama : Ar Royyan Utama T
No. Mhs : 21522312
Dosen Pembimbing : Ir. Muhammad Ridwan Andi Purnomo, S.T., M.Sc., Ph.D., IPM.

Telah selesai melaksanakan penelitian yang berjudul "*IMPLEMENTATION OF RANDOM FOREST REGRESSION WITH HYPERPARAMETER TUNING AND SHAP ANALYSIS TO PREDICT PULL TEST RESULT AT SOFT LAMINATING MACHINE*" di Laboratorium Data Mining, Program Studi Teknik Industri, Fakultas Teknologi Industri, Universitas Islam Indonesia tercatat mulai tanggal 12 November sampai dengan tanggal 20 Desember 2024

Demikian surat keterangan kami keluarkan, agar dapat dipergunakan sebagaimana mestinya.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Yogyakarta, 06 Mei 2025

Kepala Laboratorium
Data Mining

Dr. Dwi Adi Purnama, S.T.

SUPERVISOR APPROVAL SHEET
IMPLEMENTATION OF RANDOM FOREST REGRESSION WITH
HYPERPARAMETER TUNING AND SHAP ANALYSIS TO PREDICT PULL TEST
RESULT AT SOFT LAMINATING MACHINE

UNDERGRADUATE THESIS

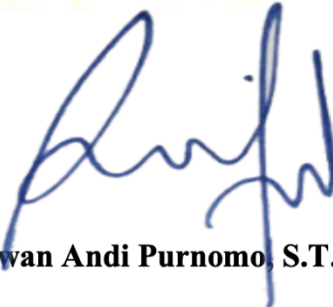
Written by:

Name : Ar Royyan Utama T

Student Number : 21522312

Yogyakarta, May 5th, 2025

Supervisor,



Ir. Muhammad Ridwan Andi Purnomo, S.T., M.Sc., Ph.D., IPM.

EXAMINERS APPROVAL PAGE
IMPLEMENTATION OF RANDOM FOREST REGRESSION WITH
HYPERPARAMETER TUNING AND SHAP ANALYSIS TO PREDICT PULL TEST
RESULT AT SOFT LAMINATING MACHINE

UNDERGRADUATE THESIS

Written by:

Name : Ar Royyan Utama T
Student Number : 21522312

Has been defended before the board of examiners in partial fulfillment of the requirement
for a Bachelor Degree in Teknik Industri at the Faculty of Industrial Technology
Universitas Islam Indonesia

Yogyakarta, May 9th 2025

Ir. Muhammad Ridwan Andi Purnomo, S.T., M.Sc., Ph.D., IPM.

Chairman

Dr. Drs. Imam Djati Widodo, M.Eng.Sc.

Member I

Ir. Bambang Suratno, S.T., M.T., Ph.D

Member II

Acknowledged by,

Head of Undergraduate Program in Industrial Engineering
Faculty of Industrial Technology
Universitas Islam Indonesia

Ir. Muhammad Ridwan Andi Purnomo S.T., M.Sc., Ph. D., IPM
NIK. 015220101

DEDICATION PAGE

This undergraduate thesis is dedicated to myself. I want to thank everyone involved in the making of this, especially those who are spending energy, time, and resources to help the author finish the writing. Every support has been priceless and may Allah grant with Jannah. Special shout-out to my parents, who didn't stop asking about my thesis, which boosted my spirit to finish it.

MOTTO

*Bismillahilladzi laa yadhurru ma'asmihi syai'un fil ardhi wa laa fissamaa'i, wa huwas samii'ul
'aliim*

*“In the Name of Allah with Whose Name there is protection against every kind of harm in the
earth or in the heaven, and he is the All-Hearing and All-Knowing.”*

-Sunan Abu Dawud 5088; at-Tirmidhi 3388-

PREFACE

Bismillah

All praise and gratitude to Allah *Subhanahu wa Ta'ala*, which given all His Rahmat so that the author can finish the undergraduate thesis titled **“Implementation of Random Forest Regression with Hyperparameter Tuning and Shap Analysis to Predict Pull Test Result at Soft Laminating Machine”**. The purpose of making this undergraduate thesis is to fulfill the requirement for achieving a bachelor’s degree in Industrial Engineering. The author would like to express gratitude to everyone who spent time, energy, and resources in supporting the thesis can be done.

1. My parents, which never stop believing and keep supporting me. Everything that you’ve given to me as in guidance and support have become something the author is very grateful.
2. Mr. Ir. Muhammad Ridwan Andi Purnomo, S.T., M.Sc., Ph.D., IPM, the head of the Industrial Engineering department, and also the author’s supervisor.
3. Mr. Agus Mansur, my pre-defense examiner, gave the author a lot of feedback and suggestions in the writing process of the thesis.
4. Every cool lecturer Author have met and influence on Author’s knowledge and mindset, so that this thesis can be written following the scientific principles.
5. Every amazing mind that the Author has met in the writing process and already helped the Author indirectly so that the writing can be, inshaallah, amazing too.
6. Everyone works in PT. XYZ, Pak Suhadak, Pak Asrori, Pak Yunan, Pak Ghafur, Pak Anam Saho, Pak Redi, Pak Dedi Irawan, Pak Rudi, Bu Lilik, Pak Arif, Pak Nanang, Bu Anggra, dan Bu Jeli.
7. Everyone who keeps supporting the author and is there when things do not go as expected, thank you for being there when the author needed it.
8. My classmates as comrades through laughter and down moments. The 4 years have been an unforgettable moment.
9. Everyone engages in the process, the authors cannot mention one by one.

The author acknowledges that this writing is far from perfect; thus, feedback and suggestions from anyone reading this are very welcome. Hopefully, this thesis can be useful to the science and those who are using science as a medium to be a cool person.

ABSTRACT

Efficiency is the key in the manufacturing industry. In this industry, manufacturers are competing to be the most efficient company and the one that spends the least. PT. XYZ is a company that operates in the audio industry, manufacturing speakers and audio peripherals. In the audio industry, the quality of speaker housing is exceptional. It will prove the quality of the product over a long period. The housing is usually made from MDF coated with PVC. This study tries to collect the historical data of adhesive strength produced in the manufacturing company and make a systematic predictive method that can approximate the relative strength of the product made. The result from the study shows that the method has an error rate of about 0.21 from real adhesive strength, suggesting an accurate prediction method for the company. The result can also be analyzed using SHAP analysis, which is also included as the output of the method.

Keywords: Manufacturing Industry, Prediction, Random Forest, SHAP Analysis.

TABLE OF CONTENTS

RESEARCH COMPLETION LETTER	ii
SUPERVISOR APPROVAL SHEET	iii
EXAMINERS APPROVAL PAGE	iv
DEDICATION PAGE	v
MOTTO	vi
PREFACE	vii
ABSTRACT	viii
TABLE OF CONTENTS	ix
LIST OF TABLES	x
LIST OF FIGURES	xi
CHAPTER 1 INTRODUCTION	1
1.1 Background	1
1.2 Research Objectives	5
1.3 Benefits of Research	5
1.4 Research Limitations	5
1.5 Systematic Research	5
CHAPTER II LITERATURE REVIEW	7
2.1 Literature Review	7
2.2 Theoretical Basis	13
CHAPTER III RESEARCH METHOD	22
3.1 Research Subject	22
3.2 Research Object	22
3.3 Data Analysis Method	22
3.4 Data Collection	22
3.5 Research Flow	22
CHAPTER IV DATA COLLECTING AND PROCESSING	26
4.1 Data collection	26
4.2 Pre-processing	28
4.3 Descriptive Analysis	29
4.4 Data Training and Data Testing	32
4.5 Hyperparameter Tuning	33
4.6 Random Forest Implementation	34
4.7 Prediction and Evaluation using SHAP Analysis	36
4.8 Model Evaluation	38
4.9 System Recommendations	39
CHAPTER V DISCUSSION	41
5.1 Random forest regression model	41
5.2 Hyperparameter tuning using the RandomizedSearchCV	42
5.3 SHAP Analysis method	43
CHAPTER VI CLOSING	45
6.1 Conclusions	45
6.2 Suggestions	46

LIST OF TABLES

Table 2.1 Research Gap	12
Table 2.3 Standards of product sampling based on some parties.....	19
Table 4.1 Variables used from the data	26
Table 4.2 Soft laminating machine setting data	27
Table 4.3 Pull test data	27
Table 4.4 Final data of Combined Data	28
Table 4.5 Data training and data testing division.....	33
Table 4.6 Parameters used for random search value	33
Table 4.7 Iterations and the result	34
Table 4.8 Variable importance	36
Table 4.9 Prediction example.....	37
Table 4.10 Prediction simulation on data testing	38

LIST OF FIGURES

Figure 1.1 Percentage and number of NG	3
Figure 1.2 Pull test result in quartals of 2024	3
Figure 2.2 Framework of the KDD	13
Figure 2.3 Types of machine learning.....	14
Figure 2.4 Decision tree classifier.....	15
Figure 2.5 Random Forest Tree.....	17
Figure 2.6 7 Tools	20
Figure 2.7 The PDCA method.....	21
Figure 3.2 Flowchart of the research along with the random forest flowchart	23
Figure 4.2 Descriptive analysis by building histogram visualization of variables towards the pull test result variable.....	29
Figure 4.3 Descriptive analysis by building boxplot visualization of variables towards the pull test result variable.....	30
Figure 4.4 OK and NG products distribution along the period from August 2023 to early December 2024	31
Figure 4.5 Random tree sample visualization	35
Figure 4. 6 Feature importance by the random forest model	35
Figure 4.7 The SHAP analysis on the feature value based on the built random forest regression model.....	36
Figure 4.8 SHAP Waterfall diagram on an example prediction	37
Figure 4.9 The actual pull test value VS predicted pull test value.....	38

CHAPTER 1

INTRODUCTION

1.1 Background

Manufacturing industries have been evolving significantly since the start of the 21st century. Over the past decade, the shift of manufacturers from conventional to digitization has been a trend in the area. There have been many vital improvements in the heart of manufacturing, such as promoting the paperless movement, using robots as more stable equipment, and using cloud-based servers as an integrated database. The urgency of companies to deliver and offer the best products in the market has intensified due to globalization and rapid technological improvements, and customers can compare quickly. To protect and ensure the deliveries of value of the product, quality control of a company is vital to a company.

According to Crosby (1979), in his book, *Quality is free* if you understand it; quality is the conformance to requirements. In *Project Management Book of Knowledge (PMBOK)* states that quality is the degree to which a product, service, or requirement. Quality control, as one of the most important instruments in the manufacturing industries, has been vital to companies. With the involvement of the digital era, the benefits of digitization of quality control have been studied by many. Research conducted by Baumer et al. (2016) shows that the technological improvements in the manufacturing system can increase machine productivity and reduce the cost of production. According to Shivajee et al. (2019), conversion of a manufacturer to quality control-based and real-time digitization can save the stated company from spending about 2.2 million USD, produce a cleaner production system, and contribute to the sustainability of the company. The significance of digital transformation is also stated in Shang & Zhang (2022), where the decreasing cost of manufacture can be obtained at 17.6% and 22.6% of revenue increases. To ensure the quality of products produced, a company must conduct continuous improvement in its quality control system. In line with previous studies, Mulugeta (2020) also stated that the effort in fixing wastes such as

lead time, high work-in-progress, poor line balancing, and material waste can lead the system to have better productivity, which increased by about 16.66%. In brief, companies have to put efforts to eliminate all wastes that contribute high to production costs. In a manufacturing industry, every machine is expected to be efficient and accurate. However, the result of a machine is not always satisfying the performance desire.

PT. XYZ is a company runs in sound and audio industry producing all kinds of speaker and loudspeaker to be distributed around the world. There are a lot of machines in PT. XYZ and one that still offers a big area to improve is the soft laminating machine. Soft laminating machine is a machine to process two materials to be laminated engaging the combination of heat, pressure, glue type, and material type. Study conducted by Büyüksari (2013) indicated that the heat, pressure, and different type of lamination could influence the result of hardness of the MDF layered. In PT. XYZ, soft laminating machine is used to laminate the medium-density fiberboard (MDF) with polyvinyl chloride (PVC) as the result of this process will be the main component of various body speakers. The soft laminating machine utilizes the combination of dry zone temperature, roll temperature, roll pressure, roll speed, and the glue weight to get the best result of laminated MDF. Moreover, the performance of the soft laminating machine will be evaluated through pull test.

Pull test is a measure to evaluate the machine setting and the test will be taken after seasoning the materials. The purpose of this test is to ensure the durability and the quality of how well the PVC attached to the MDF. This method can be a measurement to identify and determine how well the result and whether the product can be passed the quality control check or it needs to be rework again. Pull test also record the soft laminating machine setting in every product batch before the process began. While relying on the combination of machine setting to get the best pull test result, the work instruction (WI) of the machine still in the categorical form. There is no knowledge in PT. XYZ about what exactly a set of variable combination that can led the product to get the best result. Taking pull test after waiting for seasoning for several hours while the result of the pull test cannot be estimated is not good for the system. The product batch can be cancelled due to unsuccessful pull test.

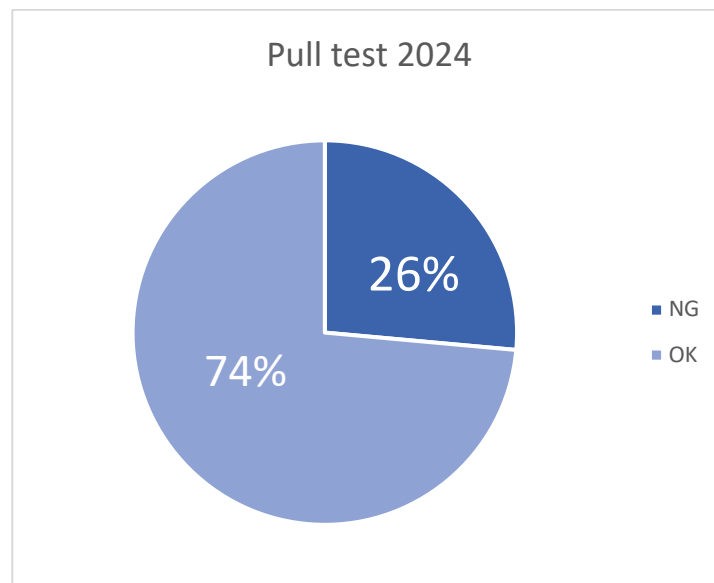


Figure 1.1 Percentage and number of NG

Source: Database PT. XYZ

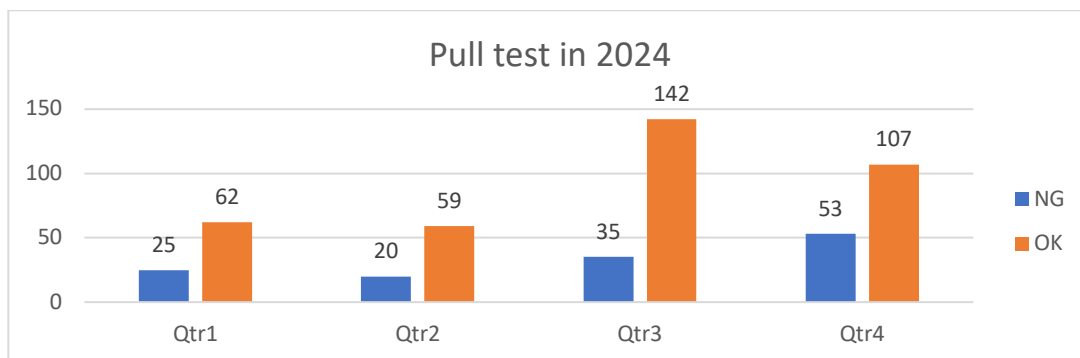


Figure 1.2 Pull test result in quartals of 2024

Figure 1.1 shows the number of the defects (NG) at soft laminating machine. It was a very high number but given rules are not strictly applied as if there is a product that was just slightly below 1, then the product will be accepted. But the needs of real and validated product with given threshold should be proposed. Therefore, the need of having knowledge can be very beneficial to the company. Based on the rough calculations, each data of NG session will cost the company about Rp10.691.250,00. The number was calculated considering the time wasted, manufacturing cost, and the product that cannot be repaired anymore. According to Figure 1.2, in 2024, the company has already loss about Rp1.421.936.250,00.

Random forest is a great algorithm to form a model that can predict outcome. Wu et al. (2017) conduct a comparative study in the performance of machine learning algorithms, the random forest is the most standout among artificial neural network (ANN) and support vector regression (SVR) where it generates a more accurate prediction. The performance of machine learning algorithms can also be increased using hyperparameter tuning. There are many research shows the impact of hyperparameter tuning to the machine learning algorithms. Hyperparameter tuning is a robust optimization technique to identify the best feasible value of hyperparameters for a machine learning model to get the best outcome of the model (Isabona et al., 2022). In conclusion, a hyperparameter tuned random forest is a robust, increased accuracy model to predict outcome based on the trained data to the model.

Random forest can be a suitable method to predict any outcomes, and it is not limited to linear problems; it can also handle non-linear relationships. Unlike a linear problem where variables' contributions can be analyzed for the global model, a non-linear problem cannot explain this issue. This is in line with the pull test problem, where each variable could affect the outcome in a different direction depending on how the other variables acted. In some cases, when the pull test result predicted by the model does not satisfy the desired result, the operator of the machine needs to see which variables influence the outcome. To address this issue, the Shapley additive explanations (SHAP) analysis method can be the solution. SHAP analysis supports various interpretations based on the influence of each input variable on the non-linear model (Kim & Kim, 2022).

Based on the explanation before, this research is proposed to develop a machine learning model using a random forest algorithm with hyperparameter-tuned features, and then the model will be analyzed using the SHAP analysis method. Hopefully, this research will be able to predict the outcome of certain variable inputs of the soft laminating machine, and the company could reduce the waste occurring in the pull test.

1.1 Problem Formulation

Based on the background of the research that has been explained, we can formulate some research questions:

1. How to have a prediction tool to predict the outcome of a pull test?
2. How is the performance and accuracy of the trained random forest model?

3. How is the influence of input variables on the pull test result?

1.2 Research Objectives

The research objectives answer research questions. The following are the research objectives:

1. To build a machine learning model that can predict the outcome of a pull test.
2. To know the performance and accuracy of the trained model.
3. To know the influence of input variables on the pull test result.

1.3 Benefits of Research

Benefits of this research are:

1. This research is very beneficial to the company because of the implemented method can transform the traditional method to an advanced one. The result can also motivate other area that have the same issue to implement this method.
2. The benefit for the student is that this research can train the student to be more agile in choosing the best method to apply in the field.

1.4 Research Limitations

The following are the limitations of the research:

1. The data used in this thesis research is secondary data obtained from the database of PT. XYZ. There are two kinds of data retrieved, the first one is the machine setting data, second is the pull test result data. Data retrieved from August 2023 to November 2024.
2. The statistical methods used in this research are random forest, hyperparameter tuning, and the SHAP method.
3. The software to support the research is Python with the environment of Google Colaboratory and Microsoft Excel.
4. The object of the research is the soft laminating machine in wood working area, production department, PT. XYZ.

1.5 Systematic Research

The research is structured into several chapters, and each chapter will be described as follows:

CHAPTER 1 INTRODUCTION

This section describes the discussion in Chapter I consist of background, problem formulation, research objectives, benefits of research, and research limitations.

CHAPTER II LITERATURE REVIEW

This section contains the inductive and deductive literature.

CHAPTER III RESEARCH METHODOLOGY

This section elaborates on the methodology, object, subject, and research flow.

CHAPTER IV DATA PROCESSING AND RESEARCH RESULTS

This section shows the result of the research.

CHAPTER V DISCUSSION

This section describes the challenges, obstacles, benefits, and performance of the research.

CHAPTER VI CONCLUSIONS AND RECOMMENDATIONS

This section contains the conclusion and recommendations of the research based on the obtained results.

BIBLIOGRPHY

ATTACHMENT

APPENDIX

CHAPTER II

LITERATURE REVIEW

2.1 Literature Review

Literature review is a section to explain and to correlate previous and recent research related to this research. The purpose is to find the gap between studies that have been conducted, and obtain recent knowledge to conduct similar research.

Random forest with the combination of SHAP analysis has been used for a very wide area. Study conducted by Wang et al. (2021) emphasizes the use of random forest in spatial heterogeneity modeling of water quality in China. The low number of physical monitoring stations has slowed the examination of spatial water quality distribution, and the identification of factors that affect water quality has been the motivation of the research. By developing the random forest regression (RFR) model to model the distribution of water quality in Taihu Lake basin, Zhejiang, China, Wang et al. hope that they can interpret the result along with the SHAP method analysis as well. The study uses 3 parameters, which are the permanganate index (COD_{Mn}), total phosphorus (TP), and total nitrogen (TN), from 16 watershed features. The result of the research shows that the COD_{Mn} results were consistent among the datasets, where TN and TP have slight differences based on the spatial distribution. Based on SHAP analysis, the TN feature was generally affected by the agricultural non-point sources, while COD_{Mn} TP was influenced by the domestic resource and agricultural sources. In conclusion, the research obtained knowledge through the modelling of RFR with SHAP interpretation. The SHAP analysis helps explaining on the influence of features on the parameter.

Another research by Antosz et al. (2022) provided results in implementing random forest, in the manufacturing field. Sustainability was the main purpose, which was implemented through eliminating waste and reducing the use of materials. The objective of the research is to identify parameters that influencing the state of the cutting blade. The model using random forest that was developed and evaluated using the quality of classification. The obtained results enlighten the quality of the model, as the proposed model can be used as a decision-support system in determining the cutting tool's life. This has implications for reducing a material. Similar object of

research by Qureshi et al. (2020), the study targets sustainable manufacturing. Findings of this research indicate that practices on the environmental assessment were the most work in the earlier era rather than providing a holistic solution for industrial engineering.

Alhakeem et al. (2022) have conducted research on making a model that can predict eco-friendly concrete compressive strength. This prediction model uses a gradient boosting regression tree with GridSearchCV hyperparameter as the model method. To build the desired model, 164 experimental data points on the object research were obtained. The dataset included water per binder ratio (W/B), curing time (age), the recycled aggregate percentage from the total aggregate in the mixture (RA%), ground granulated blast-furnace slag (GGBFS) material percentage (GGBFS%), and the superplasticizer (kg). The result of the GBRT model can be boosted using the GridSearchCV in this research. With the root mean squared error (RMSE) and R^2 used as the evaluation parameters of the model, values of 2.3212 and 0.9612 are obtained, respectively. As the final approach, the research uses the SHAP approach to explain the influence on the compressive strength.

The use of parameter tuning is also used in another field of machine learning, like time-series forecasting. Research conducted by Bacanin et al. (2023) shows the significance of parameter tuning in the time-series forecasting challenge. In this research, the number of prediction models is consistently rising. The model developed in this study used the long short-term memory (LSTM) deep learning model, tuned for multivariate time-series forecasting. The result shows the significance of parameter tuning using the metaheuristics rather than the grid search, where it obtained the values of R^2 of 0.9515 and 0.9136, respectively. This study shows another applicable parameter tuning method using the metaheuristic one, which results in higher R^2 values compared to the grid search.

A comprehensive comparative analysis of the implementation of hyperparameter tuning on machine learning was conducted by Elgeldawi et al. (2021). The research comprises the performance of each hyperparameter tuning technique and the kinds of machine learning models in solving the Arabic sentiment classification problem. Sentiment analysis is a method to see whether a text contains positive, negative, or neutral sentiment. However, it is explained that it is not an easy task to conduct such an analysis. Despite that, all classifiers' performance is evaluated using our created dataset both before and after the hyperparameter tweaking process. The result of

the study shows that the highest score was obtained by the support vector machine (SVC) hyperparameter tuned with the Bayesian optimization, with a score of 95.6208.

The significance of using the SHAP analysis method was noted by Antonini et al. (2024), where it was implemented in a random forest classification model. El Fierro intrusive body is part of the La Jovita-Las Aguilas mafic-ultramafic belt, which is found in Argentina's Sierra Grande de San Luis. This belt's units contain base metal sulphide (BMS) mineralization as well as platinum group minerals. The macroscopic description of mafic and ultramafic rocks, as often performed by mining exploration corporations, results in an imperfect modal categorization of the rocks. The object of the research is to predict the mafic and ultramafic rocks intercepted by drill cores by the geochemical parameters. Random forest, which was used as the classifier algorithm model, obtained about 86% to 94% of accuracy and an F1_score of 96%. As the model becomes really difficult, the need to use an advanced interpretation method is urgent.

In research by Li (2022), they stated that machine learning and artificial intelligence, previously considered as "black box" techniques, are becoming more interpretable as the benefits of recent advances in interpreting results. One important breakthrough is the creation of local interpretation methods such as the Shapley Additive Explanation (SHAP), which gives a reliable framework for modelling, interpreting, and visualizing non-linear problems. For example, SHAP has been used to evaluate machine learning models such as extreme gradient boosting (XGBoost), allowing researchers to gain a better understanding of these models. Comparative simulation studies have shown that SHAP-explained XGBoost may generate geographic impact estimates comparable to classic spatial statistical models like the geographic Lag Model (SLM) and Multi-scale Geographically Weighted Regression (MGWR). Furthermore, in an actual case study of ride-hailing services in Chicago, SHAP successfully showed complicated spatial and non-spatial effects, such as nonlinearities and interactions. These findings demonstrate the strength of locally interpretable machine learning models as alternatives to spatial statistical approaches, particularly when dealing with complex and previously unknown spatial interactions.

In the effort continuous customer satisfaction for the business stability, Kiangala & Wang (2021) in their research, try to construct an effective adaptive customisation platform that translates the customization data history of a small manufacturing plant from a static database into a dynamic machine learning model to accurately generate individualized items for their clients. The research

has a purpose to improve the purpose of improving customer satisfaction by reducing the complexity of the customization system, which involves several parameters to obtain the desired personalized products in a single entry. Powerful machine learning algorithms were used in this research, such as extreme gradient boosting (XGBoost) and random forest (RF) ensemble learning, to match one customer's input to the desired customized category of product. The result of the study shows the domination of XGBoost over RF as the best machine learning method for the regression problem with non-linear data.

An application of machine learning in the medical field was noted by Lundberg et al. (2020). In this research, Lundberg et al. highlight the use of game theory to compute optimal explanations based on a polynomial-time algorithm. They also mention that these tools can directly measure the influence of a feature in a nonlinear problem. Lastly, they also stated that this tool could help in understanding the global structure (not only locals) by gathering information from local explanations. The problem that will be further analysed in this research is the mortality risk factors in the US. By combining the use of machine learning and SHAP analysis in this research, it is obtained that these tools can identify high-probability but low-frequency non-linear risk factors of mortality in the US. Furthermore, these tools can help in highlighting the distinct subgroups of the population with shared risk characteristics. Another benefit is that the interaction effects between non-linear risk factors for chronic kidney disease. Lastly, tools can help monitor the machine learning model running in the hospital by analyzing the features that degrade the performance of the model over time.

Another case of the use of machine learning with tuning and advanced non-linear influence interpretation was held by Isabona et al. (2022). The object of the research was to provide an extrapolative analysis of user communication distances to the gNodeB transmitter in 5G new radio networks by modelling the adaptive learning based on machine learning regression models. The next objective of this research is to experiment with the combined random forest (RF), least-squares (LS), and the Bayesian hyperparameter tuning to get the best possible modelled algorithms. Therefore, the name of the model was named RF-LS-BPT method. It is hoped from the research that LS and BPT can boost the performance of the RF model through their own algorithms. The result shows the high accuracy of prediction obtained from the model compared to the traditional RF model. It was obtained that the coefficients of determination (R^2) and mean absolute error

(MAE) for the throughput prediction user at different locations vary from 0.9800 to 0.9999 and 0.42 to 4.24, respectively. As a comparison of the method, a traditional RF model's performance is 0.9644 to 0.9944 of R^2 and 5.47 to 12.56 of MAE values. This shows the significant performance by applying the LS and BPT methods.

Now getting to another topic area, Kim & Kim (2022) bring a climate change discussion. The background tells that the heat increase has worsened around the world. As the main victims of this the humans and all living creatures have led to heat-related mortality. To tackle this issue, the research is proposing a method that can predict heatwaves in different areas with different geographical characteristics. A random forest (RF) model was developed to estimate the occurrence the heat-related mortality in a spatial unit in a city. By hyperparameter optimization, the model obtained accuracy, F1-score, and AUC values of 90.3%, 94.75%, and 86%, respectively. The model was then interpreted from local (within a city) to global perspectives using the Shapley additive explanations (SHAP). As a result of the interpretation, it was found that demographic, socioeconomic, and climatic sectors were the most influential on the estimation results.

Similar to the research conducted by Alhakeem et al. (2022), Ekanayake et al.'s (2022) research was about the compressive strengths. It is obtained from the study that the best machine learning method applied in this research is the Laplacian kernel ridge regression (LKRR) and the extreme gradient boosting (XGBoost) as the best performing methods, with the score of R of 0.98. There are other machine learning methods tested in this study, such as the decision tree, extra tree, adaptive boost (AdaBoost), and light gradient boosting method (LGBM). According to the SHAP interpretation, the model can capture complex relationships. An interesting fact is that the SHAP method results are in line with what is mainly observed in the behavior of the compressive concrete.

Machine learning for SMEs was conducted by Amaliah et al. (2022) for predicting the variant of coffee in a coffee shop. This research has the objective of predicting the most wanted coffee variant by the customers. From the analysis of the research, it was found that the least error that the model can get was by setting the features $mtry$ 2 and $ntree$ 500. The performance of the research was based on the confusion matrix, and the accuracy obtained was 94.12%.

A comparative analysis conducted by Çinar et al. (2020) found that the trend of machine learning applications to shifting the manufacturing industry to the digital era. The objects of the research are how manufacturing adopts machine learning, artificial intelligence, and predictive

maintenance. From the research, it can be concluded that the use of machine learning in the industrial field has increased significantly over the years, with the most common commercial platforms for machine learning varying. Platforms like TensorFlow, IBM Watson Studio, RapidMiner, etc. Another finding is that the wide use of linear regression, logistic regression, random forest, decision tree, and other machine learning methods is being applied widely. This is a positive trend for the manufacturing area's digital transformation.

Machine learning application in manufacturing has a lot of impact. The method can be applied for scheduling, monitoring, quality, and failure prediction. Dogan & Birant (2021) shows the use of machine learning application can be in almost area. The research was conducted in order to achieve a better understanding of the main approaches and algorithms used to improve the manufacturing field in the past two decades. Using accuracy and mean absolute error, the research tries to see which the most machine learning algorithms perform the best. Different types of machine learning in supervised (decision tree, etc), unsupervised (association rules, etc), ensemble learning, and deep learning were used to get a better performance in the manufacturing industry.

Table 2.1 Research Gap

Author	Machine result prediction	Manufacturing sustainability	SHAP	Hyperparameter tuning	Random Forest Regression
Wang et al. (2021)			v		v
Antosz et al. (2022)	v	v			v
Alhakeem et al. (2022)			v	v	v
Bacanin et al. (2023)	v			v	
Elgeldawi et al. (2021)				v	
Antonini et al. (2024)			v		v
Li (2022)			v		
Kiangala & Wang (2021)	v	v			v
Lundberg et al. (2020)			v		v
Isabona et al. (2022)				v	v
Kim & Kim (2022)		v	v		v
Ekanayake et al. (2022)	v			v	v
Amaliah et al. (2022)		v		v	v
Çinar et al. (2020)		v			v
Dogan & Birant (2021)	v	v			
Proposed	v	v	v	v	v

2.2 Theoretical Basis

2.1.1 Knowledge Discovery in Database (KDD)

Knowledge Discovery in Database (KDD) is a series of non-trivial process of extracting implicit, previously undiscovered, and possibly valuable information from data (Frawley et al., 1992). It refers to a term used to explain the process of obtaining information in a dataset. KDD familiarly known as data mining which combines statistical methodology in any field like big data, machine learning, and visualization (Zanin et al., 2016). The ultimate purpose of the KDD is to find and seek information which coming from the pattern identification where each method has different algorithms.

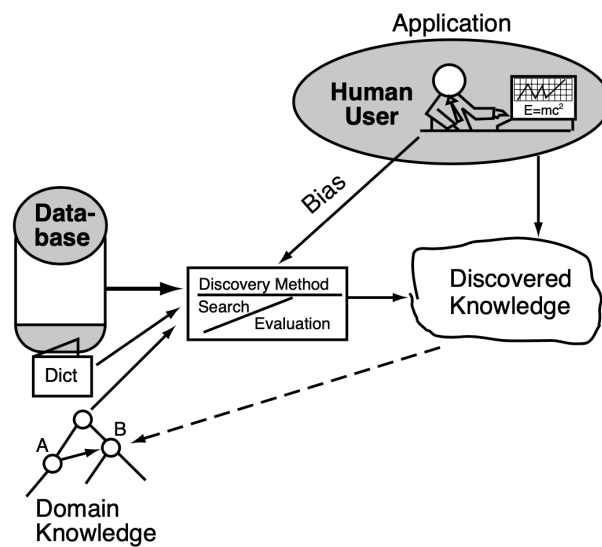


Figure 2.1 Framework of the KDD
Source: (Frawley et al., 1992)

2.1.2 Machine Learning

According to Dietterich (1990), the definition of learning in machine learning is increase in knowledge. Furthermore, there are two conditions where learning can occur. First, the agent was told a new fact that it didn't know. Second, the agent conducts an 'inductive leap' and chooses to believe the facts. Machine learning is a concept where it can learn by its own without being directly programmed (Samuel, 1959). There are a lot of application of the machine learning concept, packaged in form of algorithms like clustering, classification, and association rules.

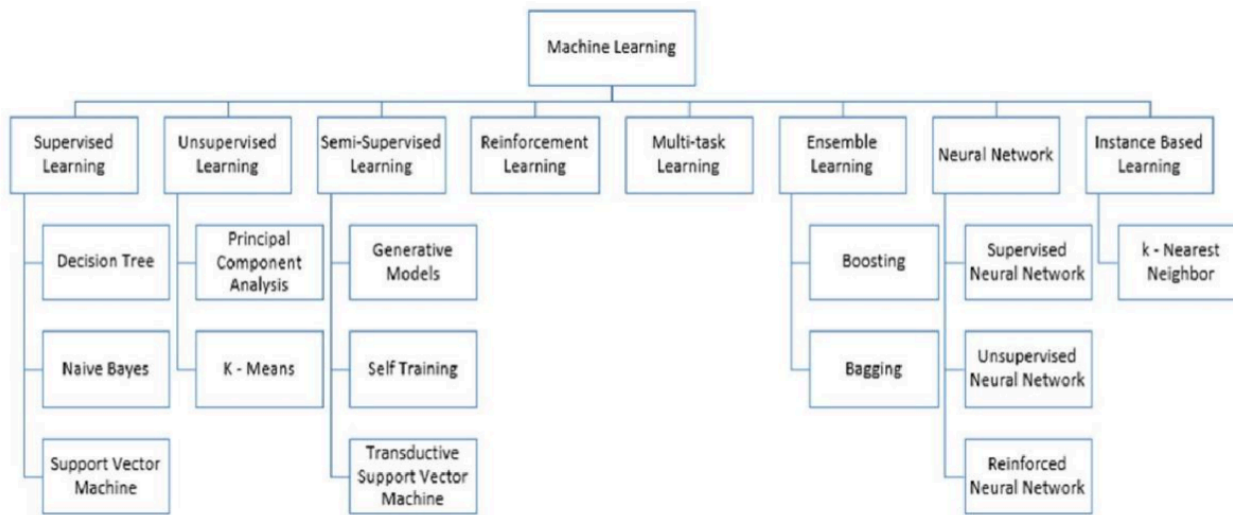


Figure 2.2 Types of machine learning

Source: Mahesh (2020)

In machine learning, each way of learning has its own branches. The development throughout years makes it a giant structure because of the updated knowledge over the years. From supervised learning, there are decision tree which very famous, naïve bayes, and support vector machine. For the unsupervised learning there are principal component analysis and k-means. For the ensemble learning there are boosting and bagging methods. And one of the most applied one, neural network.

1. Supervised Learning

Supervised learning is a method to let a dataset to have data training and data testing (Cunningham et al., 2008). In the same research, Cunningham et al. (2008), implies that supervised learning learns by finding pattern between a set of input variables x and output variable y and apply this pattern to predict the outputs. The example of supervised learning is the nearest neighbor classification.

2. Unsupervised Learning

According to Hastie et al. (2009), unsupervised learning learns without teacher. Without the assistance of a teacher or supervisor who can provide accurate responses or the degree of inaccuracy for each observation, the objective is to directly infer the characteristics of this probability density. The example of unsupervised learning method is the association rules, where the use is widely use for the market basket analysis.

2.1.3 Classification and Regression Tree

A decision tree is a recursive categorization technique that divides a data set into smaller subsets based on stated tests at each node (Friedl & Brodleyf, 1997). The purpose of the classification process is to predict accurately the category in a dataset which have not yet been known. Regression trees are used for dependent variables with continuous or discrete values, whereby the prediction error is commonly calculated as the squared difference between observed and forecasted values (Loh, 2011).

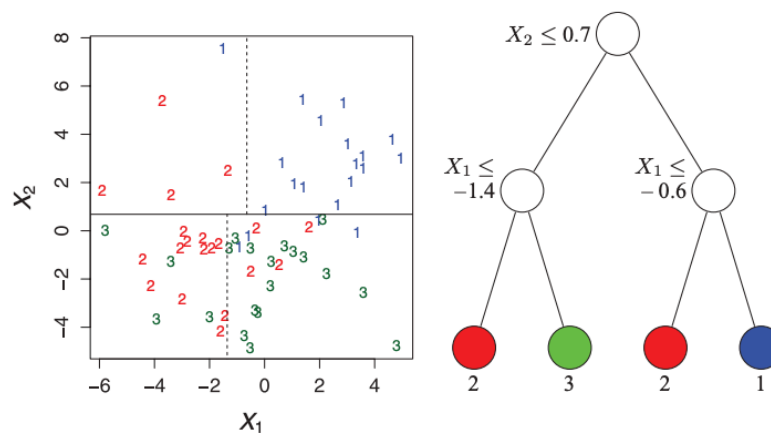


Figure 2.3 Decision tree classifier

Source: (Loh, 2011)

There are several ways to construct the classification and regression tree, it engages the use of gini index and other parameters. The construction steps are as follows:

1. Start by the root node.
2. For ordered variable X , make a conversion of unordered X' by making a subset by its values in the node into a small number of intervals. If X unordered, then $X' = X$
3. Perform chi square test of independence for each X' variable to Y for the data in the node and compute its significance probability.
4. Let X^* associated with X' that contains the smallest significance probability.
5. Identify split set $\{X^* \in S^*\}$ that minimizes the sum of Gini indexes and make it to split the node to two child nodes.

6. If a stopping criterion occurs, exit. Else, repeat 2 to 5 steps for each child nodes.
7. Prune the tree by the CART method.

In classification decision tree, the performance of each node is quantified through several numbers. Information Gain measures how successfully an attribute of attributes categorizes training data (Guleria et al., 2014). Continuing their explanation, entropy is the parameter that best categorizes data (H). A useful indicator of the information contained by a collection of events is entropy. From set S, there are two states: positive and negative instances. $H(S)$ represents the entropy of set S.

If S is the sample of n training events and P_i is the event's probability of occurring, then entropy may be found using:

$$H(S) = \sum_{i=1}^n -P_i \log_2 P_i \quad (2.1)$$

Such that: $\sum_{i=1}^n P_i = 1$

Where: S = Dataset
 C = Number of classes
 P_i = Probability of n^{th} class frequency in the dataset

At the end, attribute with the highest information gain will be the root of the decision tree of tree. The calculation of information gain will be as follows:

$$Gain(S, A_i) = H(S) - \sum_{i=1}^k \frac{|S_i|}{|S|} \times H(S) \quad (2.2)$$

Where: S = Dataset
 A = Attribute
 $|S_i|$ = Number of samples in i
 $|S|$ = Number of all samples

2.1.4 Random Forest

Random forests are a set of tree predictors, in which each tree is determined by the values of a random vector selected separately and uniformly across the forest (Breiman, 2001). As the number of trees in a forest increases, the generalization error approaches a limit. The generalization error of a forest of tree classifiers is determined by the strength of each tree in the forest and their

correlation. Random forest is one of the most famous ensemble methods that is developed by Breiman (2001). It is suitable for classification which contained with categorical output, and regression which used numbers as the output.

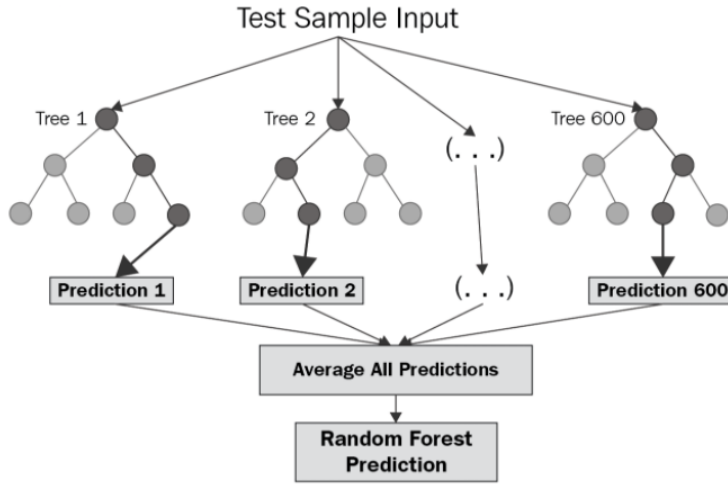


Figure 2.4 Random Forest Tree

Source: <https://levelup.gitconnected.com/>

There are several steps that are needed to be conducted:

1. Initially, the random subset with k data points from training dataset will be pickup with replacement. This step familiarly called bootstrapping.
2. Then the decision tree from the data subset will be build based on the k data points.
3. Choose the number N of trees that will be built then repeat steps 1 and 2
4. For a new data point, have each of your N -tree trees estimate the value of y for the data point in question, and then assign the new data point to the average of all of the anticipated values.

Random forest use Gini index to choose features in every internal node from the decision tree (Amaliah et al., 2022). Gini index will be calculated as follows:

$$Gini(S_i) = 1 - \sum_{i=0}^{c-1} P_i^2 \quad (2.3)$$

where P_i is the relative frequency of C_i in the dataset.

The quality of split in k features in S_i is the number of C_i samples and will be calculated as the sum of the probability of Gini index. It can be calculated as follows:

$$Gini_{split} = \sum_{i=0}^{k-1} \left(\frac{n_i}{n}\right) Gini(S_i) \quad (2.4)$$

where n_i is the numer of samples in S_i after the split, and n is the total sample.

The theory of the random forest is explained in Breiman (2001). Let a function $\{h(x, \Theta_k), k = 1, \dots\}$ where Θ_k is a random vector which independent identically distributed and let each trees choose the class from the majority vote. For random forest, two factors that gauge the strength of individual classifiers and their interdependencies can be used to create an upper bound on generalization error.

$$mr(X, Y) = P_{\Theta}(h(X, \Theta) = Y) - \max_{j \neq Y} P_{\Theta}(h(X, \Theta) = j) \quad (2.5)$$

and the strength of the set $\{h(x, \Theta_k)\}$ classifiers is

$$s = E_{X,Y} mr(X, Y) \quad (2.6)$$

With the assumption of $s \geq 0$, Chebychev's inequality and the derivation of mr from the margin function random forest

$$PE \leq \frac{\bar{\rho}(1 - s^2)}{s^2} \quad (2.7)$$

where $\bar{\rho}$ is the correlation average of

$$\bar{\rho} = \frac{E_{\Theta, \Theta'}(\rho(\Theta, \Theta')sd(\Theta)sd(\Theta'))}{E_{\Theta, \Theta'}(sd(\Theta)sd(\Theta'))} \quad (2.8)$$

2.1.5 Hyperparameter Tuning (Boosting)

Hyperparameter tuning is a process of selecting the best feature values for a machine learning model hyperparameter. It allows user to tweak the model performance by finding the best feature values for optimal results. In the early years of development, hyperparameter tuning has traditionally been a manual task because of the limited number of iterations, and often carried out by hand (Bardenet et al., 2013). Brendel & Schoenauer (2011) proposed a neural network that is used to train an evolutionary method that maps problems to hyperparameters. This option may be useful if the issue descriptions can reliably determine the best hyperparameter. Hyperparameter tuning used two kinds of searching namely randomizing the number of finding optimal values in interval defined, and grid searching which finding all possible combination of features within the limitation values set.

2.1.6 Quality Control

Quality is the conformance to requirements (Crosby, 1979). Quality control is a wide term that encompasses a lot of controlling the quality of products. One of the most traditional ways to do quality control is by taking the product sample of a product batch, testing it, and seeing whether there are any defects. There are many rules set by different parties for the acceptance number of product sampling in a product batch to be tested. Party like the International Standard Operating Procedures also plays a role in determining the standard rules of product sampling in a company. The number of acceptance criteria is crucial. It can be the critical factors that determine the success of a product lot. The following are the acceptance numbers of defects in product sampling based on some parties:

Table 2.2 Standards of product sampling based on some parties

Standards	Lot Size	Sample Size	Acceptance Number
ISO 2859	1201 - 3200	125	2 or fewer defects
ANSI/ASQ Z1.4	10001 - 35000	315	5 or fewer defects
MIL-STD-105E	1201 - 3200	125	7 or fewer defects

One of the most developing quality control methods in the industry is the ability to predict the quality of products. Predicting product quality would enable a producer to choose system characteristics more wisely throughout the early stages of design, increasing competitiveness by reaching higher quality levels (Nada et al., 2006). Furthermore, Nada et al. explain that the research conducted has concluded that the efforts should be directed to the prevention of errors in the process, so that the quality can be maximized. Despite taking samples to test the quality of a product lot becoming a very decent thing in the quality control process, controlling the quality with data analytics is popular lately. According to Sokovic et al. (2009), seven tools of quality were first proposed by Ishikawa (in the 1960s), which one of the quality management gurus. These tools are:

1. Flow chart
2. Pareto diagram
3. Check sheet
4. Control chart
5. Histogram
6. Scatter plot

7. Cause and effect diagram

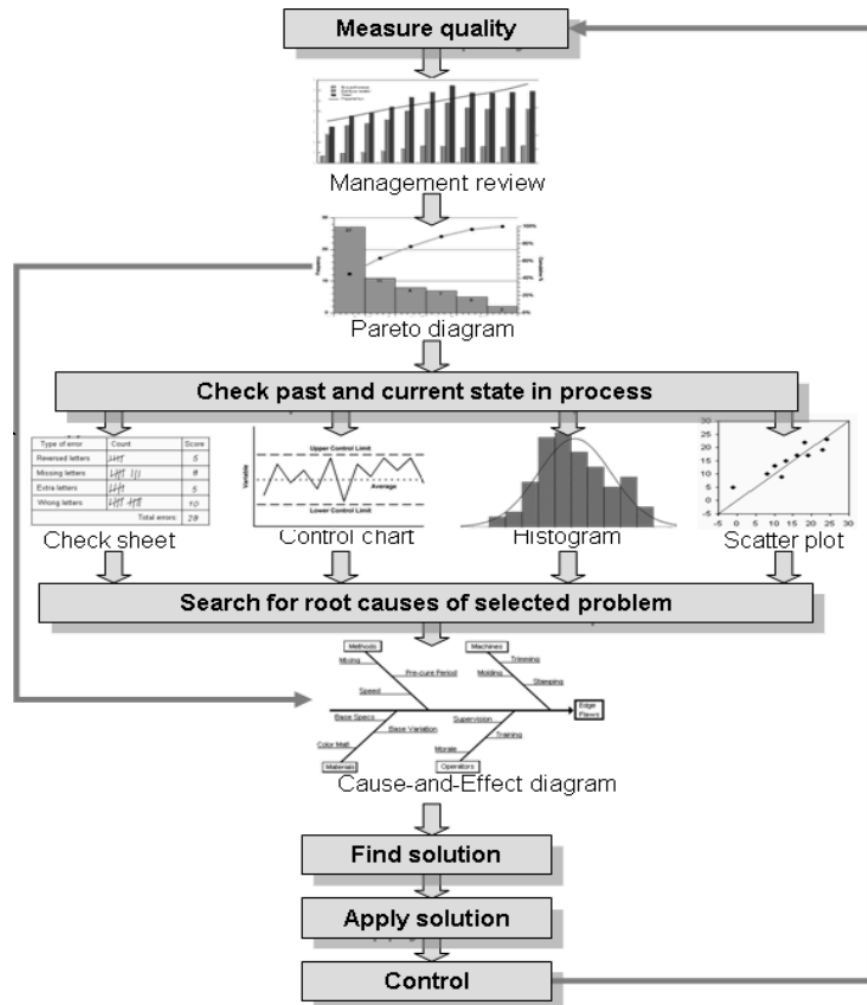


Figure 2.5 7 Tools

Source: Sokovic et al. (2009)

In order to tackle the problem captured from the analysis, 7 tools are frequently aligned with the Plan Do Check Action (PDCA) method. The use of PDCA-cycle is an effort for continuous improvement. The method engages improvement by starting with a careful planning, effective action, and repeat to obtain the maximum benefits.

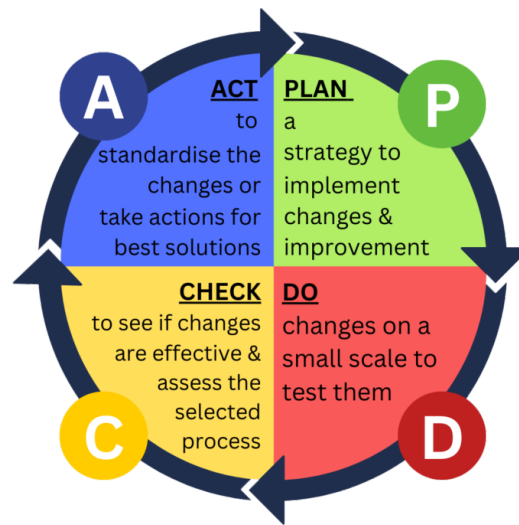


Figure 2.6 The PDCA method

Source: <https://sixsigma.com.my/>

CHAPTER III

RESEARCH METHOD

3.1 Research Subject

Research was conducted in a speaker manufacturing industry, namely PT. XYZ. In PT. XYZ, the subject of research, is at the soft laminating machine, wood working (WW) department, under the division of production.

3.2 Research Object

The research object is the pull test result conducted after producing a product batch and the machine setting, which is taken almost every day due to the needs of quality control for the product result. A pull test is an action taken to ensure the quality of the product in the soft laminating machine by pulling the PVC glued to MDF.

3.3 Data Analysis Method

In this research, there are some data analysis methods are used. For the model, the Random Forest Regression (RFR) model and boosted with hyperparameter tuning using with RandomizedSearchCV method were employed. And the result of the prediction will be interpreted using the Shapley Additive exPlanation (SHAP) analysis method.

3.4 Data Collection

The data needed for this research are the soft laminating machine settings and the pull test results. Both datasets are obtained from the database of PT. XYZ. Thus, secondary data was gathered for this research. However, in testing and implementing the model, primary data was gathered to check the reliability of the model. Secondary data gathered for was gathered from August 2023 to November 2024, while primary data was gathered for two weeks from December 9th to December 20th of 2024.

3.5 Research Flow

The flow of the research are as follows:

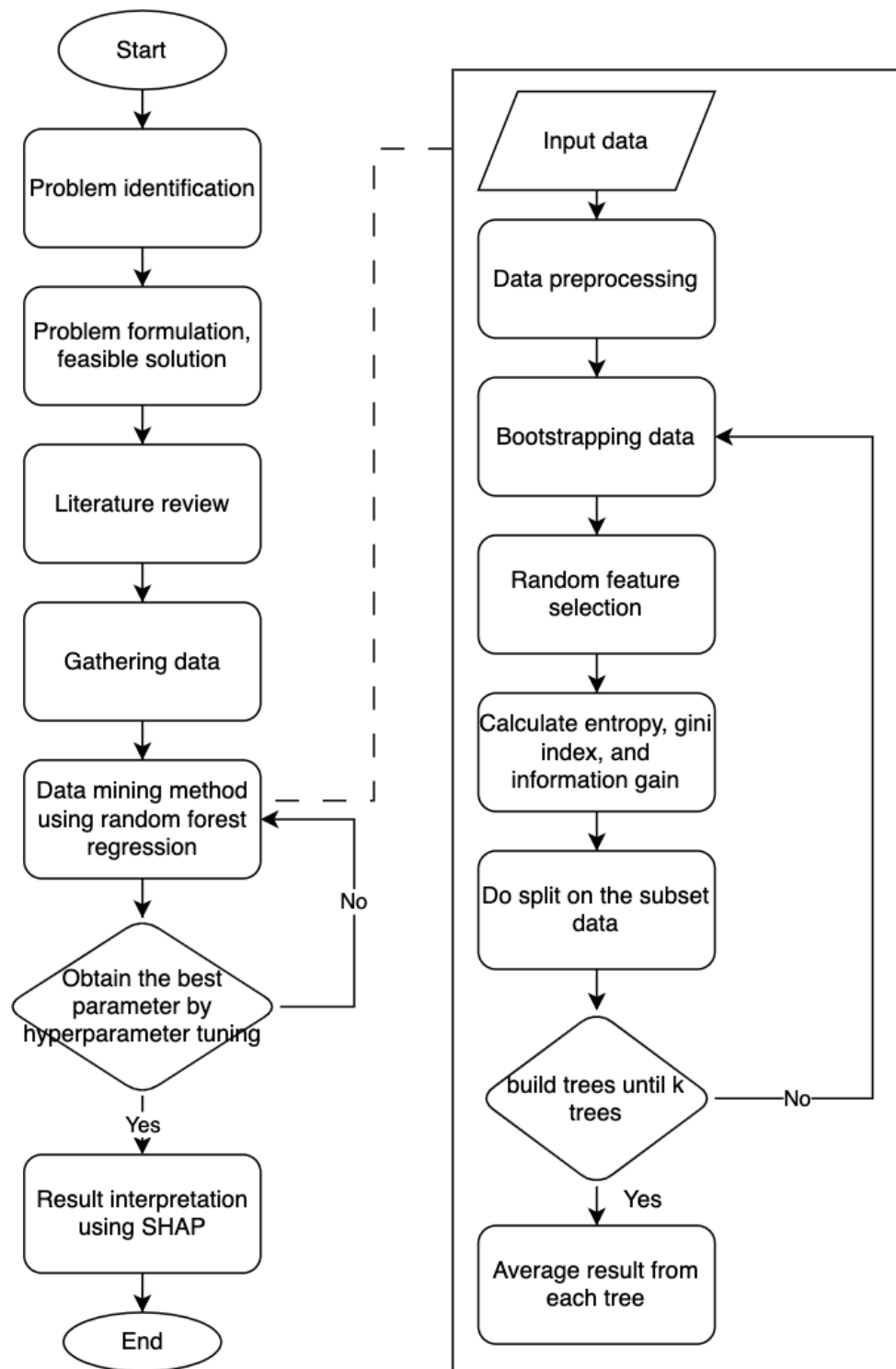


Figure 3.1 Flowchart of the research along with the random forest flowchart

The description of each flow in the flowchart:

1. Start

This point where the research was started.

2. Problem identification

At this point, there is some issue with the waste occurred from the soft laminating machine. After doing some discussion with the foreman and manager of the wood working department, it was obtained that improvement needed in the area. Report from the operator was also obtained that they need to know the output result of the machine when inputting certain values of variables.

3. Problem formulation, feasible solution

After identifying the problem, researcher was trying to find any idea about how might to address the issue. Several solutions with no support and background were on researcher minds. One idea that is set is to conduct a machine learning modelling and apply it so that it can predict the outcome of the result.

4. Literature review

At this point, researcher needs an enlightenment to address similar issues. Some research conducted and published have similar issues but one research was different at the object, while other research was different at the way to address the issue. At the end, the random forest method was set as the method of predicting the pull test result.

5. Gathering data

After finding and ensuring the way to resolve the method, the data needed to conduct research and run the method were asked to employee of PT. XYZ. The data was obtained through the database of PT. XYZ.

6. Data mining method applied using random forest

The data then processed through a data mining method, and from here a model that can predict the outcome was obtained. Several processes in this method were applied:

- a. Input data

The data was inputted to the python environment. Data that were inputted were two, which are the soft laminating machine setting and the pull test result.

- b. Data preprocessing

Data first need to be preprocessed through several actions like cleaning noise data, uncomplete data, and to transform some of the data so the data can be in one format.

c. Bootstrapping data

After the data was ready to be applied with the random forest algorithm, the data then will be divided into several subset. This process named bootstrapping data.

d. Random feature selection

After getting several subset of data, then random feature from the dataset were selected as variables.

e. Calculate the entropy, gini index, and infomartion gain

After building trees, the attributes of the tree need to be calculated.

f. Do split on the subset of data

Then do splitting based on the attributes of each tree.

g. Build trees until k trees

Decide whether the number of tree are already satisfied or not.

h. Average result from each tree

At last, prediction result from each tree then averaged.

7. Obtain the best parameter by hyperparameter tuning

After building a random forest model, then try to run the hyperparameter tuning method. This method will try to search the best scenario of number tree, random state, and number of jobs. Then the best model with tuned parameter obtained.

8. Interpret the result using SHAP

The result of the model will be interpret then using the SHAP. In here, the model can be analyzed and see the influencing variables, contributed to the result of the pull test.

CHAPTER IV

DATA COLLECTING AND PROCESSING

4.1 Data collection

The data collected for this research were gathered from the database of PT. XYZ. There are two types of data obtained. First is the data for the soft laminating machine setting, which was uploaded to the database by the operator. The second one is the data for the pull test of the product. Every data type has specific variables as shown in Table 4.1.

Table 4.1 Variables used from the data

Data	Variables
Soft laminating machine setting data	1. MDF
	2. Roll Pressure
	3. Roll speed
	4. Roll temperature
	5. Glue weight
	6. Dry zone temperature
Pull test data	1. Power 1
	2. Power 2
	3. Power 3
	4. Power a
	5. Power b
	6. Power c

The variables used for the soft laminating machine setting were chosen by the judgment of the operator and foreman of the related department. For the pull test data, the value of each power is then averaged, and it will be judged whether this represents the power desired for the product, or it will be classified as NG. Below are the highlights of each data point:

a. Soft laminating machine setting data

Table 4.2 Soft laminating machine setting data

[illegible]

b. Pull test data

Table 4.3 Pull test data

[illegible]

4.2 Pre-processing

In this research, the data that will be used was prepared so that it can be processed using the desired method. The pre-processing phase consists of several steps:

1. Cleaning data noise

Several data from the database include noise that comes from human error, and for this reason, cleaning the data for the dataset is required.

2. Transforming data

Some of the values in the data weren't written in a uniform format. Some data on the roll temperature and dry zone temperature were recorded differently from the rest of the data in the variables. To address this, the data which are not follow the majority format will be uniformed. Transforming the pull test result data was also conducted by averaging the values from power a, power b, power c, power 1, power 2, and power 3 into the pull test result variable.

3. Merge data

Because the data was separated into two different datasets, the need to merge the setting machine and pull test result data is required. Data mergers were conducted following these rules:

- a. The data for the pull test will be classified using the variable of the model. This value of data will then be matched with data from the soft laminating machine setting from the model variable as well.
- b. Dates of the pull test data will also be considered, which needs to be at least seven days after production of the particular model.
- c. Any data that does not satisfy the rules will be considered as data noise and deleted.

As a result, the soft laminating machine setting data that were previously contained in 1885 data points, were reduced to 1161 amount of data points because some of the production batches did not pass the pull test phase.

Table 4.4 Final data of Combined Data

date	model	Pressure	Roll Temperature	Dry Zone Temperature	Glue Weight	Speed	MDF	Pull test result
02/01/24	NS-B210	7.66667	57	38	2.72	414	9	0.78
02/01/24	GA-15 II	13.6667	60	40	2.95	350	15	1.04

date	model	Pressure	Roll Temperature	Dry Zone Temperature	Glue Weight	Speed	MDF	Pull test result
03/01/24	L100	16.6667	60	40	2.85	300	18	1.25
03/01/24	NS-BP101	7.66667	59	34	2.62	367	33	1.42
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...

4.3 Descriptive Analysis

Kaur et al. (2018) imply that by outlining the connection between variables in a sample or population, descriptive statistics help to organize and summarize data. When doing research, calculating descriptive statistics is an essential initial step that should always be completed before performing inferential statistical comparisons. This section shows the descriptive analysis of the dataset obtained. Furthermore, any result will be analyzed.

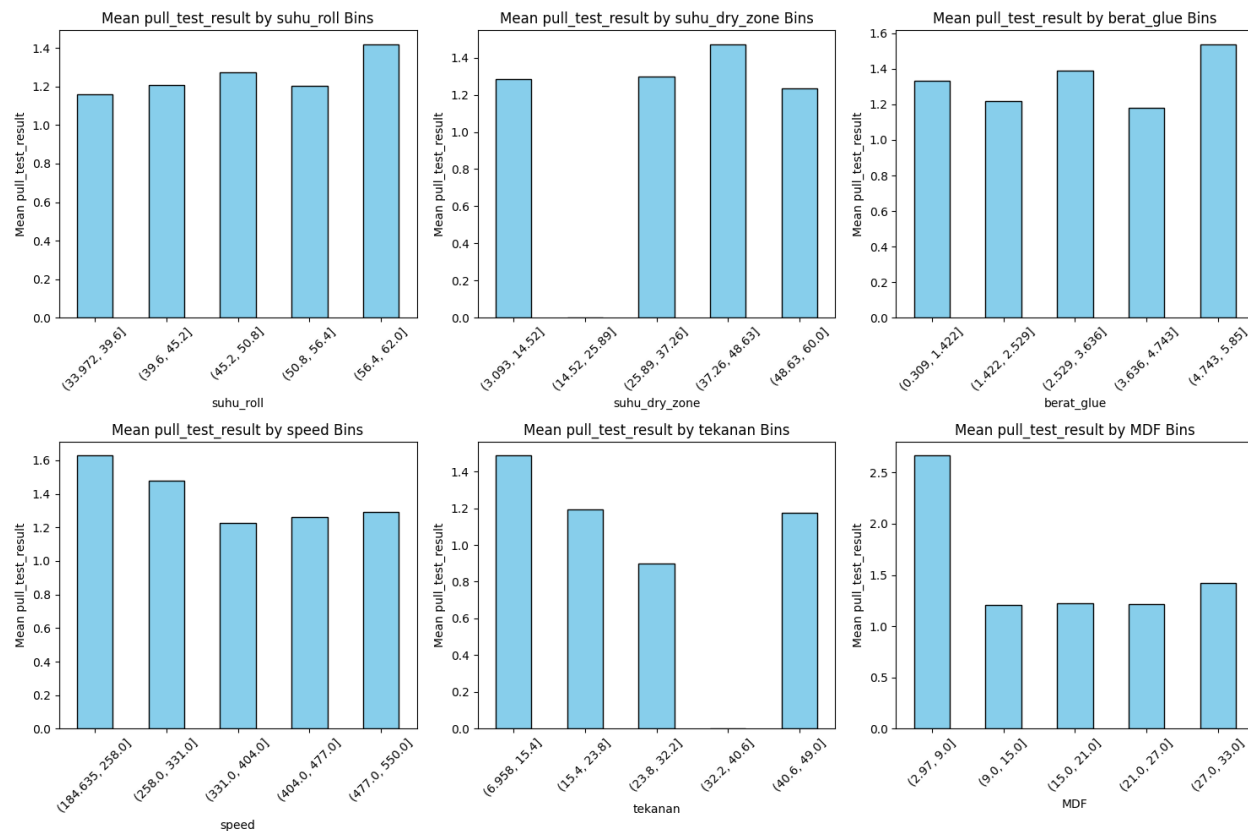
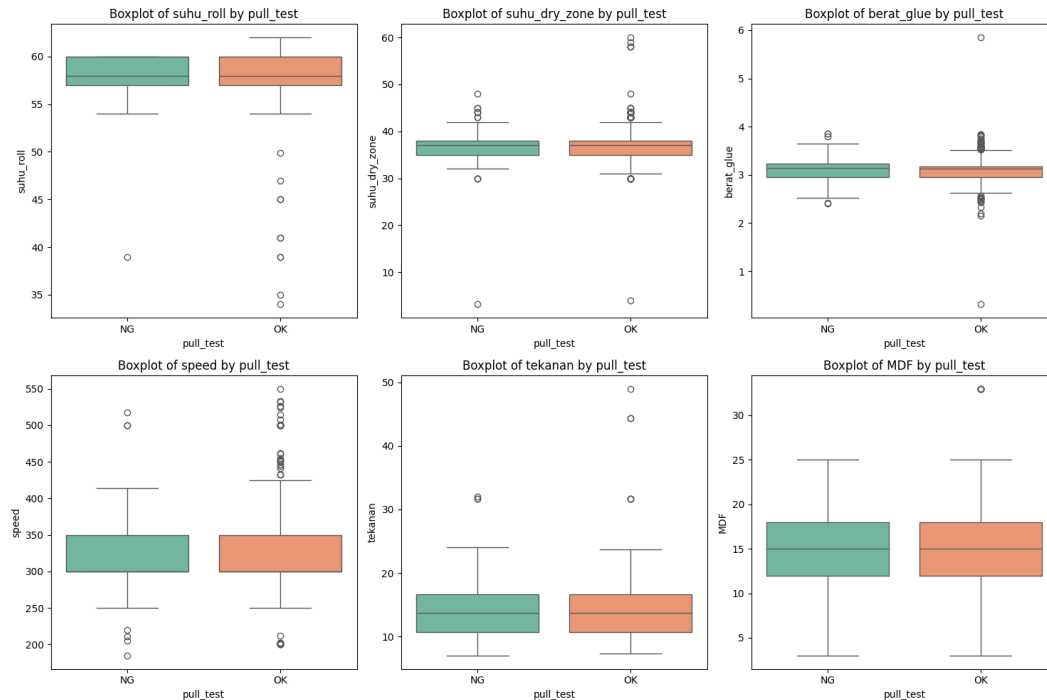


Figure 4.1 Descriptive analysis by building histogram visualization of variables towards the pull test result variable.

From Figure 4.1, we can observe a positive trend in the independent variables, such as the roll temperature. It indicates that the pull test result is positive, above the increase in the roll temperature values. The results of the MDF factors and roll speed indicated a downward trend. Negative trends indicate that the pull test result will be bearish as the values of these factors increase. The roll pressure variables revealed a novel finding: the pull test result indicates a negative trend in the bins for values between 7 and 32. The graph did, however, rise with the bins for 40 to 50 roll pressure values, and the mean of the pull test result was around 1.1. When glue weight and pull test results are compared, the results indicate a balanced trend, with the pull test result variable showing a little upward trend. This indicates that the pull test yields a slightly better result as the weight of the adhesive increases. Based on the aforementioned result, we can conclude that each variable's contribution to the pull test result varies and is dependent on its value. Another conclusion is that, in general, the independent factors do not show any significant trend results in the pull test, with the exception of the MDF variable, where it is evident that most low MDF values yield more positive outcomes than the remaining MDF data.



In Figure 4.2 it can be inferred that the proportion of OK and NG products are visually the same, but differs in terms of the outliers. The existence of outliers can be spotted through each boxplot. From the boxplots, it can be incurred that some boxplots were rich of outliers. The boxplot uses a rule which if the pull test result <1 then it will be classified as NG, with pull test result ≥ 1 will be considered as OK.

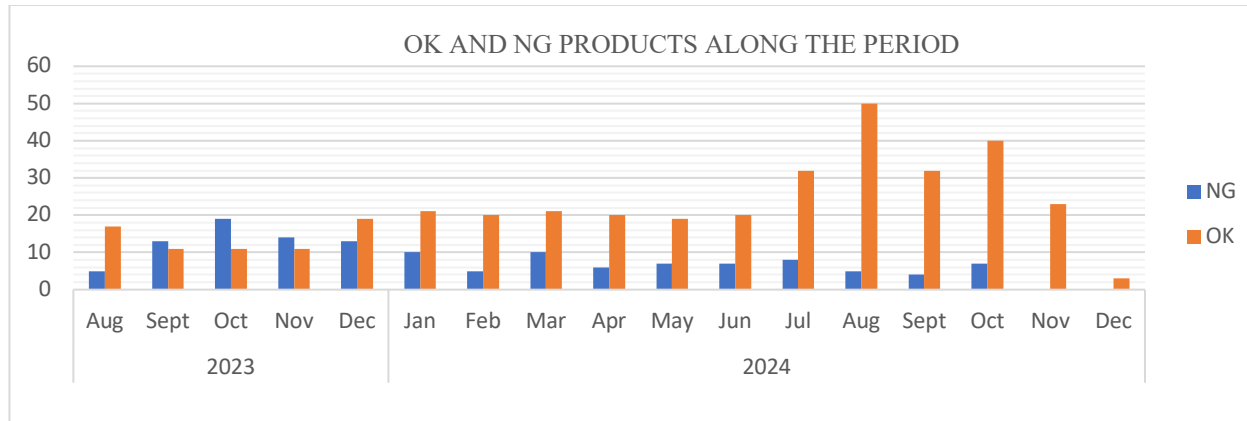


Figure 4.3 OK and NG products distribution along the period from August 2023 to early December 2024

According to Figure 1.1 the percentage of NG products were about 29% across all pull test conducted along the period. This is such a big number related to the number of NG products. Across all periods, the highest number of NG were obtained in October 2023. From the Figure 4.3 we can see that the trend of NG was bearish. At last, the very high performing month is November 2024 where in that month there is no NG products occurred. The graph incurred that there was a negative trend in number of NG over the time while the number of most pull test conducted was in August 2024 with very low number of NG compared to other months.

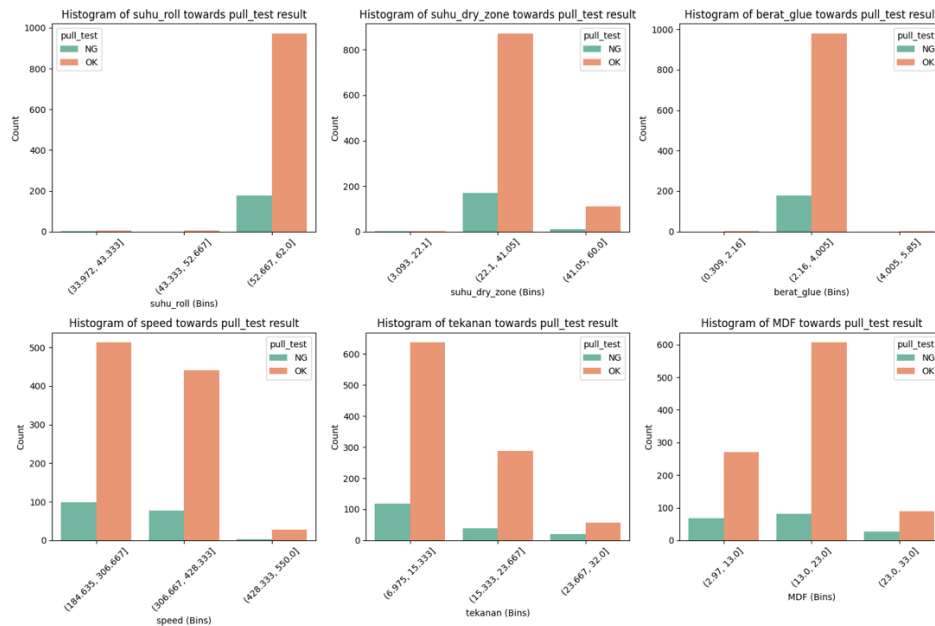


Figure 4. 4 Histogram of each variable towards the pull test result

The histogram of each variable for the pull test result can be seen in Figure 4.4. From the roll temperature variable, the majority of the pull test results lie in the bin of the interval of 52.667 to 62. For the dry zone temperature, it is noted that most of the data lies in the bin of interval of 22.1 to 41.05, with some data lying in a higher interval of bin of 41.05 to 60. Another information from the speed variable is that the highest bin contribution to NG products was the 184.635 to 306.667, followed by the second most NG, which is the bin that covers the interval of 306.667 to 428.333. The variable of pressure, on the other hand, lies mostly in the first bin, which covers the 6.975 to 15.333 interval of the pressure variable. This bin was followed by the second highest contributor, which was a bin with the interval from 15.333 to 23.667 of the pressure variable. And lastly, for the MDF variable, the bin that covers 13 to 23 of the MDF variables was filled with the most data. But on the other hand, the number of NG was surprisingly the same as another bin that filled with data from interval 2.97 to 13 of the MDF variable.

4.4 Data Training and Data Testing

Before the classification algorithm is conducted, the data needs to be divided into two portions of data, which are the data training and the data testing. Data training is a subset of the dataset which used to train and build the model, while data testing is useful to measure the performance of the

trained model. Further performance measurement will be conducted. The proportion used to divide the dataset into data training and data testing is 80% and 20%, respectively.

Table 4. 5 Data training and data testing division

	Data Training	Data Testing	Total
Percentage	80%	20%	100%
Amount	929	232	1161

Based on Table 4.5, the amount of data training and data testing used from the dataset is 929 and 232 data points, respectively. The data was chosen randomly to avoid overfitting, and the data will be processed using the Python programming language.

4.5 Hyperparameter Tuning

Following the separation of the dataset into training and testing data, we must use the hyperparameter tuning technique from scikit-learn, RandomizedSearchCV, to adjust the parameters before moving on with the real random forest implementation. The RandomizedSearchCV will look for the optimal parameter combination within the specified scale. Utilizing this strategy will save time because the prediction algorithms will be used repeatedly, updating each time due to new entries.

Table 4.6 Parameters used for the random search value

Parameter	Random search value
n_estimators	50 to 1000
max_depth	10 to 1000
min_samples_leaf	1 to 50

The table shown in Table 4.6 is the list of parameters that will be randomly searched through the given possible values. RandomizedSearchCV will need a number for iterations since it uses a number of iterations for searching the best value. In this study, the number of iterations is 30, which means it will search 30 random combination values from the given parameters, and it will result in the best combinations based on the mean squared error. For a better comprehension of the case study, the performance will then be evaluated by the root mean squared error (RMSE) method. This research will use 5-fold cross-validation to evaluate each iteration combination, and each value of the combination will be tested through 5-fold cross-validation to ensure the best scenario

among all searched possibilities. Since it uses the randomized search, repeated processes were conducted to see the stability and the best result of the given method.

Table 4.7 Iterations and the result

Iteration	n_estimators	max_depth	min_samples_leaf	RMSE
1	664	710	21	0.2107031
2	789	254	13	0.2102450
3	992	437	8	0.2110042
4	389	574	11	0.2108029
5	393	671	16	0.2105144

Results given by the Table 4.7, from 5 iterations, the value of RMSEs are vary but not significantly different from one another. This could be the legitimation to use RandomizedSearchCV to save time. The best iteration was the 2nd one with the lowest value of RMSE. Therefore, the value of the given parameters for n_estimators, max_depth, and min_samples_leaf give 789, 254, and 13 respectively, will be used to build the model.

4.6 Random Forest Implementation

Given the optimum result using the hyperparameter tuning, the random forest regressor will be build based on value of these parameters. In the application of random forest, n_estimators used to determine the number of the trees that will be used which in this case, 789 number of trees with 254 of depth which shown by the value of max_depth and 13 min_samples_leaf. These set of parameters combination will make a very large training field area, and it will be hard to identify each one. The Figure 4.4 shows an example of a random tree built by the random forest implementation. From the top of the tree, the pressure variable was on top because of the variable importance which influence the whole dataset. Following the root node, the glue weight variable was in the second split node if the first rule was true. After that the speed variable was following the weight glue variable if the rule is true, and lastly if all rules were satisfied, then the value of this rule would be 1.11.

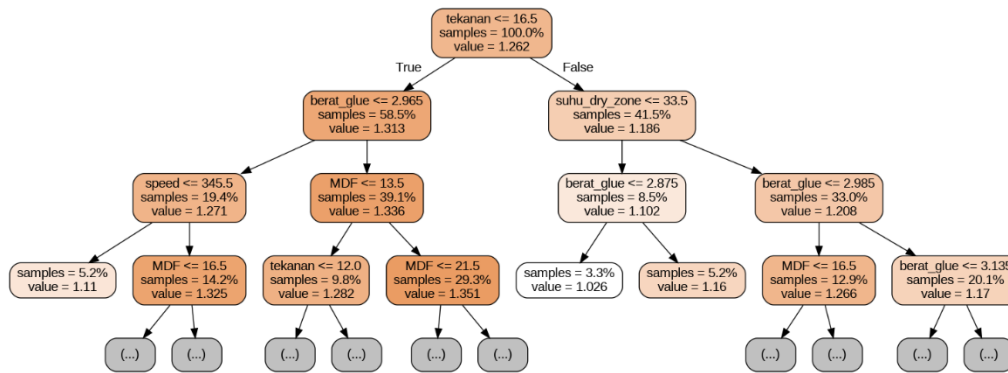


Figure 4.4 Random tree sample visualization

Referring the random tree example as shown in Figure 4.4, there are several rules were formed in this tree, and several of those are:

1. If pressure less than 16.5, and dry zone temperature less than 33.5, and glue weight less than 2.875, then the value of pull test will be 1.026. If the glue weight rule is not satisfied, then the value of pull test will be 1.16.
2. If pressure less than 16.5, and glue weight less than 2.965, and speed less than 345.5, then the value of pull test will be 1.11.
3. Because of the highly set of max_depth parameters, the other rule still continues until they reach the leaf node.

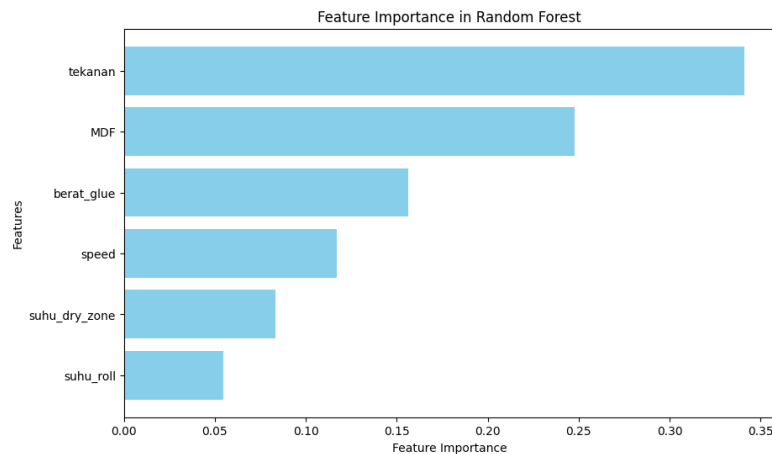


Figure 4. 5 Feature importance by the random forest model

The random forest model obtained also produces for the feature importance of each variable used. In this study, the figure shown by Figure 4.6 has the list of the most important variable from the model. The graph implies that the pressure variable was the most important variable, followed

by MDF, glue weight, speed, dry zone temperature, and roll temperature. The top position obtained by pressure variable was in line with the random tree example shown in Figure 4.4. This feature importance was displayed and measured as the sum of the total is 1.

Table 4.8 Variable importance

Variable	Importance value
Pressure	0.340986
MDF	0.247678
Glue Weight	0.156135
Speed	0.116936
Dry Zone Temperature	0.083408
Roll Temperature	0.054857

4.7 Prediction and Evaluation using SHAP Analysis

The random forest regression model that has been built to predict the outcome of the pull test will be evaluated using the SHAP analysis. Figure shown in Figure 4.7 gives insights of the impact of various machine parameters on the model's predictions toward the pull test outcomes of the soft laminating process.

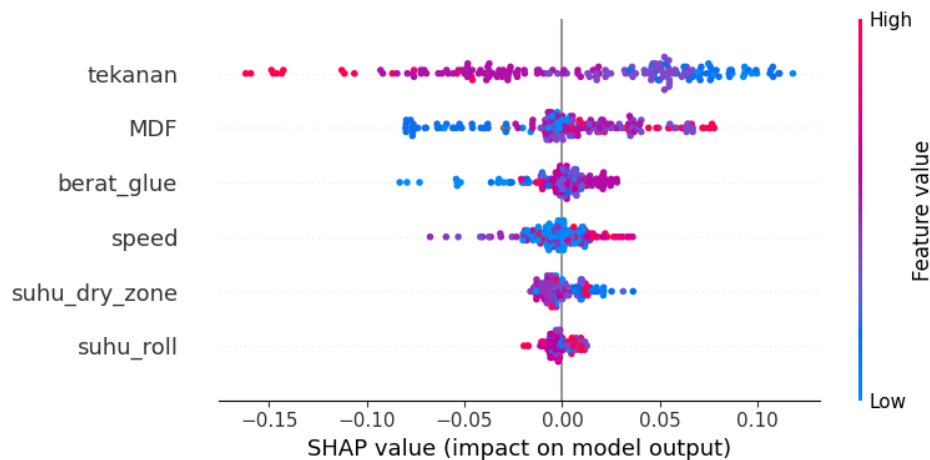


Figure 4.6 The SHAP analysis on the feature value based on the built random forest regression model

The x-axis displays SHAP values, which indicate how much each attribute effects the model's output. Positive SHAP numbers imply an increase in the expected outcome, whilst negative ones indicate a drop. From the figure, "*tekanan*" (pressure) appears to have a substantial influence, with a wide range of SHAP values. High pressure (indicated in red) often declines the prediction, but low pressure (blue) has the opposite effect. "MDF" also plays an important role, with a broad SHAP

distribution, implying that differences in MDF qualities have a significant impact on the pull test findings. "*Berat_glue*" (glue weight) has a modest impact, with increasing glue weight tending to increase adhesion strength. "Speed" has a lower impact, with a concentrated SHAP value distribution at zero, implying that its influence is less noticeable than other characteristics. "*Suhu_dry_zone*" (dry zone temperature) and "*suhu_roll*" (roll temperature) both have significant impacts, demonstrating that maintaining ideal temperatures is critical for good pull test result. As we also know that the order of importance value by the SHAP analysis is in line with the one obtained from the random forest regression as shown in Figure 4.6 and Figure 4.7. This indicates an accurate order of importance among the variables used as two different methods result in the same.

For a better understanding, an analysis of each prediction result will be analyzed using the SHAP analysis method as well.

Table 4.9 Prediction example

Variable	Value
Pressure	16.6667
MDF	18
Glue Weight	2.98
Speed	314
Dry Zone Temperature	42
Roll Temperature	55
Prediction result	1.22

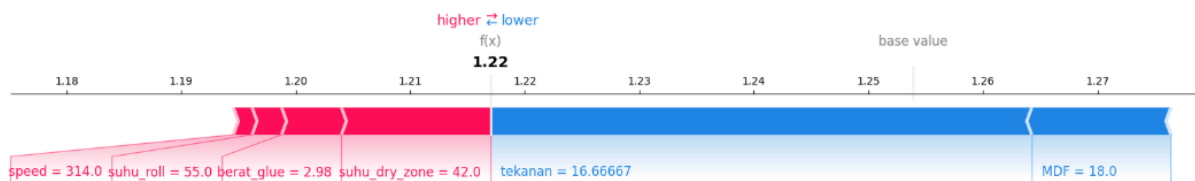


Figure 4.7 SHAP Waterfall diagram on an example prediction

This SHAP waterfall diagram shows how different machine settings affect the final prediction for a single data point in the pull test result model. The base value provides the predicted model forecast prior to feature contributions. Each characteristic either raises (red) or lowers (blue) the

anticipated value, with the final predicted value displayed at the top (1.22). In this scenario, "speed" (314.0), "suhu_roll" (55.0), "berat_glue" (2.98), and "suhu_dry_zone" (42.0) all help to improve the anticipated outcome, as represented by the red color. This implies that greater values for these factors drive the forecast upward. On the other hand, "tekanan" (16.67) and "MDF" (18.0) have a negative influence on the prediction, as seen in blue, lowering the final number.

4.8 Model Evaluation

An evaluation was conducted to assess the performance of the random forest algorithm model. This to ensure the reliability of the model to the real-world application. Table 4.10 and Figure 4.8 were shown below as the result of the model evaluation.

Table 4.10 Prediction simulation on data testing

Actual	Predicted	Absolute Error
1.44	1.38	0.05
1.02	1.12	0.10
1.15	1.18	0.03
1.25	1.13	0.12
1.06	1.14	0.07
1.44	1.24	0.20
1.29	1.34	0.05
1.54	1.30	0.24
0.89	1.09	0.21
1.32	1.30	0.02
RMSE		0.2105

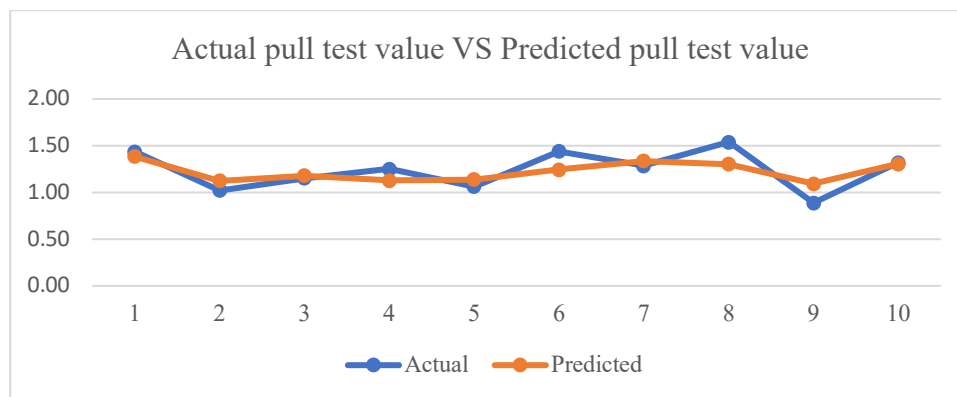


Figure 4.8 The actual pull test value VS predicted pull test value

Table 4.10 displays the actual pull test results, the predicted values, and their respective absolute errors. Furthermore, Figure 4.5 presents a visual comparison of real and predicted values. The absolute errors presented in Table 4.10 vary from 0.02 to 0.24, suggesting that the model's predictions are predominantly around the actual values. The most significant discrepancy (0.24) arises at an actual value of 1.54, when the model overestimates the pull test result as 1.30. In contrast, the minimal variation (0.02) is noted in the last test scenario, indicating a significant degree of accuracy in certain circumstances. The Root Mean Squared Error (RMSE) of 0.2105 indicates that, on average, the model's predictions diverge by around 0.21 units from the actual values. The low RMSE value implies good prediction performance, indicating that the model effectively captures underlying patterns in the dataset with little mistakes.

4.9 System Recommendations

To ease the operator effort, this research also provides recommendations the operator can use to get the best result. Below is the result of simulated variable values recommendations so the operator can use.

Table 4.11 Simulated system recommendations

Roll Temp.	Dry Zone Temp.	Glue Weight	Speed	Pressure	MDF	Predicted Pull Test
40	49	1.33	533	1.67	3	1.21
41	58	1.18	205	1.67	3	1.23
52	52	2.86	515	1.67	3	1.26
54	33	4.23	544	1.67	3	1.29
55	50	4.31	442	1.67	3	1.24
39	31	5.81	376	7.67	9	1.30
34	41	3.22	498	7.67	9	1.24
45	54	3.22	233	7.67	9	1.30
43	57	2.34	372	7.67	9	1.23
48	59	2.43	235	7.67	9	1.23
53	54	5.07	315	10.67	12	1.30
40	50	0.68	513	10.67	12	1.20
37	54	3.43	500	10.67	12	1.23
59	38	0.86	530	10.67	12	1.21
53	57	0.51	448	10.67	12	1.20

Roll Temp.	Dry Zone Temp.	Glue Weight	Speed	Pressure	MDF	Predicted Pull Test
47	46	3.98	234	13.67	15	1.31
35	59	3.34	238	13.67	15	1.29
55	58	4.98	375	13.67	15	1.32
45	31	5.27	386	13.67	15	1.32
47	45	0.80	455	13.67	15	1.22
56	57	2.12	374	16.67	18	1.18
49	42	3.30	522	16.67	18	1.22
57	55	1.87	401	16.67	18	1.20
61	42	1.95	225	16.67	18	1.18
46	30	5.78	529	16.67	18	1.21
48	55	2.75	226	23.67	25	1.11
52	36	2.05	425	23.67	25	1.15
37	59	2.59	415	23.67	25	1.13
44	58	5.53	220	23.67	25	1.08
40	56	1.21	427	23.67	25	1.13
57	30	0.35	320	31.67	33	1.12
44	48	4.25	265	31.67	33	1.08
36	30	5.44	445	31.67	33	1.12
59	38	5.09	212	31.67	33	1.09
41	41	0.67	217	31.67	33	1.10

CHAPTER V

DISCUSSION

5.1 Random forest regression model

The purpose of the research is to build a machine learning model that can predict the outcome of the adhesive strength of products in the soft laminating machine. By doing so, the possibility of a product batch failure caused by unmet specifications of the product can be prevented, and thus, the risk of loss can be minimized. Random forest, specifically random forest regression, was used to predict the value of the pull test result because the prediction is in the form of integer values.

There are 1161 amount of data that being used in this research to train and test the model built with the portion of 929 data training and tested with data testing as much as 232. The data was obtained from 2 different databases of the company. The pull test dataset was manually input into the website by the operator, so the risk of the operator inputting noisy data is high. Cleaning data was done on the machine, setting data before it took the synchronization process to the other dataset. There were some modifications done before the data could be synchronized, like renaming the model and the glue, since these two were manually inputted by the operator. Another dataset was obtained from a database that contains the machine setting variables such as the roll temperature, roll pressure, glue weight, temperature of the dry zone, speed of the roll, and the MDF size. The system initially recorded the machine setting data and uploaded it to the database. After some effort was made to the datasets, these two are combined based on the model and the date of production. Therefore, each machine setting of data will have the results of the pull test. In this process, some data that wasn't matched based on the model and the date of production were then erased, assuming that there was no pull test conducted for the product batch.

Random forest is a machine learning technique that utilizes the quantity of trees built and some other features as the foundation of the model, like the number of sample leaves, the depth of the tree, and methods to choose the best leaf. In this research, Python 3.11.11 was used as the programming language to execute the code with Google Colaboratory as the medium for it. Using the scikit-learn library, RandomizedSearchCV is used to optimize the value of the model's attributes. The use of this method is to ensure that this technique will be applicable in the real world, where efficiency is a must. The operator can use this method to get a quick model

development as the data was being updated over time. Therefore, a reliable method where the training data will be kept updated can be achieved.

Table 4.10 displays the outcome of the model's real-world application. In the trial session of the model, the operator of the soft laminating machine tested the reliability of the prediction application. Over a week, the operator conducted 10 pull tests. Furthermore, the researcher helps the operator in applying the method to ensure the smoothness and to prevent any unwanted errors that might occur. Figure 4.5 graphically depicts the model's performance throughout the test samples. The actual values (blue line) and anticipated values (orange line) exhibit a comparable pattern, with only minor differences observed. The model seems to mitigate modest changes in the real values, resulting in slight underestimations or overestimations. The overall consistency between the two lines indicates that the model accurately predicts the pull test values to a satisfactory extent. The results from both numerical and graphical studies demonstrate that the Random Forest Regression model well predicts pull test values, exhibiting a low RMSE and very little absolute error. Although slight inconsistencies exist, they do not substantially affect the model's overall predictive accuracy.

Table 4.11 shows the recommendations by system that the operator can use if they have no idea how to achieve a good pull test result. The table provides combinations based on the MDF since it is an uncontrollable variable. From the table operator can choose 5 options for each MDF value that exists in the database. By this idea, the operator can have a satisfactory pull test result by using the combination in the table.

5.2 Hyperparameter tuning using the RandomizedSearchCV

The model built in this research was expected to have the best performance without having to do a lot of research in building it. The hyperparameter tuning method used in this research was the RandomizedSearchCV. The technique was chosen for its rapid performance in determining the best model's attributes like the n estimators, max depth, and min samples leaf. These three attributes were chosen as the variable attributes, meaning that it will search in the interval of values that have been set. The interval of search values for n estimators, max depth, and min samples leaf is 50 to 1000, 10 to 1000, and 1 to 50, respectively. To make it more efficient, the number of

iterations is limited to 30 iterations, and the performance of each iteration will be evaluated using the RMSE.

The results of hyperparameter tweaking for the Random Forest Regression model, using RandomizedSearchCV with 30 iterations and 5-fold cross-validation, illustrate how different hyperparameter configurations affect model performance. The table shows five iterations with different settings for `n_estimators`, `max_depth`, and `min_samples_leaf`, along with the related Root Mean Squared Error (RMSE). Among the evaluated setups, Iteration 2 had the lowest RMSE (0.2102450), with `n_estimators` = 789, `max_depth` = 254, and `min_samples_leaf` = 13. This implies that a somewhat high number of estimators, paired with a deep tree structure and slightly higher leaf size, yields the best performance for this dataset. Meanwhile, Iteration 3, which employed 992 estimators, `max_depth` of 437, and `min_samples_leaf` of 8, had the highest RMSE (0.2110042), demonstrating that raising these parameters did not necessarily increase the model's predictive capacity and may have resulted in overfitting. Iterations 4 and 5 also gave somewhat comparable RMSE values (0.2108029 and 0.2105144, respectively), although they had different parameter choices. This shows that Random Forest is resistant to minor alterations in the hyperparameters, and that various configurations can attain near-optimal performance. Notably, Iteration 1, which had the fewest trees (`n_estimators` = 664) and the highest `max_depth` (710), fared worse than Iteration 2, highlighting the necessity of tweaking hyperparameters rather than merely increasing model complexity.

5.3 SHAP Analysis method

Random forest regression is a complex and non-linear machine learning model that can be challenging to interpret. The SHAP analysis is a technique that is useful for breaking down the outcome of the random forest regression model. It clarifies how features influence the predictions, even in non-linear problems. Based on Figure 4.6, the variable pressure (*tekanan*) had a widespread SHAP value among other features. It shows how high pressure (noted in red dots) on the machine could lead to a declining pull test result. The pressure variable measures the roll's ability to exert force on the product at a specific value. This finding means that a higher pressure of the roll towards the product can result in a lower adhesive strength, where a lower pressure (noted in blue dots)

leads to a higher pull test result. This is an intriguing discovery, as one could intuitively assume the opposite outcome.

Apart from the pressure feature, the MDF has an effortless interpretation where a lower MDF could push the prediction result to a lower result of the pull test, too. For glue weight (*berat_glue*), a lower glue weight might result in a lower adhesive strength of the product. The interesting discovery regarding this feature is that a decreasing prediction result for a lower value does not necessarily imply a positive impact on a higher value. This means that increasing the glue on the machine, hoping that it will have a better adhesive strength, is wrong. In the middle of the x-axis of the glue weight feature, average and high feature values were gathered, showing that the feature's performance in this interval value cannot be determined. For the speed feature, the low and average feature magnitudes lie at the middle point of the axis, with some of the average values showing a negative influence on the prediction result. In the same variable, red dots were seen to be on the right-hand side of the SHAP value, showing its positive impact on the adhesive's strength.

The dry zone temperature (*suhu_dry_zone*) and roll temperature (*suhu_roll*) features mainly lie at the middle of the SHAP value, whether low, average, or high values of the variable. This shows a low impact of these features on the prediction results or the inability of the method to decide how these variables behave in the model. This study indicates that increasing speed, roll temperature, glue weight, and drying zone temperature may enhance adhesion in the pull test. Nonetheless, excessive pressure and MDF characteristics may adversely affect adhesion strength. PT. XYZ may utilize this data to optimize machine settings for superior quality in the soft laminating process.

CHAPTER VI

CLOSING

6.1 Conclusions

Based on the research that has been conducted, it is obtained some conclusions have been obtained, as follows:

1. The machine learning model that has been developed in this research is a model using random forest regression. Based on a previous comparative study, it was obtained that random forest regression is the best-performing machine learning model when compared to artificial neural network (ANN) and support vector regression (SVR), where it generates more accurate predictions (Wu et al., 2017). The model built was supported by the RandomizedSearchCV from scikit-learn (a Python library) to ensure the best performance and quick result, as it will be implemented in the industry. Random forest regression was chosen to cover the non-linear problem as well as the integer value of the pull test, which cannot be handled by random forest classification.
2. From the model that has been developed and tested, it is obtained that using the random forest regression and tuned by the hyperparameter tuning RandomizedSearchCV repeated for 5 times, the best machine learning model in predicting the pull test result at soft laminating machine is the one that has n estimators of 789, max depth of 254, and min samples leaf of 13, with the RMSE value of 0.2102450. The model was applied and tested to face the real-world case and got an RMSE of 0.2105, with the highest error of 0.24, and the most accurate prediction recorded an error of only 0.02 from 10 times of model trial sessions.
3. It is obtained from the model that the most influential features are pressure, MDF, glue weight, speed, dry zone temperature, and roll temperature sequentially. For the pressure feature, it will influence the pull test prediction outcomes based on its value, where it will positively push the prediction if it is set to a low value. For the MDF feature, it will negatively affect the prediction outcomes if it is set to a low value. For the glue weight feature, it will affect negatively if it is set to a low value, and it will generally have no big impact on the prediction if it is set to an average or high value. For the speed feature, it will

positively affect the prediction outcomes if it is set to a high value. For the temperature features, they generally have little impact on the prediction results.

6.2 Suggestions

According to the research, there is knowledge learned along the way of this study. Lessons learned by this study can be reflected and implemented in the upcoming study, which are:

1. Future studies should use and perform deeper research on the best methods that are needed to tackle the problem. It will be better if the method used to build the machine learning model can address the problem of the case study that needs to be solved.
2. In building a machine learning model, it is needed to pay attention to all variables that are involved in the case study environment. Lack of variables that may be crucial to the prediction may result in an inaccurate prediction result.
3. The company can reflect on the use of the prediction technique as it will provide the insight that may be needed to make decisions in the area of company.

BIBLIOGRAPHY

- Alhakeem, Z. M., Jebur, Y. M., Henedy, S. N., Imran, H., Bernardo, L. F. A., & Hussein, H. M. (2022). Prediction of Ecofriendly Concrete Compressive Strength Using Gradient Boosting Regression Tree Combined with GridSearchCV Hyperparameter-Optimization Techniques. *Materials*, 15(21). <https://doi.org/10.3390/ma15217432>
- Amaliah, S., Nusrang, M., & Aswi. (2022). Penerapan Metode Random Forest Untuk Klasifikasi Varian Minuman Kopi di Kedai Kopi Konijiwa Bantaeng. *VARIANSI: Journal of Statistics and Its Application on Teaching and Research*, 4(3), 121–127. <https://doi.org/10.35580/variansiunm31>
- Antonini, A. S., Tanzola, J., Asiain, L., Ferracutti, G. R., Castro, S. M., Bjerg, E. A., & Ganuza, M. L. (2024). Machine Learning model interpretability using SHAP values: Application to Igneous Rock Classification task. In *Applied Computing and Geosciences* (Vol. 23). Elsevier B.V. <https://doi.org/10.1016/j.acags.2024.100178>
- Antosz, K., Kozłowski, E., Sęp, J., & Zabinski, T. (2022). The use of random forests to support the decision-making process for sustainable manufacturing. *Journal of Physics: Conference Series*, 2198(1). <https://doi.org/10.1088/1742-6596/2198/1/012006>
- Bacanin, N., Stoean, C., Zivkovic, M., Rakic, M., Strulak-Wójcikiewicz, R., & Stoean, R. (2023). On the Benefits of Using Metaheuristics in the Hyperparameter Tuning of Deep Learning Models for Energy Load Forecasting. In *Energies* (Vol. 16, Issue 3). MDPI. <https://doi.org/10.3390/en16031434>
- Bardenet, R., Brendel, M., Kégl, B., Sebag, M., & Fr, S. (2013). *Collaborative hyperparameter tuning* (Vol. 28).
- Baumers, M., Dickens, P., Tuck, C., & Hague, R. (2016). The cost of additive manufacturing: Machine productivity, economies of scale and technology-push. *Technological Forecasting and Social Change*, 102, 193–201. <https://doi.org/10.1016/j.techfore.2015.02.015>
- Breiman, L. (2001). *Random Forests* (Vol. 45).
- Brendel, M., & Schoenauer, M. (2011). *Proceedings of the 13th annual conference companion on Genetic and evolutionary computation*. Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation - GECCO '11. <https://doi.org/doi:10.1145/2001858.2002053>
- Büyüksari, Ü. (2013). Surface characteristics and hardness of MDF panels laminated with thermally compressed veneer. *Composites Part B: Engineering*, 44(1), 675–678. <https://doi.org/10.1016/j.compositesb.2012.01.087>
- Çinar, Z. M., Nuhu, A. A., Zeeshan, Q., Korhan, O., Asmael, M., & Safaei, B. (2020). Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0. *Sustainability (Switzerland)*, 12(19). <https://doi.org/10.3390/su12198211>
- Crosby, P. B. (1979). *Quality is free-if you understand it* (4th ed.). Winter Park Public Library History and Archive Collection.
- Cunningham, P., Cord, M., & Delany, S. J. (2008). Supervised Learning. *Springer, Berlin, Heidelberg*. https://doi.org/https://doi.org/10.1007/978-3-540-75171-7_2
- Dietterich, T. G. (1990). Machine Learning 1. In *Annual Review of Computer Science* (Vol. 4).
- Dogan, A., & Birant, D. (2021). Machine learning and data mining in manufacturing. In *Expert Systems with Applications* (Vol. 166). Elsevier Ltd. <https://doi.org/10.1016/j.eswa.2020.114060>

- Ekanayake, I. U., Meddage, D. P. P., & Rathnayake, U. (2022). A novel approach to explain the black-box nature of machine learning in compressive strength predictions of concrete using Shapley additive explanations (SHAP). *Case Studies in Construction Materials*, 16. <https://doi.org/10.1016/j.cscm.2022.e01059>
- Elgeldawi, E., Sayed, A., Galal, A. R., & Zaki, A. M. (2021). Hyperparameter tuning for machine learning algorithms used for arabic sentiment analysis. *Informatics*, 8(4). <https://doi.org/10.3390/informatics8040079>
- Frawley, W. J., Piatetsky-Shapiro, G., & Matheus, C. J. (1992). *Knowledge Discovery in Databases: An Overview*.
- Guleria, P., Thakur, N., Sood, M., & Pradesh, H. (2014). Predicting Student Performance Using Decision Tree Classifiers and Information Gain. *International Conference on Parallel*, 126–129. <https://doi.org/10.1109/PDGC.2014.7030728>
- Hastie, S., Tibshirani, R., & Friedman, J. (2009). *Unsupervised Learning*. Springer, New York, NY. https://doi.org/https://doi.org/10.1007/978-0-387-84858-7_14
- Isabona, J., Imoize, A. L., & Kim, Y. (2022). Machine Learning-Based Boosted Regression Ensemble Combined with Hyperparameter Tuning for Optimal Adaptive Learning. *Sensors*, 22(10). <https://doi.org/10.3390/s22103776>
- Kaur, P., Stoltzfus, J., & Yellapu, V. (2018). Descriptive statistics. *International Journal of Academic Medicine*, 4(1), 60. https://doi.org/10.4103/IJAM.IJAM_7_18
- Kiangala, S. K., & Wang, Z. (2021). An effective adaptive customization framework for small manufacturing plants using extreme gradient boosting-XGBoost and random forest ensemble learning algorithms in an Industry 4.0 environment. *Machine Learning with Applications*, 4, 100024. <https://doi.org/10.1016/j.mlwa.2021.100024>
- Kim, Y., & Kim, Y. (2022). Explainable heat-related mortality with random forest and SHapley Additive exPlanations (SHAP) models. *Sustainable Cities and Society*, 79. <https://doi.org/10.1016/j.scs.2022.103677>
- Li, Z. (2022). Extracting spatial effects from machine learning model using local interpretation method: An example of SHAP and XGBoost. *Computers, Environment and Urban Systems*, 96. <https://doi.org/10.1016/j.compenvurbsys.2022.101845>
- Loh, W. Y. (2011). Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1), 14–23. <https://doi.org/10.1002/widm.8>
- Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., & Lee, S. I. (2020). From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, 2(1), 56–67. <https://doi.org/10.1038/s42256-019-0138-9>
- Mahesh, B. (2020). Machine Learning Algorithms - A Review. *International Journal of Science and Research (IJSR)*, 9(1), 381–386. <https://doi.org/10.21275/art20203995>
- Mulugeta, L. (2020). Productivity improvement through lean manufacturing tools in Ethiopian garment manufacturing company. *Materials Today: Proceedings*, 37(Part 2), 1432–1436. <https://doi.org/10.1016/j.matpr.2020.06.599>
- Qureshi, M. I., Khan, N., Qayyum, S., Malik, S., Sanil, H. S., & Ramayah, T. (2020). Classifications of sustainable manufacturing practices in ASEAN region: A systematic review and bibliometric analysis of the past decade of research. In *Sustainability (Switzerland)* (Vol. 12, Issue 21, pp. 1–19). MDPI. <https://doi.org/10.3390/su12218950>
- Samuel, A. L. (1959). *Some Studies in Machine Learning Using the Game of Checkers*.

- Shang, Z., & Zhang, L. (2022). The Sustainable Digitalization in the Manufacturing Industry: A Bibliometric Analysis and Research Trend. *Mobile Information Systems*, 2022. <https://doi.org/10.1155/2022/1451705>
- Shivajee, V., Singh, R. K., & Rastogi, S. (2019). Manufacturing conversion cost reduction using quality control tools and digitization of real-time data. *Journal of Cleaner Production*, 237. <https://doi.org/10.1016/j.jclepro.2019.117678>
- Sokovic, M., Jovanović, J. Š., Krivokapic, Z., & Vujovic, A. (2009). *Basic Quality Tools in Continuous Improvement Process*. <https://www.researchgate.net/publication/299050377>
- Wang, F., Wang, Y., Zhang, K., Hu, M., Weng, Q., & Zhang, H. (2021). Spatial heterogeneity modeling of water quality based on random forest regression and model interpretation. *Environmental Research*, 202. <https://doi.org/10.1016/j.envres.2021.111660>
- Wu, D., Jennings, C., Terpenney, J., Gao, R. X., & Kumara, S. (2017). A Comparative Study on Machine Learning Algorithms for Smart Manufacturing: Tool Wear Prediction Using Random Forests. *Journal of Manufacturing Science and Engineering, Transactions of the ASME*, 139(7). <https://doi.org/10.1115/1.4036350>
- Zanin, M., Papo, D., Sousa, P. A., Menasalvas, E., Nicchi, A., Kubik, E., & Boccaletti, S. (2016). Combining complex networks and data mining: Why and how. In *Physics Reports* (Vol. 635, pp. 1–44). Elsevier. <https://doi.org/10.1016/j.physrep.2016.04.005>

ATTACHMENT

Pre-Processing

```
colab.research.google.com/drive/1EjsMplXULcwWYaDDYjRcNLX3lgBt57rH#scrollTo=lxwyhywbii8S
```

```
preprocessing - regres - pull_test.ipynb ☆ Saving...
```

```
File Edit View Insert Runtime Tools Help
```

```
Q Commands + Code + Text Connect ^
```

```
[ ] import pandas as pd
from datetime import timedelta

# Load the datasets
setting_mesin = pd.read_csv('/content/setting_mesin.csv', delimiter=';')
pull_test = pd.read_csv('/content/pull_test.csv', delimiter=';')

# List of power variables
power_columns = ['power1', 'power2', 'power3', 'powera', 'powerb', 'powerc']
# List of additional variables to include (text-based)
additional_columns = ['glue', 'pvc', 'MDF']

# Convert power variables to numeric and coerce errors to NaN
pull_test[power_columns] = pull_test[power_columns].apply(pd.to_numeric, errors='coerce')

# Drop rows with NaN values in the power columns (text columns are not affected)
pull_test = pull_test.dropna(subset=power_columns)

# Ensure the 'date' columns are in datetime format
setting_mesin['date'] = pd.to_datetime(setting_mesin['date'], format='%d/%m/%y')
pull_test['date'] = pd.to_datetime(pull_test['date'], format='%d/%m/%y')

# Step 1: Initialize the new columns in 'setting_mesin'
for col in power_columns + additional_columns + ['pull_test_result']:
    setting_mesin[col] = None

# Step 2: Matching logic
for idx, row in setting_mesin.iterrows():
    model = row['model']
    date = row['date']

    # Filter pull_test by matching 'model' and look for dates >= current date
    matching_rows = pull_test[(pull_test['model'] == model) & (pull_test['date'] >= date)]

    if not matching_rows.empty:
        # Look for the closest date
        closest_row = matching_rows.loc[matching_rows['date'].idxmin()]

        setting_mesin.at[idx, col] = closest_row[col]
    else:
        # Expand search to the next 7 days
        matching_rows_7_days = pull_test[
            (pull_test['model'] == model) &
            (pull_test['date'] >= date) &
            (pull_test['date'] <= date + timedelta(days=7))
        ]

        if not matching_rows_7_days.empty:
            closest_row = matching_rows_7_days.loc[matching_rows_7_days['date'].idxmin()]
            setting_mesin.at[idx, 'pull_test_result'] = closest_row[['power1', 'power2', 'power3', 'powera', 'powerb', 'powerc']].mean()
            for col in power_columns + additional_columns:
                setting_mesin.at[idx, col] = closest_row[col]

# Step 3: Drop rows where 'pull_test_result' is still None
setting_mesin.dropna(subset=['pull_test_result'], inplace=True)

# Save the updated dataset
setting_mesin.to_excel("setting_mesin_updated.xlsx", index=False)

print(setting_mesin.head())
```

```
id date clock model part color \
0 1571 2023-08-01 12.41.00 PM L100 SIDEBORD PM
1 1572 2023-08-01 3.36.00 PM NS-F150 PM BODY BOARD PM
2 1573 2023-08-01 7.32.00 PM GA-15 II BODY BOARD BLACK
3 1574 2023-08-01 8.31.00 PM HS85 BODY BOARD BLACK
4 1575 2023-08-01 9.50.00 PM NS-SW850 BODY BOARD BLACK

posisi_rol_satu tekanan posisi_rol_dua posisi_rol_tiga ... power1 \
0 17 NaN 16 17 ... 0.8
1 8 NaN 7 8 ... 0.68
2 14 NaN 13 14 ... 1.13
3 14 NaN 13 14 ... 1.61
4 11 NaN 10 11 ... 1.18

power2 power3 powera powerb powerc glue \
0 0.97 0.92 1.06 1.1 1.09 DORRUS V698990
1 0.71 0.85 0.77 0.93 0.88 ADHESIVE KLEIBERIT V025870
2 0.82 1.1 1.18 1.63 1.5 DORRUS V698990
3 1.54 1.57 1.26 1.73 1.62 DORRUS V698990
4 1.1 1.13 1.15 1.12 1.22 DORRUS V698990

pvc MDF pull_test_result
0 MEIWA VEX3310 18.0 0.99
1 DTKEN 713666A 0.0 0.003333
```

preprocessing - regressi - pull_test.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text

[5 rows x 30 columns]

```

import pandas as pd

data = setting_mesin

print("Before Cleaning:")
print(data[['suhu_roll', 'suhu_dry_zone']].head())

# Function to clean the column
def clean_temperature(value):
    if '/' in str(value): # Check if the value contains '/'
        return str(value).split('/')[0] # Extract the first part before '/'
    else:
        return value # Keep the original value if it's already clean

# Apply the function to the columns
data['suhu_roll'] = data['suhu_roll'].apply(clean_temperature)
data['suhu_dry_zone'] = data['suhu_dry_zone'].apply(clean_temperature)

# Convert the cleaned data to numeric, if needed
data['suhu_roll'] = pd.to_numeric(data['suhu_roll'], errors='coerce')
data['suhu_dry_zone'] = pd.to_numeric(data['suhu_dry_zone'], errors='coerce')

# Preview the dataset after cleaning
print("After Cleaning:")
print(data[['suhu_roll', 'suhu_dry_zone']].head())

```

Before Cleaning:

	suhu_roll	suhu_dry_zone
0	56/60	30/33
1	54/60	31/33
2	59/60	30/35
3	60/60	32/35
4	34/60	32/35

After Cleaning:

	suhu_roll	suhu_dry_zone
0	56.0	30.0
1	54.0	31.0
2	59.0	30.0
3	60.0	32.0
4	34.0	32.0

random forest - pull_test.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text

```

import gdown
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Google Drive shareable link
shareable_link = "https://docs.google.com/spreadsheets/d/1x0ThccoJRizwH2Vu3vFAKfgyxwujf-u/edit?usp=sharing&ouid=107323683781395158363&rtopof=true&sd=true"

# Extract the file ID
file_id = shareable_link.split('/')[4].split('/')[-1]

# Create a direct download link
download_url = f"https://drive.google.com/uc?id={file_id}"

# Specify the output file name
output_file = "setting_mesin_updated.xlsx"

# Download the file
gdown.download(download_url, output_file, quiet=False)

print(f"File downloaded as {output_file}")

# Load the data
data = pd.read_excel(f'/content/setting_mesin_updated.xlsx')

# Function to clean the column #
def clean_temperature(value):
    if '/' in str(value): # Check if the value contains '/'
        return str(value).split('/')[0] # Extract the first part before '/'
    else:
        return value # Keep the original value if it's already clean

# Apply the function to the columns
data['suhu_roll'] = data['suhu_roll'].apply(clean_temperature)
data['suhu_dry_zone'] = data['suhu_dry_zone'].apply(clean_temperature)

# Convert the cleaned data to numeric, if needed
data['suhu_roll'] = pd.to_numeric(data['suhu_roll'], errors='coerce')
data['suhu_dry_zone'] = pd.to_numeric(data['suhu_dry_zone'], errors='coerce')

# RF MODEL WITH HYPERPARAMETER TUNING RANDOMIZEDSEARCHCV #

```

Random forest implementation

random forest - pull_test.ipynb

File Edit View Insert Runtime Tools Help

Commands Code Text

Connect

```

# RF MODEL WITH HYPERPARAMETER TUNING RANDOMIZEDSEARCHCV #

from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, precision_score, recall_score, ConfusionMatrixDisplay
from sklearn.model_selection import RandomizedSearchCV, train_test_split
from scipy.stats import randint
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
import pandas as pd
import matplotlib.pyplot as plt

# Define independent variables (features)
X = data[['suhu_roll', 'suhu_dry_zone', 'berat_glue', 'speed', 'tekanan', 'MDF']] # Example features

# Define the dependent variable (target)
y = data['pull_test_result'] # Replace with your actual target variable

# Split data into training and testing sets (80% train, 20% test)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Hyperparameter space for RandomizedSearchCV
param_dist = {
    'n_estimators': randint(50, 500),
    'max_depth': randint(10, 100),
    'random_state': randint(10, 100)
}

# Initialize the RandomForestRegressor
rf = RandomForestRegressor(random_state=42)

# RandomizedSearchCV setup
random_search = RandomizedSearchCV(
    estimator=rf,
    param_distributions=param_dist,
    n_iter=30, # Number of random combinations to try
    scoring='neg_mean_squared_error',
    cv=5, # 5-fold cross-validation
    verbose=2,
    random_state=42,
    n_jobs=-1 # Use all available cores
)

# Fit RandomizedSearchCV
random_search.fit(X_train, y_train)

# Fit RandomizedSearchCV
random_search.fit(X_train, y_train)

# Get the best model and parameters
best_model = random_search.best_estimator_
best_params = random_search.best_params_
print("Best Parameters:", best_params)

# Predict on the test set with the best model
y_pred = best_model.predict(X_test)

# Calculate Mean Squared Error (MSE) on test data
mse = mean_squared_error(y_test, y_pred)
print(f"Mean Squared Error of the Best Model: {mse}")

# Feature Importance
feature_importances = pd.DataFrame(
    best_model.feature_importances_,
    index=X.columns,
    columns=['Importance']
).sort_values('Importance', ascending=False)

print("\nFeature Importances:")
print(feature_importances)

results = pd.DataFrame(random_search.cv_results_)
results['mean_test_score'] = -results['mean_test_score'] # Convert to positive MSE for readability

Fitting 5 folds for each of 30 candidates, totalling 150 fits
Best Parameters: {'max_depth': 84, 'n_estimators': 137, 'random_state': 33}
Mean Squared Error of the Best Model: 1.3279579238164807

Feature Importances:
      Importance
tekanan      0.317488
berat_glue   0.190155
MDF          0.152492
speed        0.147287
suhu_roll    0.112935
suhu_dry_zone 0.079723

# RULES #

from sklearn.tree import export_text

# Assuming rf_model is your trained RandomForestRegressor

```

```

random forest - pull_test.ipynb
File Edit View Insert Runtime Tools Help

# RULES #
from sklearn.tree import export_text

# Assuming rf_model is your trained RandomForestRegressor
# Let's say you want to extract rules from the first tree in the RandomForest
tree = rf_model.estimators_[0] # Access the first tree in the forest

# Export the tree as text
tree_rules = export_text(tree, feature_names=X.columns.tolist())

# Print the rules of the tree
print(tree_rules)

[ ] # HYPERPARAMETER TUNING #
param_dist = {'n_estimators': randint(50, 500),
              'max_depth': randint(1, 20),
              'random_state': randint(20, 60)}
# Create a random forest classifier
rf_model = RandomForestRegressor()

# Use random search to find the best hyperparameters
rand_search = RandomizedSearchCV(rf_model,
                                param_distributions = param_dist,
                                n_iter=5,
                                cv=5)

# Fit the random search object to the data
rand_search.fit(X_train, y_train)

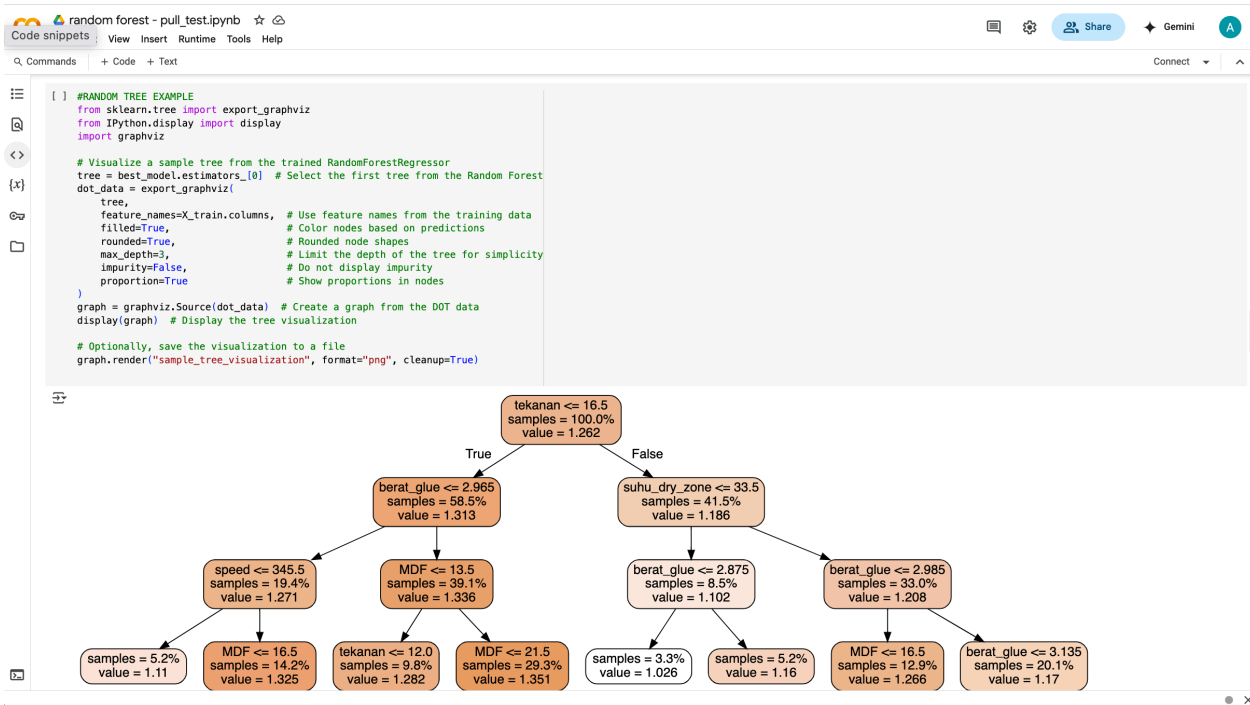
# Create a variable for the best model
best_rf = rand_search.best_estimator_

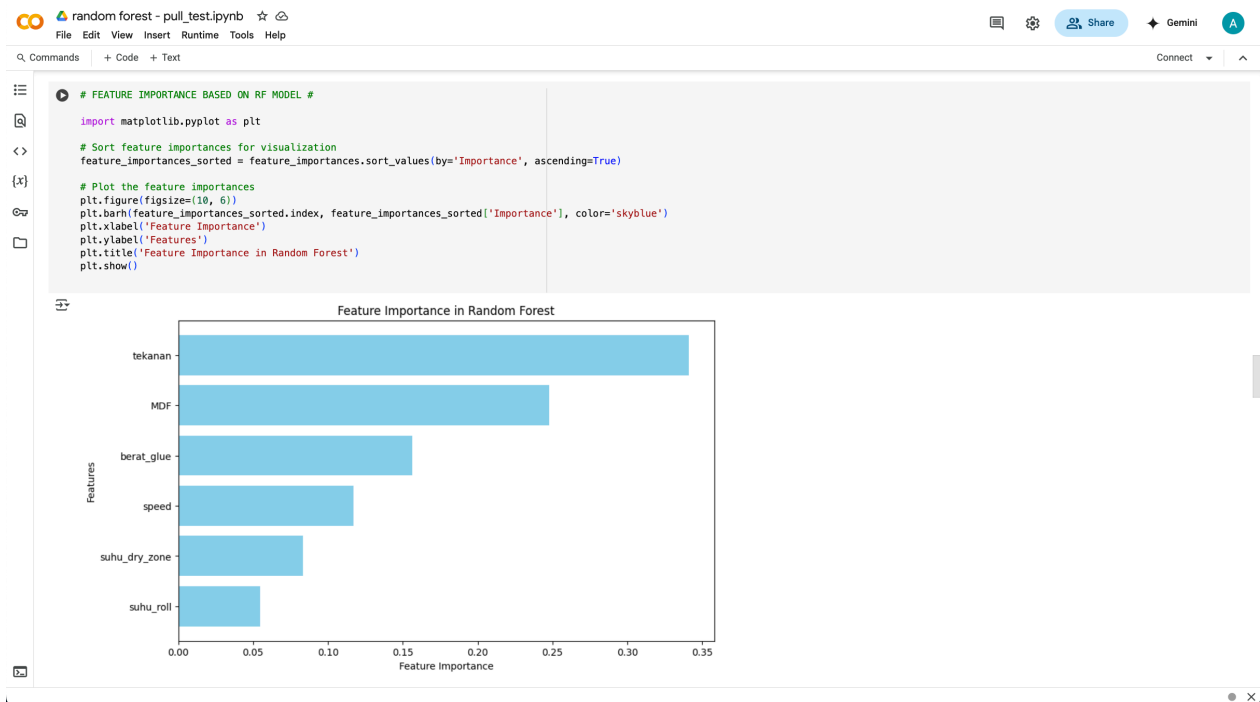
# Print the best hyperparameters
print('Best hyperparameters:', rand_search.best_params_)

Best hyperparameters: {'max_depth': 9, 'n_estimators': 87, 'random_state': 56}

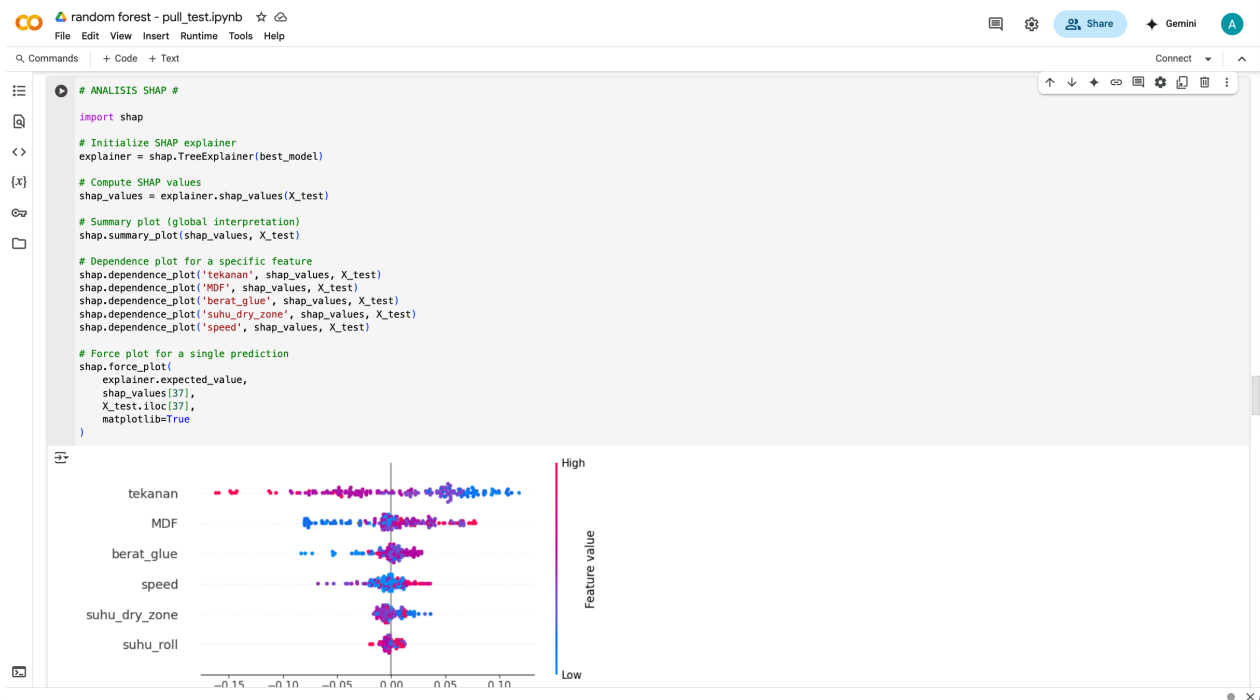
[ ] # RANDOM FOREST WITH RANDOMIZEDSEARCHCV BASED ON GLUE DORRUS #
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, precision_score, recall_score, ConfusionMatrixDisplay

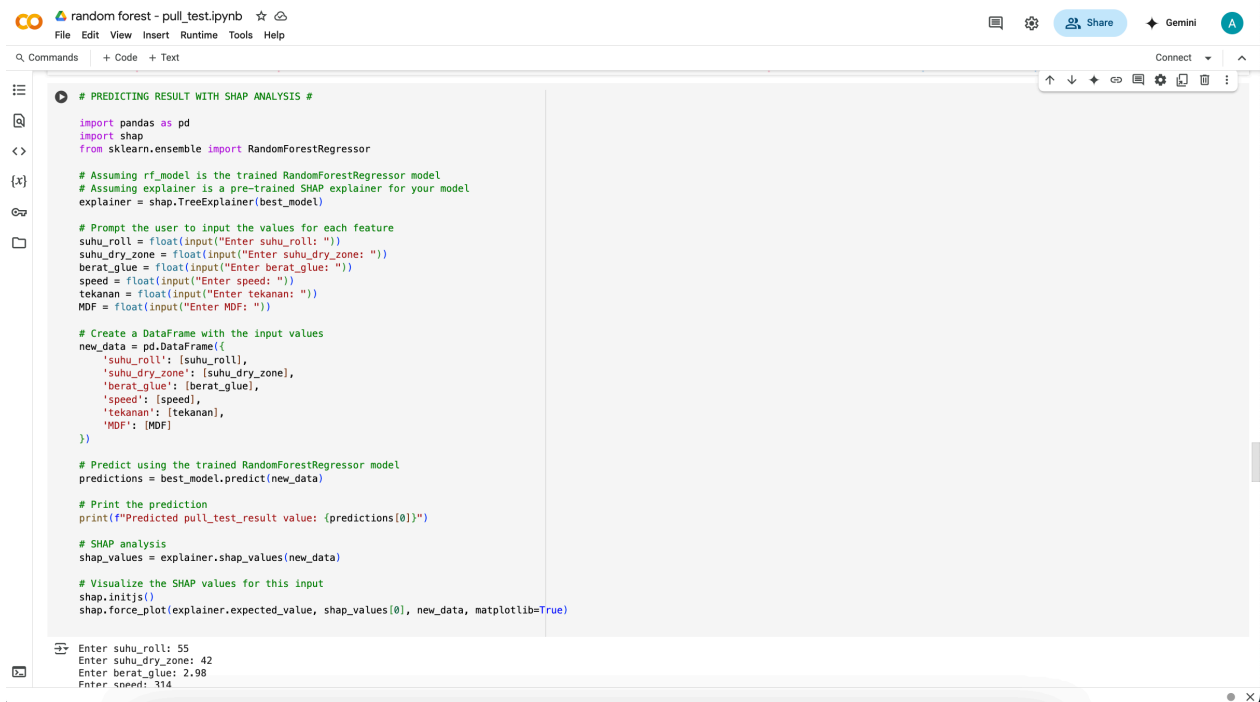
```





SHAP Analysis





```

# PREDICTING RESULT WITH SHAP ANALYSIS #

import pandas as pd
import shap
from sklearn.ensemble import RandomForestRegressor

# Assuming rf_model is the trained RandomForestRegressor model
# Assuming explainer is a pre-trained SHAP explainer for your model
explainer = shap.TreeExplainer(best_model)

# Prompt the user to input the values for each feature
suhu_roll = float(input("Enter suhu_roll: "))
suhu_dry_zone = float(input("Enter suhu_dry_zone: "))
berat_glue = float(input("Enter berat_glue: "))
speed = float(input("Enter speed: "))
tekanan = float(input("Enter tekanan: "))
MDF = float(input("Enter MDF: "))

# Create a DataFrame with the input values
new_data = pd.DataFrame({
    'suhu_roll': [suhu_roll],
    'suhu_dry_zone': [suhu_dry_zone],
    'berat_glue': [berat_glue],
    'speed': [speed],
    'tekanan': [tekanan],
    'MDF': [MDF]
})

# Predict using the trained RandomForestRegressor model
predictions = best_model.predict(new_data)

# Print the prediction
print(f"Predicted pull_test_result value: {predictions[0]}")

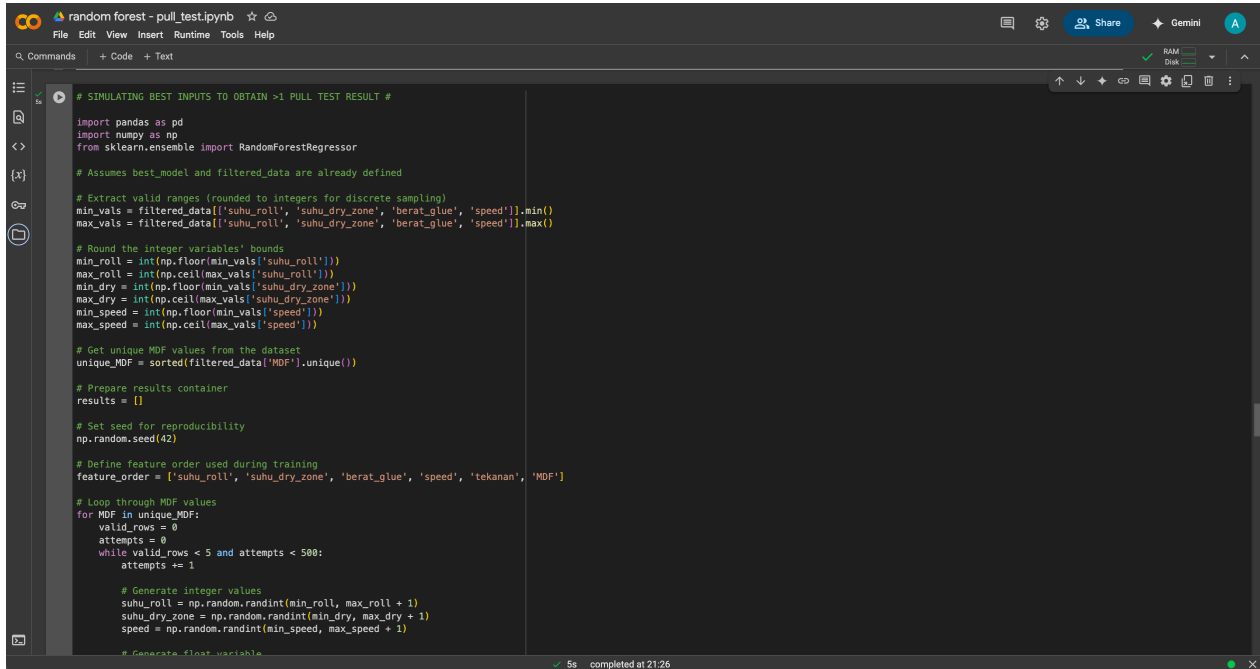
# SHAP analysis
shap_values = explainer.shap_values(new_data)

# Visualize the SHAP values for this input
shap.initjs()
shap.force_plot(explainer.expected_value, shap_values[0], new_data, matplotlib=True)

Enter suhu_roll: 55
Enter suhu_dry_zone: 42
Enter berat_glue: 2.98
Enter speed: 314

```

System recommendations



```

# SIMULATING BEST INPUTS TO OBTAIN >1 PULL TEST RESULT #

import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestRegressor

# Assumes best_model and filtered_data are already defined

# Extract valid ranges (rounded to integers for discrete sampling)
min_vals = filtered_data[['suhu_roll', 'suhu_dry_zone', 'berat_glue', 'speed']].min()
max_vals = filtered_data[['suhu_roll', 'suhu_dry_zone', 'berat_glue', 'speed']].max()

# Round the integer variables' bounds
min_roll = int(np.floor(min_vals['suhu_roll']))
max_roll = int(np.ceil(max_vals['suhu_roll']))
min_dry = int(np.floor(min_vals['suhu_dry_zone']))
max_dry = int(np.ceil(max_vals['suhu_dry_zone']))
min_speed = int(np.floor(min_vals['speed']))
max_speed = int(np.ceil(max_vals['speed']))

# Get unique MDF values from the dataset
unique_MDF = sorted(filtered_data['MDF'].unique())

# Prepare results container
results = []

# Set seed for reproducibility
np.random.seed(42)

# Define feature order used during training
feature_order = ['suhu_roll', 'suhu_dry_zone', 'berat_glue', 'speed', 'tekanan', 'MDF']

# Loop through MDF values
for MDF in unique_MDF:
    valid_rows = 0
    attempts = 0
    while valid_rows < 5 and attempts < 500:
        attempts += 1

        # Generate integer values
        suhu_roll = np.random.randint(min_roll, max_roll + 1)
        suhu_dry_zone = np.random.randint(min_dry, max_dry + 1)
        speed = np.random.randint(min_speed, max_speed + 1)

        # Generate float variable

```

```

random forest - pull_test.ipynb
File Edit View Insert Runtime Tools Help

# Generate float variable
suhu_dry_zone = np.random.randint(min_dry, max_dry + 1)
speed = np.random.randint(min_speed, max_speed + 1)

# Generate float variable
berat_glue = np.random.uniform(min_vals['berat_glue'], max_vals['berat_glue'])

# Calculate tekanan from MDF
tekanan = ((MDF - 1) + (MDF - 1) + (MDF - 2)) / 3

# Construct sample
sample = pd.DataFrame({
    'suhu_roll': [suhu_roll],
    'suhu_dry_zone': [suhu_dry_zone],
    'berat_glue': [berat_glue],
    'speed': [speed],
    'MDF': [MDF],
    'tekanan': [tekanan]
})

# Reorder columns
sample = sample[feature_order]

# Predict
prediction = best_model.predict(sample)[0]

if prediction > 1:
    sample['predicted_pull_test_result'] = prediction
    results.append(sample)
    valid_rows += 1

# Combine all results
final_results = pd.concat(results, ignore_index=True)

# Show output
print(final_results)

# Save to Excel
final_results.to_excel("valid_combinations_pull_test_gt.1.xlsx", index=False)

24      46      38      5.777419      529      16.666667      18.0
25      30      0.345565      320      31.666667      33.0
26      44      48      4.252278      265      31.666667      33.0
27      36      30      5.442075      445      31.666667      33.0
28      38      5.092277      212      31.666667      33.0
29      41      0.666795      217      31.666667      33.0
30      56      59      4.000880      400      NaN      NaN
31      45      37      5.225723      398      NaN      NaN

```

5s completed at 21:26

```

random forest - pull_test.ipynb
File Edit View Insert Runtime Tools Help

suhu_roll  suhu_dry_zone  berat_glue  speed  tekanan  MDF  \
0          40           49      1.330312    533      1.666667      3.0
1          41           58      1.178503    285      1.666667      3.0
2          52           52      2.856943    515      1.666667      3.0
3          54           33      4.234182    544      1.666667      3.0
4          55           50      4.311263    442      1.666667      3.0
5          39           31      5.806891    376      7.666667      9.0
6          34           41      3.219527    498      7.666667      9.0
7          45           54      3.219628    233      7.666667      9.0
8          43           57      2.342813    372      7.666667      9.0
9          48           59      2.431927    235      7.666667      9.0
10         53           54      5.074770    315      10.666667      12.0
11         40           50      0.675061    513      10.666667      12.0
12         37           54      3.432800    500      10.666667      12.0
13         59           38      0.855615    530      10.666667      12.0
14         53           57      0.505340    448      10.666667      12.0
15         47           46      3.982061    234      13.666667      15.0
16         35           59      3.341041    230      13.666667      15.0
17         55           58      4.977046    375      13.666667      15.0
18         45           31      5.267869    386      13.666667      15.0
19         47           45      0.804806    455      13.666667      15.0
20         56           57      2.115793    374      16.666667      18.0
21         49           42      3.302196    522      16.666667      18.0
22         57           55      1.869973    401      16.666667      18.0
23         61           42      1.954074    225      16.666667      18.0
24         46           30      5.777419    529      16.666667      18.0
25         57           30      0.345565    320      31.666667      33.0
26         44           48      4.252278    265      31.666667      33.0
27         36           30      5.442075    445      31.666667      33.0
28         59           38      5.092277    212      31.666667      33.0
29         41           41      0.666795    217      31.666667      33.0
30         56           59      4.000880    400      NaN      NaN
31         45           37      5.225723    398      NaN      NaN
32         34           32      5.693426    285      NaN      NaN
33         36           30      3.421659    189      NaN      NaN
34         47           36      3.208326    457      NaN      NaN
35         48           55      2.746728    226      23.666667      25.0
36         52           36      2.054960    425      23.666667      25.0
37         37           59      2.508469    415      23.666667      25.0
38         44           58      5.533695    220      23.666667      25.0
39         40           56      1.207360    427      23.666667      25.0

predicted_pull_test_result
0          1.209924
1          1.228474
2          1.260530
3          1.285573
4          1.238046
5          1.304820

```

5s completed at 21:26