DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

A DECISION SUPPORT SYSTEM SOFTWARE FOR INTELLIGENT MANUFACTURING

by
İlknur AYDINOĞLU

February, 2023
İZMİR

A DECISION SUPPORT SYSTEM SOFTWARE FOR INTELLIGENT MANUFACTURING

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University In Partial Fulfillment of the Requirements for the Degree of Master of Computer Science

by
İlknur AYDINOĞLU

February, 2023
İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"A DECISION SUPPORT SYSTEM SOFTWARE FOR INTELLIGENT MANUFACTURING"** completed by **İLKNUR AYDINOĞLU** under supervision of **PROF.DR. EMEL KURUOĞLU KANDEMİR** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

.....

Prof. Dr. Emel KURUOĞLU KANDEMİR

Supervisor

.....

Prof. Dr. Çağın KANDEMİR ÇAVAŞ

(Jury Member)

.....

Prof. Dr. Deniz KILINÇ

(Jury Member)

Prof. Dr. Okan FISTIKOĞLU

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGMENT

I would like to express my sincere gratitude to Prof. Dr. Emel KURUOĞLU KANDEMİR for her unwavering support and invaluable guidance throughout the duration of this project. Her invaluable expertise and mentorship have been instrumental in providing me with the necessary knowledge and understanding to effectively analyze and implement the details of this project. I am truly appreciative of the time and effort she has invested in me.

İlknur AYDINOĞLU

A DECISION SUPPORT SYSTEM SOFTWARE FOR INTELLIGENT MANUFACTURING

ABSTRACT

In the fast-paced and complex world of production, production companies that have a data infrastructure with the realization of the internet of things (IoT) and industry 4.0 concepts have started to prefer methods that can make the most benefit from data. Today, manufacturing companies benefit from decision support systems (DSS) in many areas of manufacturing. DSS plays a critical role in smart manufacturing environments by providing relevant information and analysis tools to help managers and other decision makers make better decisions. DSS provides identify bottlenecks in the production process, optimize resource utilization and improve production planning and predictive maintenance facilities. The goal of this study is to identify breakdowns machine parts and schedule maintenance at appropriate times through the use of a DSS and predictive maintenance techniques. Azure predictive maintenance dataset are used for this study. There are sensor data, failure records, maintenance record from a hundred machines in this dataset. A software has been developed to predict breakdown due to machine component failure. At the stage of system and model implementation and design, four continuous, four categorical, one discrete and one class features data sets consisting of approximately one million instances are used for this study. as a consequence, the system that was developed accurately predicts when machine components will fail.

Keywords: Industry 4.0, intelligent manufacturing, machine learning, decision support system, support vector machine, logistic regression, random forest decision tree

AKILLI ÜRETİM İÇİN KARAR DESTEK SİSTEMİ

ÖZ

Üretimin hızlı tempolu ve karmaşık dünyasında, nesnelerin interneti ve endüstri 4.0 konseptlerinin uygulanması ile birlikte veri altyapısı oluşan üretim şirketleri veriden en fazla fayda sağlayabilecek yöntemleri tercih etmeye başlamışlardır. Günümüzde üretim şirketleri, üretimin birçok alanında karar destek sistemlerinden (KDS) yararlanmaktadır. KDS, yöneticilerin ve diğer karar vericilerin daha iyi kararlar almasına yardımcı olmak için ilgili bilgileri ve analiz araçlarını sağlayarak akıllı üretim ortamlarında kritik bir rol oynar.. KDS, üretim sürecindeki darboğazları belirlemeyi, kaynak kullanımını optimize etmeyi ve üretim planlama ile kestirimci bakım olanaklarını iyileştirmeyi sağlar. Bu çalışmada, karar destek sistemi ve kestirimci bakım yöntemleri kullanılarak arızaya neden olan makine bileşenlerinin bulunması ve uygun zamanda bakımlarının yapılması amaçlanmaktadır. Bu çalışma için Azure kestirimci bakım veri kümesi kullanılmıştır. Bu veri setinde yüz makineye ait sensör verileri, arıza kayıtları, bakım kayıtları bulunmaktadır. Makine bileşenlerinin arıza yapma durumunu tahmin eden bir sistem geliştirilmiştir. Sistemin ve modelin tasarımı sırasında yaklaşık bir milyon örnekten oluşan dört sürekli, dört kategorik, bir ayrık ve bir sınıf özellikli veri seti kullanılmıştır. Geliştirilen sistem sonucunda, makine bileşenlerinin arıza zamanları yüksek doğruluk oranıyla tahmin edilmiştir.

Anahtar kelimeler: Endüstri 4.0, akıllı üretim, makine öğrenmesi, karar destek sistemi, destek vektör makinası, lojistik regresyon, rastgele orman karar destek algoritması

CONTENTS

Page

THESIS EXAMINATION RESULT FORM ii
ACKNOWLEDGEMENTS iii
ABSTRACT iv
ÖZ v
LIST OF FIGURES ix
LIST OF TABLES xi
ABBREVIATIONS xii

CHAPTER 1 – INTRODUCTION 1

1.1 Background Information 2
1.2 Problem Definition 2
1.3 Motivation/Related Works 3
1.4 Goal/Contribution 3
1.5 Project Scope 4
1.6 Methodology/Tools/Libraries 4
1.7 Literature Review 6

CHAPTER 2 – INTELLIGENT/SMART MANUFACTURING 8

2.1 Internet Of Things (IoT) 9
2.2 Industry 4.0 11
2.3 Lean Smart Maintenance 12
2.3.1 Breakdown Maintenance 12
2.3.2 Preventive Maintenance 13
2.3.3 Condition-Based Maintenance (CBM) 13
2.3.4 Predictive Maintenance 14

CHAPTER 3 – DECISION SUPPORT SYSTEM.....17

3.1 Decision Support Techniques.....17
3.2 Data Science.....19
3.2.1 Business Understanding.....20
3.2.2 Data Understanding.....21
3.2.3 Exploratory Data Analysis.....22
3.2.4 Data Preparation and Cleaning.....23
3.2.5 Model Building and Training.....24
3.2.5.1 Model Selection.....24
3.2.5.2 Hyper Parameter and Performance Metrics.....25
3.2.5.3 Training.....27
3.2.6 Model Evaluation.....28
3.2.7 Model Deployment.....29
3.3 Machine Learning Algorithms.....30
3.3.1 Support Vector Machine (SVM).....31
3.3.2 Logistic Regression.....33
3.3.3 Random Forest Decision Tree.....35

CHAPTER 4 – REQUIREMENTS/REQUIREMENT ENGINEERING..... 38

4.1 Feasibility Study 38
4.2 Requirement Gathering..... 39
4.3 Decision Support System Requirements..... 39
4.3.1 Data Science Study Requirements..... 39
4.3.2 Software Design Requirements 40

CHAPTER 5 – IMPLEMENTATION 42

5.1 Data Science Case Study 42
5.1.1 Business Understanding..... 42

5.1.2 Data Understanding 43
5.1.3 Exploratory Data Analysis 46
5.1.3.1 Descriptive Analysis 47
5.1.3.2 Normality Analysis 50
5.1.3.3 Correlation Analysis 52
5.1.4 Data Preparation/Cleaning 54
5.1.4.1 Merging and Labelling 54
5.1.4.2 Feature Selection 55
5.1.4.3 Encoding 58
5.1.5 Model Building and Training 60
5.1.5.1 Model Selection 60
5.1.5.2 Training 61
5.1.5.3 Model Results 63
5.1.6 Deployment 67
5.2 Software Design And Development 68
5.2.1 Architectural View 68
5.2.2 Class Diagram 69
5.2.3 Use Case Diagram 70
5.2.4 Activity Diagram 70
5.2.5 Sequence Diagram 71
5.2.6 Web Site Software Development 72
5.2.7 Web Site User Interface (UI) 75

CHAPTER 6 – CONCLUSION 77

REFERENCES 78

LIST OF FIGURES

Page

Figure 2.1 Example of the IoT environment for production.....10

Figure 2.1 Industry steps.....11

Figure 2.3 Breakdown type.....12

Figure 3.1 Data science steps.....20

Figure 3.2 Supervised learning flow.....31

Figure 3.3 Support vector machine algorithm logic.....31

Figure 3.4 SVM algorithm.....33

Figure 3.5 Sigmoid function.....34

Figure 3.6 Decision tree component.....35

Figure 3.7 Random forest algorithm flow.....36

Figure 3.8 Random forest algorithm code.....36

Figure 5.1 Machine age chart.....48

Figure 5.2 Machine model chart.....48

Figure 5.3 Machine component chart.....49

Figure 5.4 Machine failure chart.....49

Figure 5.5 Continuous attributes histogram.....50

Figure 5.6 Q-Q plot.....51

Figure 5.7 Correlation matrix.....53

Figure 5.8 Single dataset with labelling.....55

Figure 5.9 Simple k-best ANOVA scoring result.....56

Figure 5.10 Logistic regression training logic.....62

Figure 5.11 Sıppot vector machine training logic.....62

LIST OF FIGURES

Page

Figure 5.12 Random forest training logic.....63

Figure 5.13 Logistic regression confusion matrix.....64

Figure 5.14 Weighted logistic regression confusion matrix.....65

Figure 5.15 SVM with polynominal kernel function.....66

Figure 5.16 Random forest decision tree confusion matrix.....66

Figure 5.17 Model deployment.....67

Figure 5.18 Architectural view.....68

Figure 5.19 Class diagram.....69

Figure 5.20 Use case diagram.....70

Figure 5.21 Activity diagram.....71

Figure 5.22 Sequence diagram.....72

Figure 5.23 SQL creating store procedure query.....73

Figure 5.24 .NET Core get data from SQL store procedure function.....74

Figure 5.25 Highcharts sensor data visualization code.....75

Figure 5.26 Home page.....76

Figure 5.27 Sensor data page.....76

LIST OF TABLES

Page

Table 4.1 Data science study requirements.....40

Table 4.2 Software design requirements.....41

Table 5.1 Dataset files explanation.....43

Table 5.2 Example of machine sensor data.....44

Table 5.3 Example of failure data.....44

Table 5.4 Example of error data.....45

Table 5.5 Example of machine features.....45

Table 5.6 All datasets attribute types.....46

Table 5.7 Descriptive analysis.....47

Table 5.8 Shapiro-wilk results.....51

Table 5.9 Pearson correlation significance (p-values)..... 53

Table 5.10 Single dataset.....54

Table 5.11 Contingency example.....57

Table 5.12 Chi-squared result.....57

Table 5.13 Encoding result.....59

Table 5.14 Dataset information.....59

Table 5.15 Logistic regression models result.....64

Table 5.16 SVM models result.....65

Table 5.17 Final model results.....67

ABBREVIATIONS

AUC	:	Area Under Curve
DSS	:	Decision Support System
IoT	:	Internet of Things
AI	:	Artificial Intelligence
IDE	:	Integrated Development Environment
MVC	:	Model View Controller
LSM	:	Lean Smart Maintenance
PLC	:	Programmable Logic Controller
CBM	:	Condition Based Maintenance
EDA	:	Exploratory Data Analysis
ROC	:	Receiver Operating Characteristic
SVM	:	Support Vector Machine
DSS	:	Decison Support System
UML	:	Unified Modelling Language
ORM	:	Onject Relational Mapping
HTML	:	Hypertext markup language
JSON	:	Javascript Object Notation
ML	:	Machine Learning
TPR	:	True Positive Rate
FPR	:	False Positive Rate
RBF	:	Radial Basis Function
DOM	:	Document Object Model

CHAPTER ONE
INTRODUCTION

Manufacturing companies or other companies follow various strategies to ensure their sustainability. While some create innovations in the right planning areas, some companies try various methods in the field of maintenance. Data becomes a key tool at this point. Now, companies save many of their data in order to see their future and can make evaluations with these data. New concepts started to be mentioned with the beginning of saving the data. The most important of these are internet of things (IoT), industry 4.0 and intelligent manufacturing. These concepts advance in technological areas, making data available in appropriate environments and providing insight. Looking at the big picture, companies need to catch up with the industry 4.0 revolution and develop sustainability strategies that fit this concept.In this situation companies may choose more than one strategy. For example: by creating decision support systems, efficient information can be provided to employees or the system can automatically make and implement decisions without the need for employees. It is major requirement to have knowledge of some concepts to understand this transformation in companies..

First of all, since the internet of things applied production companies, progress has been made in data collection and these technologies. The evolution of the internet led to the standardization of the Internet of Things, which in turn facilitated the emergence of Industry 4.0. The term smart manufacturing began to be mentioned with the implementation of these concepts in production and the approach and strategy of companies in production was determined by using machine learning (ML) or other methods.

In this study, a predictive maintenance dataset has been selected considering this concepts. Predictive maintenance dataset consists of data such as sensor data and fault records. Warning system, taking automatic action or helping maintenance workers by establishing a decision support system are carried out through predictive maintenance datasets according to the workflow in manufacturing.

1.1 Background Information

In order to understand the scope of the study, it is necessary to have knowledge about some concepts. The study involves understanding smart manufacturing and decision support systems to create a decision-making system.

Smart manufacturing utilizes cutting-edge technologies such as the Internet of Things, artificial intelligence, and data analytics to optimize manufacturing processes and enhance efficiency, flexibility, and output. It is an integral component of Industry 4.0, which represents the fourth industrial revolution characterized by the integration of advanced technologies. The IoT provide to enable technology for intelligent manufacturing, as it allows manufacturers to collect data from various sources, such as sensors, machines and products, to gain insights into the performance and efficiency of their processes. The IoT enables real time data collection and analytics, which can be used to optimize production, reduce waste, improve quality and increase responsiveness to changing customer demands.

The concepts of smart manufacturing, industry 4.0 and IoT are mentioned in detail in the next chapters.

1.2 Problem Definition

The problem to be addressed in this thesis is the maintenance processes in production plants. Every production plant has a maintenance strategy. While some production plants intervene a breakdown, some production plants take action to prevent or predict breakdowns. Intervention of a breakdown causes a lot of production downtime, shortening the life of the mechanical parts, more work of the personnel and many other negative situations. This situation has become a problem for production plants and there has been an increase in the number of factories that have started preventive or predictive activities recently.

The Azure predictive maintenance dataset that is a publicly available dataset provided by Microsoft Azure and is designed to support machine learning and predictive maintenance applications was examined within the scope of predictive activity. The problem for the dataset is the disruption of quality and maintenance processes due to the failure of a machine component.

1.3 Motivation/Related Works

One of most important motivations when starting this project is to find a solution to a production problem and to use data while for this solution. Because companies started to collect data with industry 4.0 and they are working to use the data correctly. Predictive maintenance and predictive quality concepts are at the forefront of studies using data in manufacturing companies. There are many companies that have started to work in these fields. Business models that provide services to other companies by developing predictive models are also increasing today. These types of companies can quickly produce models for predictive maintenance or offer solutions by generalizing some problems. It can even be said that predictive maintenance works are competitive landscape. In this regard, Gartner firm (a technology consultancy firm based in Stamfort Connecticut) examined the companies working in the field of predictive maintenance and talked about the companies. This study analyzed the relevant research on companies that offer predictive maintenance solutions. The success rates, which methods are used and for which datasets effective solutions are produced are examined.

1.4 Goal/Contribution

The aim of this study is to develop a predictive maintenance solution in production using machine learning algorithms. It is aimed to ensure efficiency in production, increase productivity, reduce downtime, determine the maintenance strategy and evaluate machine learning algorithms academically for this problem with this solution.

1.5 Project Scope

The scope of the project has been determined as the building of the predictive model and creating of a platform for the use of this model. In order to find the successful model, ML algorithms used for predictive maintenance problems were examined. Models were build using support vector machine, random forest and logistic regression algorithms.

1.6 Tool/Library/Methodology

A decision support system is being created to aid users in making decisions by providing them with insights and information gained from analyzing data. The system consists of two phases: a data science step and a web site implementation.

In the data science phase, Google Colab has been chosen as the platform for running data science Python notebooks, exploratory data analysis and experimenting with machine learning models. Cloud SQL has also been selected as the database technology to store and manage the data.

As for the web site implementation phase, C# language has been selected..NET has been chosen as the framework for building the web application. .NET is a popular and powerful framework that provides a comprehensive set of tools and libraries for building web, mobile, desktop and cloud applications. It supports a wide range of programming languages and allows developers to build robust and scalable applications with ease.

To develop the web application using .NET, Visual Studio has been selected as the integrated development environment (IDE). Visual Studio is a popular and feature-rich IDE that provides a comprehensive set of tools and features for building .NET applications. It offers a modern and intuitive interface, code completion, debugging, testing and deployment capabilities, among other features.

In addition to .NET and Visual Studio, HTML has been chosen as the markup language for creating the structure and layout of the website and JavaScript has been selected for adding interactivity and enhancing the user experience.

A lot of libraries are used for data analysis and model building. The python libraries that used are explained.

NumPy is used to work with large, multi-dimensional arrays and matrices of numerical data.

Pandas handles data structures and offers data analysis tools. It offers functionality for reading and writing data, handling missing values, merging and joining data sets and manipulating data in various ways.

Matplotlib is utilized to create static, animated and interactive visualizations. It has functions that can produce various types of plots like line plots, scatter plots, bar plots and histograms.

Seaborn is employed for creating statistical visualizations. It builds on top of Matplotlib, providing a high-level interface to create beautiful and informative statistical plots.

Scikit-learn has been chosen as a library for machine learning and data mining tasks. It provides functions for training and evaluating models, preprocessing data and performing various types of analysis.

For web application development, the following libraries of the .NET framework have been chosen: ASP.NET Core, Entity Framework, ASP.NET MVC, ASP.NET Web API. These libraries provide a comprehensive set of tools and libraries for building web applications and APIs, working with data in a database, organizing the code of a web application, creating HTTP-based APIs and building real-time web applications.

For the web site interface, bootstrap and jquery have been chosen as the libraries to be used. Bootstrap is a library for building responsive and mobile-first websites and jquery is a library for simplifying the process of working with the DOM. These libraries have been selected to provide the necessary tools and functions for creating an attractive and user-friendly web site interface.

1.7 Literature Review

Many DSS and machine learning studies have been carried out for manufacturing processes.

Furniture companies often face the challenge of identifying high-quality wood for their products, as well as a lack of knowledge about the specific characteristics and specifications of good wood. To address these issues, Roimasro Sibagariang and et al. (Sibagariang, & Riandari, 2019) conducted a study to determine the best wood for use in the furniture industry. The study employed the bayesian method, a statistical approach that involves using prior knowledge and data to make predictions about future events. The results of the study showed that the DSS developed using the bayesian method was successful in identifying the best wood for use in furniture production. This system aim potentially help furniture companies improve the quality and reliability of their products, as well as streamline their production processes by providing a reliable means of selecting wood.

Denny Trias Utomo et al. (Utomo, & Pratikto, 2020) developed a web based DSS for the supply chain management area, with the aim of creating a system that could help manufacturing companies select the best suppliers. Choosing suppliers for the manufacturing industry can be a complex task, as there are many criteria to consider. To address this challenge, the team used the Fuzzy Analytical Hierarchy Process (FAHP) method to identify the most suitable suppliers. One of the innovative aspects of this study was the development of an DSS web site, which allows manufacturers to easily access and utilize the system from any location with an internet connection. The use of the FAHP method and the creation of this web-based system are aimed

manufacturing companies streamline their supplier selection process and improve the efficiency and effectiveness of their supply chain management.

Foivos Psarommatis et al. (Psarommatis, & Kiritsis, 2021) developed two decision support systems to help address quality issues that may arise during the manufacturing process. One of these systems was data-driven, while the other was based on knowledge and expertise. By combining these two approaches, the team was able to create a hybrid solution that effectively addressed quality problems and improved productivity.This hybrid system was based on the Manufacturing Semantics Ontology (MASON), which was used to extract knowledge from manufacturing data. Using SPARQL queries, the team was able to develop a rule-based hybrid model that could identify and address quality issues in production. With the development of these decision support systems, it is aimed to help manufacturing companies improve the efficiency and reliability of their production processes by enabling them to quickly identify and resolve quality issues.

Buathong et al. (Buathong, & Vilasdaechanont, 2020) developed a decision support system that uses machine learning and optimization techniques to help address the challenge of selecting machines for use in the manufacturing process. The system comprises of different parts, such as data input, data transformation, a mathematical model for optimization, and presenting results. The optimization model is aimed at minimizing production costs by selecting the most suitable machines for the manufacturing process. Through this study, the team was able to create a DSS that was validated by a manufacturing company. The development of this DSS has the potential to enhance the efficiency and effectiveness of production processes in the manufacturing industry.

CHAPTER TWO
INTELLIGENT/SMART MANUFACTURING

Manufacturing systems are modernized and made more intelligent.. Smart manufacturing, also known as intelligent manufacturing, involves the use of advanced information and manufacturing technologies to optimize production and improve product transactions. It is a comprehensive approach to manufacturing that aims to maximize efficiency and effectiveness (Kusiak,1990). Smart manufacturing utilizes various technologies, such as sensors, decision-making models, advanced materials, intelligent devices and data analytics, to facilitate the entire product life cycle (Li et al.,2017). Smart manufacturing can enhance production efficiency, product quality and service levels (Davis et al.,2012).

Intelligent manufacturing encompasses the integration of various technologies such as the IoT, Industry 4.0, AI/ML and advanced robotics to create a more flexible and responsive manufacturing system.

Industry 4.0 represents the next step in the evolution of manufacturing, where advanced technologies like IoT, AI, and robotics play a central role in the production process. The integration of these technologies enables new levels of automation, data analysis, and connectivity. The goal of Industry 4.0 is to create a more flexible, efficient and responsive manufacturing system.

The internet of things (IoT) includes to the network of physical objects (e.g., machines, vehicles and buildings) that are embedded with sensors, software and other technologies, allowing them to collect and exchange data.

In the context of smart manufacturing, the maintenance process refers to the activities that are carried out to ensure that the equipment and machinery used in the manufacturing process are operating at peak performance. This includes activities such as regular inspections and maintenance, as well as the identification and correction of any issues that may arise.

Smart manufacturing can improve the maintenance process in several ways. For example, the use of the IoT allows for the real-time monitoring of equipment and machinery, enabling manufacturers to detect and address any issues before they become more serious. This can help to prevent unplanned downtime and improve the overall reliability of the manufacturing process. Additionally, the use of AI and ML can help to improve the maintenance process by enabling manufacturers to identify patterns and trends in the data collected from their equipment and machinery. This can help to predict when maintenance is likely to be needed, allowing manufacturers to schedule maintenance activities in advance and avoid potential issues.

Maintenance strategies have begun to be data-driven. Using data in maintenance strategies help manufacturers to improve the efficiency and effectiveness of their maintenance activities. This situation provides to create of the concept of lean smart maintenance in smart manufacturing.

Lean smart maintenance is a method of maintenance that combines the principles of lean manufacturing with the use of advanced technology and data analysis. Lean manufacturing is a method of production that focuses on maximizing value and minimizing waste and it has been widely adopted by manufacturers as a way to improve efficiency and reduce costs. The most popular subject of the lean smart maintenance concept is predictive maintenance.

In this chapter, the concepts of IoT, industry 4.0 and predictive maintenance under the lean smart maintenance are explained covered by smart manufacturing.

2.1 Internet of Things (IoT)

The concept of the IoT was introduced by Kevin Ashton in 1999, when he proposed the use of RFID technology for property management (Lombardi et al.,2021). The practical birth of the IoT is thought to be 2008. The companies started to talk about the huge IoT market with the increase in IoT devices and the technological developments since then and investments of company (Lombardi et al.,2021).

The Internet of Things (IoT) involves various functions, including device modeling, control, data collection, publishing and detection. (Villamil et al.,2020). The device mentioned here may vary depending on the place of use. While sensors, robots and programmable logic controllers devices are used in the production sector, electrocardiograph devices can be mentioned in the health sector. Different definitions can be mentioned since there is more than one point of view to IoT. One of them is defined as a computing network that can exchange data without human intervention, has device traceability and consists of digital-analog machines (Kuzlu et al.,2021). If the concept of IoT is considered specifically for production facilities, currently the most important technology used for monitoring production activities, providing transparency and operational efficiency, obtaining and sharing information to follow the production plan and finally making decisions (Jagtap et al,2021).

Today, many companies have IoT infrastructure. In this regard, they invest in devices such as sensors, robots and machines. In order for all these devices to be IoT devices, they must have processes such as data reception and communication. The factory has robots, sensors, industrial cameras and programmable logic controllers (PLC). Some of them can communicate with each other, some with the server. This creates the IoT environment. An IoT environment consisting of camera, robot, PLC and sensors with server communication is shown in Figure 2.1.

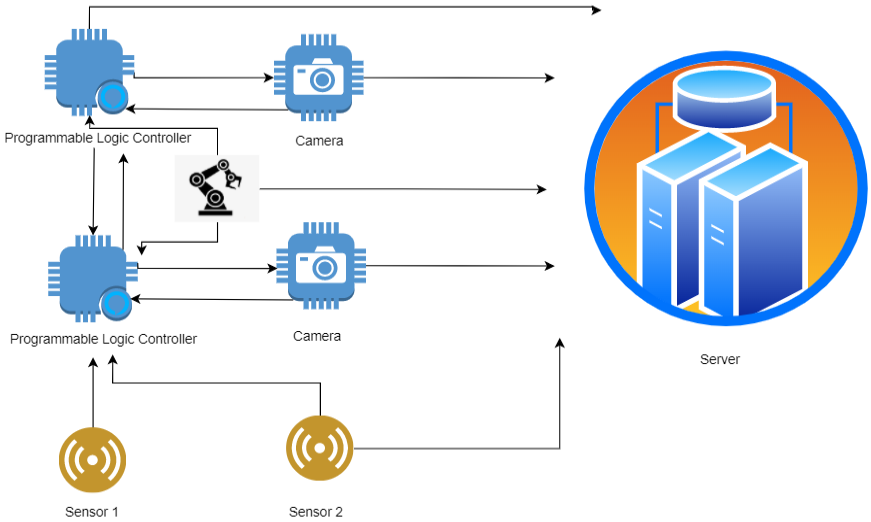

Figure 2.1 Example of the IoT environment for production

2.2 Industry 4.0

Industry 4.0 also referred to as the fourth industrial revolution, was first introduced at the Hannover Fair in Germany in 2011. In 2012, a working group including Robert Bosch GmbH and Henning Kagermann submitted a proposal for this new era of industrial development to the German government (Dalenogare et al.,2018). Industry 4.0 provides to a modernized, connected form of manufacturing that integrates information technology and industry to improve efficiency and performance. It contains the integration of all types processes and the connectivity of products. The main objective of Industry 4.0 is to enhance and optimize industrial production by using advanced technologies and automation (Kolberg, & Zühlke,2015). Although some people are at least skeptical of this new approach, industry 4.0 technologies are particularly beneficial for the serial manufacturing company (Zhong et al.,2017).

Industry 4.0 is the latest phase in the development of industrial production, preceded by Industry 1.0, 2.0, and 3.0. It utilizes the IoT to create a more advanced and interconnected industrial system. The steps involved in Industry 4.0 are depicted in Figure 2.2.

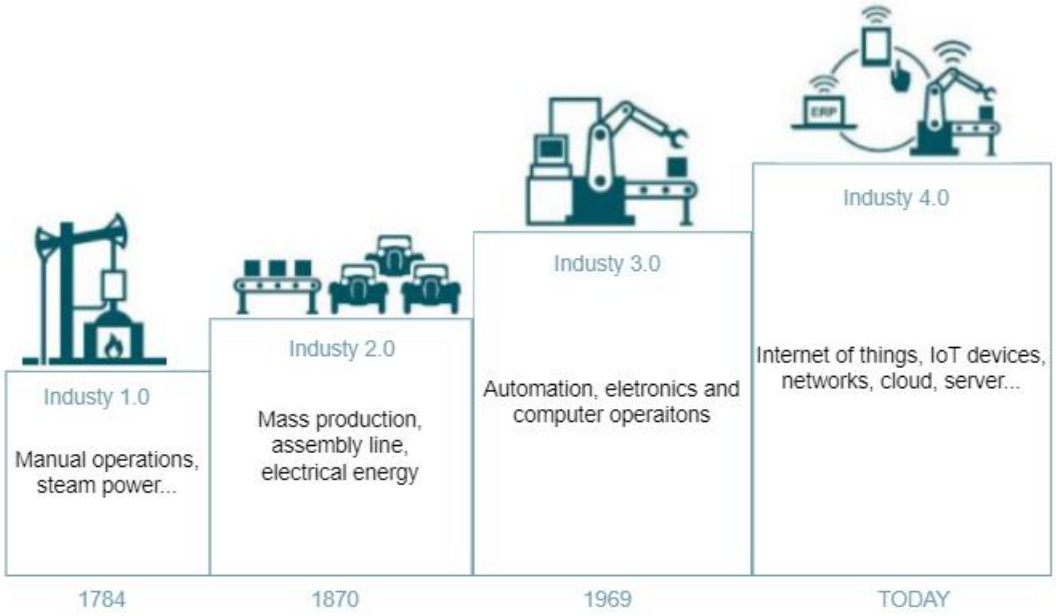

Figure 2.2 Industry step

2.3 Lean Smart Maintenance

Under the concept of smart manufacturing, maintenance processes can be considered. Lean Smart Maintenance (LSM) is a comprehensive maintenance strategy that aims to optimize the efficiency and effectiveness of asset management. LSM combines the principles of lean methodology and continuous improvement with the use of technology and data analytics to reduce costs, improve safety, and increase the reliability and availability of assets. This approach focuses on eliminating waste, maximizing value, and involving all stakeholders in the maintenance process (Biederman, & Kinz,2019). Lean smart maintenance is the stage companies want to reach. But before that, there are various maintenance types. These types are breakdown, preventive condition-based and predictive. In Figure 2.3, the breakdown types are shown. Efficiency and productivity increase while going to the lean smart maintenance step.

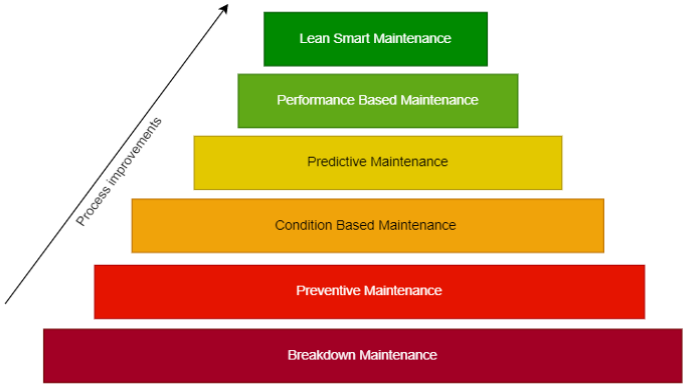

Figure 2.3 Breakdown types

2.3.1 Breakdown Maintenance

Breakdown maintenance is the situation in which it is fixed at that moment in case of breakdown (Flavio Trojan, & Rui, 2017). This approach is typically used when the cost of regular preventative maintenance is seen as too high or when the downtime associated with preventative maintenance is not feasible. The downside to this

approach is that it can lead to unexpected equipment failures and unplanned downtime, which can be costly and disrupt production.

2.3.2 Preventive Maintenance

Preventive maintenance involves taking proactive measures to prevent equipment failure, such as repairing or replacing components, in order to avoid breakdowns (Flavio Trojan, & Rui, 2017). This method is used to extend the life of equipment and reduce the need for costly repairs. It can also help to improve the performance and reliability of equipment, which can increase productivity and reduce costs. Some examples of preventive maintenance activities include cleaning and lubricating equipment, tightening bolts and other fasteners and replacing worn or damaged parts.

2.3.3 Condition-based Maintenance (CBM)

Condition based maintenance is the determination of the appropriate conditions of a machine and creating a rule-based maintenance procedure accordingly (Flavio Trojan, & Rui, 2017). CBM allows for a more proactive and cost-effective approach to maintenance, as it enables maintenance to be performed only when it is needed, rather than on a fixed schedule. CBM provides the use of sensors and other monitoring devices that are attached to equipment or machinery. These devices can measure various factors such as temperature, vibration and other parameters that can indicate the health and condition of the equipment. The data collected by these sensors is then analyzed using specialized software and algorithms to identify potential issues and determine when maintenance is needed.

CBM advantage is that it allows for more targeted and efficient maintenance activities. For example, instead of performing the same maintenance tasks on all equipment on a fixed schedule, CBM can identify which specific parts or components of a machine are likely to fail and maintenance can be performed only on those parts. This can help to reduce the amount of time and resources spent on maintenance and it can also help to avoid unnecessary repairs and downtime.

Another advantage of CBM is that it can help to improve the reliability and performance of equipment. By identifying potential issues before they become more serious, CBM can help to prevent equipment failures and downtime, which can increase productivity and reduce costs.

One example of CBM in action is the use of sensors and other monitoring devices to monitor the condition of machinery in a manufacturing plant. The sensors can be attached to key components of the machinery, such as bearings, gears and other parts and they can collect data on factors such as temperature, vibration and other parameters. The data collected by these sensors is then analyzed using specialized software and algorithms to identify potential issues and predict when maintenance is needed. This can help to improve the efficiency and performance of the machinery and it can also help to prevent unexpected failures and downtime.

2.3.4 Predictive Maintenance

Predictive maintenance involves predicting equipment failures and determining the optimal time to perform repairs or maintenance in order to prevent disruptions (Flavio Trojan, & Rui, 2017). This approach is similar to condition-based maintenance (CBM), but it uses more advanced techniques such as machine learning and artificial intelligence to make more accurate predictions about the health and condition of equipment.

Identifying the equipment or machinery that will be included in the predictive maintenance program is the first step. This may include critical or high-value assets, as well as equipment that is prone to frequent failures or downtime.

Collecting data on the equipment or machinery is second step. This may involve installing sensors and other monitoring devices to collect data on factors such as temperature, vibration, pressure and other parameters that can indicate the health and condition of the equipment. In this step, it is necessary to know the general flow of maintenance.

When a machine breakdown occurs, there are certain criteria that can be used to diagnose the problem and determine the appropriate course of action. There are some criterias that are often used in machine breakdown control include. Vibration is a common indicator of potential problems in machinery. Changes in vibration levels or sudden spikes in vibration can indicate issues such as imbalanced or misaligned parts, worn or damaged bearings or other problems that can impact the effectiveness or efficiency of the equipment. Temperature is another important criterion that can be used to diagnose machine breakdowns. High or low temperatures can indicate potential issues such as overheating or other problems that can lead to equipment failure or downtime. Noise is another common criterion that can be used to diagnose machine breakdowns. Changes in noise levels or sudden spikes in noise can indicate potential issues such as misaligned or worn parts that can affect the performance of the equipment. The condition of the oil or lubricant used in machinery can also be a useful criterion for diagnosing machine breakdowns (Wakiru et al.,2020). Changes in the color, consistency or other characteristics of the oil or lubricant can indicate potential issues such as contamination that can affect the health of the equipment. According to these criteria, appropriate data collection systems are developed.

Analyzing the data is third step. Once data has been collected, it needs to be analyzed using specialized software and algorithms to identify patterns and trends that can indicate potential issues or failures. This may involve using techniques such as machine learning and artificial intelligence to make more accurate predictions.

Developing a maintenance plan is fourth step. Based on the analysis of the data, a maintenance plan can be developed that outlines the specific tasks and activities that need to be performed to prevent equipment failures and downtime. This may include tasks such as cleaning, lubricating, tightening or replacing parts and it may be based on a schedule or on specific triggers or thresholds that are identified through the data analysis.

Implementing the maintenance plan is fifth step. Once the maintenance plan has been developed, it needs to be implemented according to the schedule or triggers that

have been identified. This may involve performing the maintenance tasks onsite or it may involve scheduling repairs or other interventions by maintenance personnel.

Monitoring and evaluating the results is the last step. The effectiveness of the predictive maintenance program should be monitored and evaluated over time to ensure that it is providing the desired benefits. This may involve tracking metrics such as equipment downtime, repair costs and other indicators of performance and reliability. Any issues or challenges that are identified should be addressed and addressed to improve the effectiveness of the program.

Considering the predictive maintenance steps, solutions are developed using various methods. There are several different methods such as condition monitoring, predictive modeling and expert systems that can be used in predictive maintenance and the most appropriate method will depend on the specific equipment or machinery being monitored, as well as the types of issues that are being targeted

CHAPTER THREE
DECISION SUPPORT SYSTEM

A decision support system is an information system that helps support and improve decision making (Vasquez et al.,2021). A framework for designing decision support systems (DSS) includes three main components: people, technology and development approach. As with other systems, a structured approach is necessary for the successful implementation of a DSS (Sprague, & Carlson, 1982). The four stages of the early framework for decision support systems are: intelligence, design, choice and implementation. Intelligence involves identifying the need for a decision. Design involves generating and evaluating potential solutions. Choice involves selecting the best course of action. Implementation involves implementing the chosen solution in the decision-making context.

DSSs can be used in a wide range of industries and applications, including finance, healthcare, manufacturing and marketing. They can help decision makers to make better decisions by providing them with access to relevant data, by applying analytical models to the data to identify patterns and trends and by presenting the results in a clear and concise manner.

DSSs can be used to support a wide range of decision making tasks, including strategic planning, operational decision making and tactical decision making. They can help decision makers to make better decisions by providing them with the information and tools they need to understand the implications of different choices, to evaluate the risks and opportunities associated with different options and to select the best course of action.

3.1 Decision Support Techniques

There are many techniques for decision support systems. Statistical and mathematical modeling such as regression analysis and optimization algorithms to analyze data and identifying trends and patterns, knowledge management such as expert systems, case-based reasoning to capture and reusing the expertise of domain

experts, easy representation of data with understand in an interactive way and collaborative tools such as visualization and data mining such as GIS and dashboards, group decision support systems and electronic meetings to support group decision making and communication are between DSS techniques.

Considering all these techniques, it can be said that the most common DSS techniques are data science and ML concepts in big picture. Data science and machine learning are important tools for decision support systems, as they can help to analyze large amounts of data and make predictions or recommendations based on that data. Decision support systems use algorithms and other analytical techniques to support the decision-making process by providing information and insights that are relevant to a particular problem or situation.

Data science and machine learning are related fields, but they are not the same thing. Data science is a broad term that encompasses a range of techniques and processes for extracting knowledge and insights from data. This can include everything from data cleaning and exploration to advanced statistical and machine learning methods. Data scientists may use a variety of tools and techniques, including machine learning algorithms, to analyze and interpret data. On the other hand, ML is a subfield of AI that focuses on developing algorithms and models that enable computers to learn and make predictions or decisions from data without explicit instructions (Alzubi et al.,2018).

Data science is often used in DSS, which are designed to help decision makers make more informed and effective decisions. Data science techniques, such as machine learning and predictive modeling, can be used to analyze large and complex datasets, identify patterns and trends and generate insights that can be used to support decision making. For example, a DSS might use data science to analyze sales data, customer data and market data to predict future demand for a product, identify potential customers or optimize pricing strategies. A healthcare decision support system might use data science to analyze patient data, medical records and clinical trials to predict patient outcomes, recommend treatments or identify potential risks.

Data science is an important part of many decision support systems, as it allows these systems to process and analyze large amounts of data, extract valuable insights and support better decision making.

Machine learning allows algorithms to learn from data and make predictions or recommendations based on that learning. This can help decision makers to make more informed decisions, by providing them with insights and information that they might not have been able to obtain on their own. There are many different types of machine learning algorithms that can be used in decision support systems, depending on the specific problem or situation. For example, supervised learning algorithms can be trained on labeled data to make predictions about future events, while unsupervised learning is given an unlabelled dataset and it needs to find the hidden structure or pattern in it

3.2 Data Science

Data science is an interdisciplinary field that uses techniques from statistics, computer science, and domain expertise to extract insights and knowledge from data (Provos, & Fawcett,2013). It covers a wide range of activities, including data cleaning and preparation, data mining, machine learning and visualization.

The purpose of data science is to extract insights and information from data that can be used to make better decisions, solve problems or develop new products or services. Data science is a rapidly growing field as organizations across a wide range of industries understand the value of data-driven decision making.

There are stages of realizing data science projects. The stages involved in data science can vary depending on the specific problem or opportunity being addressed, but generally, the process involves several key steps. These are business understanding, data acquisition, data understanding, data preparation/cleaning, exploratory data analysis, model building and training, model evaluation and deployment (Shukla et al.,2022). In figure 3.1, the data science stages are shown.

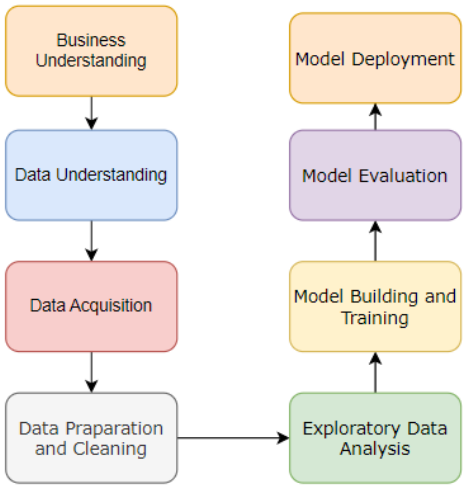

Figure 3.1 Data science stages

3.2.1 Business Understanding

Business understanding is a key step in the data science process. It involves gaining an understanding of the business context in which the data science project is being undertaken, including the problem or opportunity that the project is intended to address and the goals and objectives of the project.

Some specific activities that may be involved in the business understanding phase of a data science project include:

- Identifying the problem or opportunity that the project is intended to address, including the specific business question or challenge that the project aims to solve.
- Determining the objectives and requirements for the project, including any specific performance metrics or success criteria.
- Defining the scope of the project, including any constraints or assumptions that will affect the analysis.
- Gathering information about the business context, including any relevant data, stakeholders or external factors that may impact the project.

The business understanding phase aims to guarantee that the data science project aligns with the objectives and requirements of the company, and to establish a clear and straightforward plan for executing the project. By thoroughly understanding the business context and objectives of the project, data scientists can develop a focused and effective approach to the data analysis and ensure that the results of the project are useful and relevant to the business.

3.2.2 Data Understanding

Data Understanding data is the second step in the data science process. It involves understanding the data available for analysis, including variables, values and relationships between different data items.

A few topics that may be involved in the data understanding phase of a data science project in below:

- Detailed understanding of data attributes, variables and determining their adequacy for solving the problem
- Exploring data to gain insight into its overall nature, quality and relevance to the problem or opportunity being addressed
- Identifying potential biases or sources of error in data that could affect analysis
- Developing a plan for how data will be used in the later stages of the data science process

Generally, the purpose of the data understanding phase is to provide a solid foundation for the rest of the data science process. With a comprehensive understanding of data, data scientists can make informed decisions about how to proceed with the analysis and avoid potential pitfalls or mistakes that can arise from a lack of understanding (Dhar,2013).

3.2.3 Exploratory Data Analysis

Exploratory data analysis (EDA) is a type of analysis that is used to summarize, visualize and explore a dataset. EDA aim is to identify patterns, trends and relationships in the data and to gain a better understanding of the data and the underlying processes that generated it (Vigni et al.,2013).

EDA typically involves a range of techniques, including visualizing the data using plots and graphs, summarizing the data using descriptive statistics and applying more advanced techniques such as clustering and regression analysis.

EDA provides to guide the selection of appropriate modeling and analysis techniques. It is also a valuable tool for communicating the results of data analysis to others, as it can help to clearly and concisely present the findings of the analysis.

The general analyzes made within the scope of the EDA study are given below.

- Summarize data using descriptive statistics such as mean, median and mode
- Visualize data using plots and charts such as histograms, scatter plots and box plots
- Identifying trends, patterns and anomalies in data
- Testing relationships between different variables, such as using correlation analysis or hypothesis testing
- Confirming assumptions or hypotheses about the data, such as normality or covariance testing
- Develop a deeper understanding of the data and its impact on the problem or opportunity being addressed

EDA provides a better understanding of the data and its characteristics, in order to inform the subsequent steps in the data science process. By thoroughly exploring and summarizing the data, data scientists can gain valuable insights and can identify potential issues or opportunities that may impact the analysis.

3.2.4 Data Preparation/Cleaning

Data preparation and cleaning is the process of organizing and formatting data so that it can be used in analysis and modeling. This involves a range of tasks, including identifying and handling missing values, detecting and removing outliers and ensuring that the data is in the correct format for the specific analysis or modeling technique being used (Alasadi, & Bhaya, 2017).

Data preparation and cleaning is a crucial step in the data science process. Data preparation contains transforming the data, such as by combining or aggregating data from multiple sources or creating new variables based on existing data, validating the cleaned and prepared data to ensure it is ready for analysis.

There are several stages involved in data preparation and cleaning:

- *Data acquisition:* This is the process of collecting and importing data from various sources.
- *Data exploration*: This is the process of exploring the data to understand its structure and content. This may involve visualizing the data using various techniques, such as histograms or scatter plots, to understand its distribution and identify any trends or patterns.
- *Data transformation:* This is the process of transforming the data into the format required for analysis or modeling. This may involve merging data from multiple sources or converting data from one format to another.
- *Data cleaning:* This is the process of identifying and handling missing values, outliers and other data quality issues. This may involve imputing missing values, removing outliers or applying other techniques to ensure that the data is of high quality.
- *Data validation:* This is the process of verifying that the data has been prepared and cleaned correctly. This may involve performing various checks, such as comparing the data to known benchmarks or running statistical tests, to ensure that the data is ready for analysis and modeling.

Data preparation and cleaning is an iterative process that involves a combination of these stages. It is important to methodically examine every step to guarantee that the data is usable and prepared for examination and implementation of statistical models.

3.2.5 Model Building and Training

Model building and training is an important step in the data science process. It involves developing and applying a statistical or machine learning model that can be used to analyze data and extract insights from it.

3.2.5.1 Model Selection

The selection of a model should be based on a careful consideration of the specific problem being addressed, the characteristics of the data and the available resources. It is important to choose a model that is well-suited to the problem at hand and that can be trained accurately and efficiently (Vartak et al.,2016). There are some keypoints to consider when choosing a model.

The nature of the problem being addressed different types of models are suitable for different types of problems. For example, linear regression is suitable for predicting a continuous outcome, while a decision tree is better suited for classification problems.

The complexity of the data which is more complex data may require more sophisticated models, such as neural networks, to accurately capture the underlying patterns and relationships. On the other hand, simpler data may be better suited for simpler models, such as linear regression.

The utilization of deep learning models, for example, necessitates vast amounts of data for efficient training. In case of limited data, it may be needed to opt for simpler models that can perform well even with less data.

Considering the amount of time and computational resources available, some models, such as deep learning models, can be computationally intensive to train and may require significant amounts of time and computing power. If time and resources are limited, it may be necessary to use simpler models that can be trained more quickly.

3.2.5.2 Hyperparameters and Performance Metrics

Configure the model, including adjusting required hyperparameters and determining appropriate performance metrics. When building a model, it's often necessary to adjust the hyperparameters to optimize its performance. Identify the key hyperparameters that are likely to impact the model's performance (Yang, & Shami, 2020). This will vary depending on the type of model you're using, but common hyperparameters include the learning rate, regularization strength and the size of the model.

There are many other types of hyperparameters and the specific hyperparameters that are available will depend on the type of model being used. A few types of hyperparameters include:

- *Learning rate:* This is a hyperparameter that controls the step size that is used to update the model's weights during training. Typically, the learning rate is a small positive number, such as 0.1 or 0.01. A higher learning rate means that the model's parameters will be updated more quickly, allowing the model to learn from the data more quickly. However, a higher learning rate can also make the model more unstable and prone to overfitting. On the other hand, a lower learning rate means that the model's parameters will be updated more slowly, allowing the model to converge on a more accurate solution. However, a lower learning rate can also make the training process slower and less efficient.
- *Regularization strength:* This is a hyperparameter that controls the amount of regularization applied to the model's weights. Regularization is a technique that is used to prevent overfitting and a higher regularization strength can help the model to generalize better to new data.

- *Hidden layer size:* This is a hyperparameter that controls the number of hidden layer neuron counts. A larger hidden layer size can increase the model's capacity and improve its performance, but can also increase the risk of overfitting.
- *Number of layers:* This is a hyperparameter that controls the number of layers in a neural network. A deeper network can improve the model's performance, but can also be more difficult to train.

Performance metrics is important for model selection. There are many different performance metrics that can be used for model building and the appropriate metric will depend on the specific goals of the model and the type of data it is working with. Some common performance metrics include:

- *Accuracy:* This measures the proportion of correct predictions made by the model. It is often used for balanced dataset. The disadvantage of this metric is that it ignores incorrect estimates.
- *Precision and Recall:* These metrics are commonly used in conjunction with each other and they measure the model's ability to make correct predictions. Precision measures the fraction of predictions that are correct, while recall measures the fraction of actual positive examples that were correctly predicted by the model (Torgo, & Riberio, 2009). Precision and recall are often used together, because they provide complementary information about a model's performance. Precision is useful for evaluating the model's ability to make correct predictions, while recall is useful for evaluating the model's ability to find all of the positive examples in the data. Some common cases in which precision and recall might be used as performance metrics include evaluating the performance of a model on a classification task, such as spam detection or medical diagnosis.
- *F1 Score:* This is a measure of a model's accuracy that considers both the precision and recall of the model. It is calculated as the harmonic mean of precision and recall, where the best value is 1.0 and the worst value is 0.0. The F1 score is often used as a metric for classification problems in which the

distribution of class labels is imbalanced. Some common cases in which F1 score might be used as a performance metric include valuating the performance of a model on a classification task, such as image recognition or text classification.

- *ROC AUC:* Receiver Operating Characteristic (ROC) Area Under the Curve (AUC) is a performance metric for binary classification problems. It is a graphical representation of the model's ability to distinguish between the positive and negative classes, and it is represented by a curve. The ROC curve plots the true positive rate (TPR) against the false positive rate (FPR) at different classification thresholds (Wang, & Zhou, 2015). A few cases in which ROC AUC might be used as a performance metric include evaluating the performance of a model on a binary classification task, such as spam detection or fraud detection, comparing the performance of different models on a binary classification task, to see which model is more effective at distinguishing between positive and negative examples.
- *Root Mean Squared Error (RMSE):* This is is a measure of difference between predicted values and actual values. It calculates the average deviation of predictions from the true values. The calculation is done by taking the square root of the average of the squared differences between the predicted and actual values. RMSE is often used in conjunction with other performance metrics, such as mean absolute error or R-squared, to provide a more complete picture of the model's performance. Some common cases in which RMSE might be used as a performance metric include evaluating the performance of a model on a regression task, such as predicting house prices or stock prices.

3.2.5.3 Training

The process of model training involves using a set of data to fine-tune the model's parameters, so it can make accurate predictions. This is usually achieved through the use of optimization algorithms such as gradient descent. During this phase, the selection of the models, evaluation metrics and the tuning of the model's settings (hyperparameters) are decided, then the model is trained using these parameters.

In general, the goal of the model building and training phase is to develop a model that can accurately and reliably extract insights from data.

3.2.6 Model Evaluation

Evaluating the performance of a model is an important step in the model development process and can help to identify areas for improvement and ensure that the model is meeting its intended goals.

Defining the evaluation metric is the first step in evaluating a model is to choose an appropriate performance metric that reflects the goals of the model and the characteristics of the data. This could be a metric like accuracy, precision, recall or F1 score for classification tasks or a metric like RMSE or R-squared for regression tasks.

The next step is to divide the data into three distinct subsets that includes training, validation, and test after selecting the evaluation metric. The training dataset is used to adjust the model's parameters, the validation set is used to optimize the model's hyperparameters, and the test set is utilized to evaluate the model's final performance.

The following step after dividing the data is to use the training dataset and the chosen hyperparameters to train the model. This generally involves repeatedly adjusting the model's parameters to minimize the chosen cost function.

Evaluating the model's performance on the validation set is the model has been trained, the next step is to evaluate its performance on the validation set. This can be done by making predictions on the validation set and calculating the chosen performance metric.

Fine-tunning the model's hyperparameters is the one of final steps. If the model's performance on the validation set is not satisfactory, the next step is to adjust the hyperparameters and retrain the model. This process can be repeated multiple times until the model's performance on the validation set is acceptable (Raschka, 2018).

The last step in the process is to assess the model's performance on the test dataset once the model's hyperparameters have been optimized. This gives an objective assessment of the model's ability to handle new data, and can be used to compare the performance of different models.

3.2.7 Model Deployment

Model deployment is the process of making a trained model available for use in a production environment, where it can be used to make predictions on new data. This is an important step in the data science process, as it allows the model to be used to generate business value or solve real-world problems.

There are several steps involved in deploying a model. These are packaging, testing, deploying in production, monitoring and maintaining.

Packaging the model is the first step in deploying a model is to package it in a format that can be easily deployed and used in a production environment. This typically involves saving the trained model in a file or database and providing any necessary supporting files or libraries.

Testing the model in a staging environment is the model has been packaged, the next step is to test it in a staging environment that closely resembles the production environment. This can help to identify any potential issues or bugs before the model is deployed to production.

Deploying the model to production is after the model has been tested in a staging environment, the next step is to deploy it to production. This typically involves setting up the infrastructure and processes needed to serve the model's predictions to users or applications.

After deploying the model to real-world usage, it is crucial to keep an eye on its performance and make adjustments as needed. This process includes retraining the

model with fresh data, fine-tuning the model's hyperparameters, or deploying an updated version of the model.

3.3 Machine Learning Algorithms

Machine learning is a branch of computational algorithm that emulate the human brain and decision-making structure using by the examples that is in environment (El Naga, & Murphy, 2018).

Machine learning has many terms. **The terms attribute, feature, predictor or field** define the independent variable for machine learning (Bi et al.,2019). **The term domain** is the set of possible values of the attributes used for machine learning (can be categorical, numeric, binary) (Bi et al.,2019). **The terms input and output** refers dependent and independent variables. **The terms classifiers or predictors** are used generally in the machine learning literature to explain to algorithms that implement a prediction or classification (Bi et al.,2019). **The term loss function** represents the predictive performance of a machine learning model (Bi et al.,2019).

Considering all these machine learning terms, it can be said that there are a lot of different algorithms for solving data problems. Because there is no one-size-fits-all algorithm for solving a problem. Algorithm selection depends on the type of problem, the number of variables and the type of variables (Manesh, 2020).

Machine learning algorithms can be grouped as supervised, unsupervised and semi-supervised. **Supervised learning** is a method of machine learning that involves training a model on a labeled dataset of input-output pairs, in order to produce predictions for new, unseen data. The model is trained by learning from the input-output pairs in the dataset and can then be used to make predictions on new data (Manesh, 2020). First of all, using the labelled dataset, the model is trained with the train set and the model results are observed with the test set.

As shown in Figure 3.2, the workflow for supervised learning is presented. The dataset is divided into two parts, the training set which is used to train the model, and the testing set which is used to evaluate the model's effectiveness. After that the model results are examined.

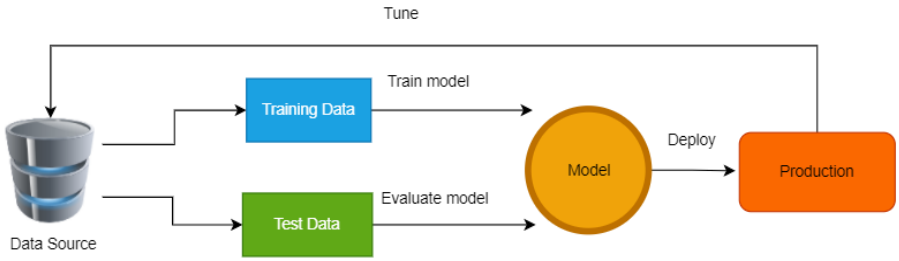

Figure 3.2 Supervised learning flow

SVM, random forest decision tree and logistic regression algorithms, which are examples of supervised algorithms, will be explained in here.

3.3.1 Support Vector Machine

The support vector machine (SVM) is a popular algorithm for data classification, first introduced by Vapnik. It has been widely used due to its success in this task (Durgesh, & Lekha, 2010). The reasons for the popularity of the SVM algorithm are its structural risk minimization, scaling in big data, flexibility due to parameterizability and accuracy (Pavlidis et al., 2004).

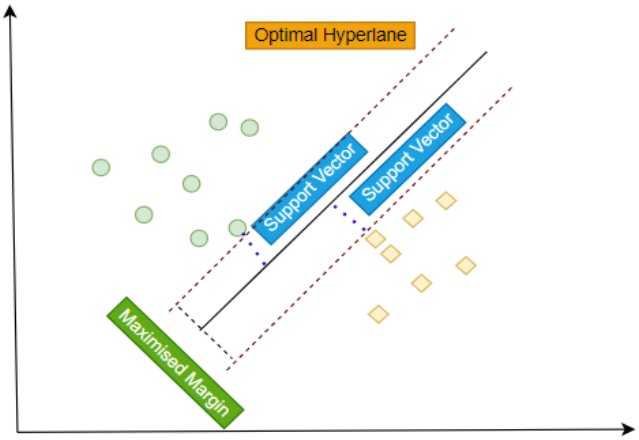

Figure 3.3 Support vector machine algoritm logic

In Figure 3.3, the working logic of the support vector machine algorithm is shown graphically. Firstly, SVM creates a hyperplane between classification points. Support vectors are the data points that are closest to the hyperplane in a support vector machine (SVM). They are used to define the boundary between different classes in the dataset. The distance between these vectors is calculated and it is aimed to maximize this distance.

An SVM algorithm uses a kernel function to map the data into a higher-dimensional space, where it can be linearly separable. The kernel function is a mathematical function that takes the data as input and transforms it into a higher-dimensional space. There are several different types of kernel functions that can be used and the choice of kernel function will depend on the specific characteristics of the data. Some of kernel functions are linear kernel, polynomial kernel and radial basis function (RBF) kernel. The linear kernel is a basic kernel function and it is employed when the data can be separated by a straight line in the original space. The linear kernel projects the data onto a higher-dimensional space using a linear transformation and it can be written as a dot product of the input vectors. The polynomial kernel is a more complex kernel function that can be used when the data is not linearly separable in the original space. The polynomial kernel projects the data onto a higher-dimensional space using a polynomial transformation and it can be written as a polynomial function of the dot product of the input vectors.The RBF kernel is a widely-used kernel function that can be used in a variety of applications. The RBF kernel projects the data onto a higher-dimensional space using a radial basis function and it can be written as the exponential of the squared Euclidean distance between the input vectors.

Once the kernel function has been chosen and the data has been mapped into a higher-dimensional space, the SVM algorithm uses an optimization process to find the maximum-margin hyperplane that best separates the classes in the data. This hyperplane is then used to make predictions on new data.

The training phase of the support vector machine algorithm is as Figure 3.4 (Connelly,2020).

```
Input: D=[X,Y]; X= array of input with m features, Y=array of class label (array(C))
Output: Performance of system
function: TrainSvm(X,Y,numberOfRuns)
    initiliaze:learningRate=Math.Random();
    for learningRate in numberOfRuns
    error=0
    for i in X
        if(Y[i]*(X[i]*w))<1 then
        update: w=w+learningRate*((Y[i]*X[i])*(-2*(1/numberOfRuns)*w))
        else
        update: w=w+learningRate*(- 2*(1/numberOfRuns)*w)
```

Figure 3.4 SVM algorithm

3.3.2 Logistic Regression

Before explaining logistic regression, it is necessary to explain the concept of regression. There are dependent and independent variables in the regression concept and the aim is to estimate the value of the dependent variable by looking at the independent variables that have a relationship between dependent variable (Hilbe,2011).

Logistic regression is between the most used methods in research. Unlike normal regression, it is used when the dependent variable values are binary (Harimoorthy, & Thangavelu, 2021). The dependent variable can have two values such as 1-0, true-false, sick-not sick, ok-not ok.

To train as logistic regression model, an optimization process is used to find the optimal values of the model weights and bias that minimize the loss function. The loss function measures the difference between the predicted probabilities and the true class labels and it is typically defined using the cross-entropy loss function.

Once the model has been trained, it can be used to make predictions on new data. Given an input, the logistic regression model calculates the linear combination of the input features using the trained weights and bias, applies the sigmoid function to the result and returns the resulting probability. This probability can then be used to classify the input as belonging to a specific class.

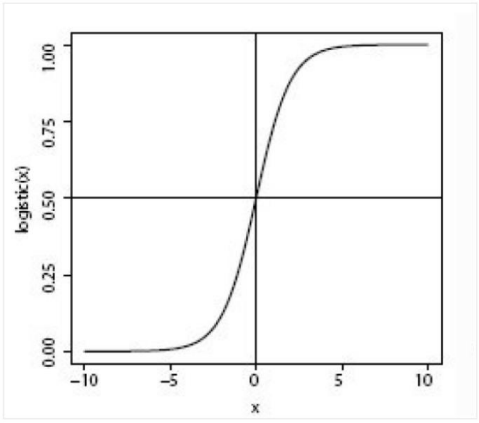

Figure 3.5 Sigmoid function

The sigmoid function is shown in Figure 3.5. For a better understanding of the sigmoid function, the sigmoid formula is explained with an example (Ito, 1991).

$$h(z)=1/(1+e^{-z}) \quad (1.1)$$

The h(z) function shows the probability that the dependent variable is positive considering the independent variables. For example, if the h(z) value is 0.85 in the data set, which is not diabetes, it means that the patient has diabetes. 1-h(z) represents a negative situation and 0.15 indicates the probability of not having diabetes. H(z) must be between 0 and 1. The z is an input and the e shows euler number.

As a result, the h(z) function result, which is, the rules h(z)>=0.5 and h(z)<0.5 are taken into consideration. In this case, it is in the negative class when it is less than 0.5 and in the positive class when it is greater than or equal to 0.5. Then, the success of the model is evaluated by using error or some success metrics (MSE, ROC etc..).

3.3.3 Random Forest Decision Tree

Random forest algorithm is one of the decision trees methods. Decision trees represent various rules created using labeled datasets. These rules are displayed in the form of a tree structure. Decision tree simply consists of leaf node: indicates target feature value, decision node: indicates test feature, edge: defines detach a feature and path: identifies test detach for final decision (Kelleher et al., 2015). Decision and leaf nodes are shown in Figure 3.6.

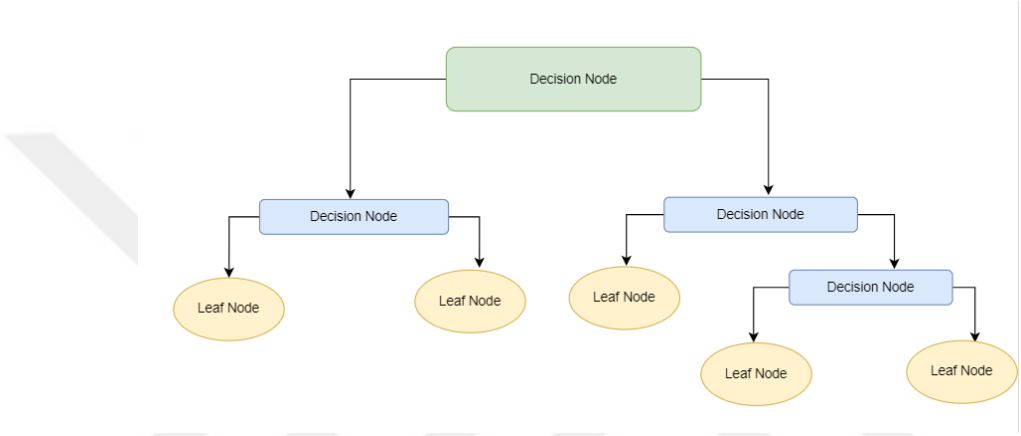

Figure 3.6 Decision tree components

Random forest algorithm creates decision tree models by selecting random forests from the training data and as a result, the decision tree models are matched with the test data. Random forests are fitted to the dataset to measure the importance of a feature in the random forest method. The error is calculated for each data point and averaged over the forest in which it is located during the fitting process. After training, a score is generated for the importance of any feature. If this score is high, that feature is more important than other features (Praiwala, 2015).

One of the advantages of the random forest algorithm is that it keeps the variance low by using the bagging method. Also, the fact that instead of learning the tree, it divides it into random subsets, lowers the correlation. Ensembles are more effective when the individual models that comprise them are uncorrelated.

The logic of the random forest algorithm is shown in Figure 3.7.

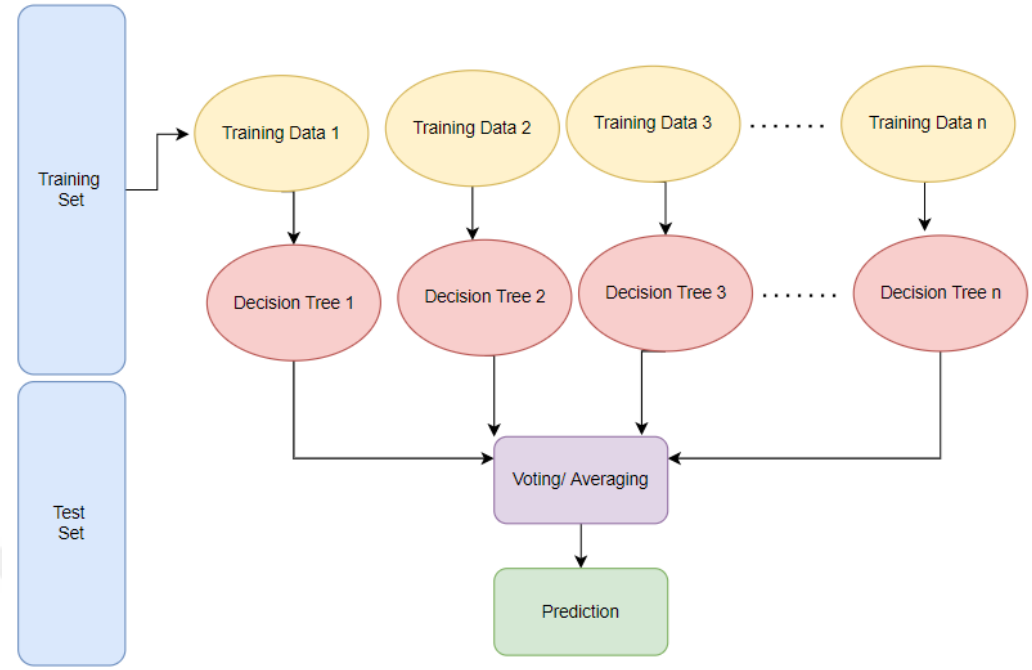

Figure 3.7 Random forest algorithm flow

In a random forest, multiple decision trees are created using different subsets of the data. Each tree in the forest makes predictions based on the data it has been trained on and the final prediction is made by combining the predictions of all the trees in the forest. This approach can help to improve the accuracy and stability of the predictions, as it reduces the influence of any individual tree that may be overfitting to the data. Random forest decision tree algoritms pseudo code steps in Figure 3.8.

```
# Initialize the random forest with a set of trees
forest = []
for i in range(n_trees):
    # Sample a subset of the data with replacement
    sample_X, sample_y = bootstrapped_sample(X, y)

    # Build a decision tree on the subset
    tree = build_tree(sample_X, sample_y)

    # Add the tree to the random forest
    forest.append(tree)

# Make a prediction for a sample by aggregating the predictions of all the trees
def predict(X):
    predictions = []
    for tree in forest:
        prediction = tree.predict(X)
        predictions.append(prediction)
    return most_common(predictions)
```

Figure 3.8 Random Forest Algorithm Code

Random forests are commonly used in a variety of applications, including classification, regression and feature selection. In the context of predictive maintenance, a random forest algorithm could be used to analyze data collected from sensors and other monitoring devices and to make predictions about the time to failure of equipment or machinery. The predictions of the random forest could then be used to schedule maintenance activities and improve the performance and reliability of the equipment.

CHAPTER FOUR
REQUIREMENTS/REQUIREMENT ENGINEERING

Requirement analysis of the project should be done before the implementation phase. It is important for data science and software projects because it helps to ensure that the project is well-defined and that the project team has a clear understanding of the goals and objectives of the project. This helps to avoid misunderstandings and miscommunications later on in the project and it also help to identify potential problems or issues that may need to be addressed before the project gets too far along. In addition, requirement analysis helps to establish a common understanding of the project among all stakeholders, which can help to ensure that the project is completed in a timely and efficient manner.

At this stage, the feasibility of the study, collection and clarification of requirements are mentioned.

4.1 Feasibility Study

Feasibility study is very important for the success of a project. Here, the feasibility study of predictive maintenance is carried out in manufacturing. The main advantage of predictive maintenance is that it can help organizations to avoid costly equipment failures and unplanned downtime. By predicting when equipment is likely to fail, organizations can schedule maintenance and repairs during planned downtime, which can save time and money. Additionally, predictive maintenance can help organizations to improve the reliability and availability of their equipment, which can improve productivity and customer satisfaction. Predictive maintenance provides cost savings, improved equipment reliability and increased productivity.

In addition, a project that combines predictive maintenance and a decision support system enables maintenance personnel to work more easily, to see the breakdown more easily, to arrange maintenance times and to take action.

Considering all these perspectives, it can be said that predictive maintenance with decision support system projects are very beneficial for the manufacturing company. Besides, having all the data describing the problem convenient and sufficient for predictive maintenance makes the project feasible.

4.2 Requirement Gathering

Requirement gathering is the process of collecting, identifying and documenting the requirements for a specific project or system. This helps ensure that the final product or system meets the needs of the user and aligns with the goals of the project.

The main goal is to develop an application that combines data science and software and can give insights to relevant people that can be used in production.

Other expected outputs are to provide acceptable model results, to predict breakdown correctly, user-friendly software design, to maintain the model if it is necessary, to see sensor data in the software and to see all machines condition.

4.3 Decision Support System Requirement

There are two parts of the system in this study. The first is the development of a predictive maintenance model using machine learning algorithms under data science and the second is an interface that will be presented to users as a decision support system software. The requirements for these two parts are explained.

4.3.1 Data Science Study Requirements

The lifecycle of data science studies is used for the data science part of the study and requirements are defined considering this cycle. In addition, the technologies and methods to be used should also be mentioned. In table 4.1, data science study requirements are listed.

Table 4.1. Data science study requirements

ID	Requirements
1	Defining predictive maintenance dataset. (Microsoft azure predictive maintenance datasets are used. These datasets are stored as excel files)
2	Import excel files to MSSQL database. (The database is in the cloud)
3	Implementing data cleaning, aggregation, transformation operations using python.
4	Displaying exploratory data analysis.
5	Preparation of dataset for model building.
6	Converting categorical values to numeric or binary using encoder methods.
7	Considering the failure times and changing parts labelling the instances as “breakdown” or “not”.
8	Normalization of the dataset
9	Training the model and testing
10	Training with different hyperparameters using support vector machine algorithm.
11	Training with different hyperparameters using random forest decision tree algorithm.
12	Training with different hyperparameters using logistic regression algorithm.
13	Selecting the appropriate model based on the selected perfomance metrics.
14	To make the model output ready for use for the decision support system.
15	If it is necessary, retraining the model.

4.3.2 Software Design Requirements

An software is required to present the model created in the data science study to the users. Software lifecycle are generally grouped as analysis, design, development, testing and release. In the analysis phase, the requirements are clarified. In Table 4.2, software design requirements are listed. The software has been developed with these requirements and the requirements cover the design, development and release phases. While developing this software, the requirements are examined in each software development step and it is checked whether the relevant requirements are completed.

Table 4.2. Software design requirements

ID	Requirements
1	Analysis the which technologies will be used.
2	Cloud, MSSQL and .NET technologies researching for web development.
3	Database design.
4	Providing the database connection to serve the data to user interface.
5	Implementing the business codes such as getting machine masurements.
6	Integrating the predictive maintenance model to website.
7	User interface design.
8	Displaying the machine measurements as chart.
9	Listing all machine by showing their current condition.
10	Coloring the machine boxes using predictive maintenance model results.
11	Testing all softwares.
12	Release.

With this website, it is aimed to present the model outputs to the users in a more understandable way. In addition, it is aimed to provide the software with extra features such as seeing the raw data and making their own analysis.

CHAPTER FIVE
IMPLEMENTATION

This study consists of two main parts. The first part is a data science case study for building the model to be used for the decision support system. The second part is the web application developed to present the decision support system to the users.

5.1 Data Science Case Study

Data science processes are different from traditional software development processes. For this reason, a different life cycle method has been adopted in data science. This study is carried out considering the data science lifecycle.

First of all, it starts from the business understanding step. Analyzes have been made to understand the problem in detail and to show that the collected data can solve this problem. Then, the datasets are examined and all features and instances are searched. In order to create an accurate data pipeline for data preparation, the data is saved in the database instead of excel. Data cleaning and merging are implemented by using this database. The distribution of sensor data, balance-unbalance of the data, correlations are examined with exploratory data analysis and the dataset is prepared for the model by looking at the correlation and a few test values. Algorithms suitable for the dataset are examined for the model building part and the results are analyzed considering the success metrics.

5.1.1 Business Understanding

Decision support systems in manufacturing are generally developed through predictive maintenance or predictive quality projects. In the predictive maintenance part, predicting the breakdown of a part or machine is the main issue. With the concept of Industry 4.0 and IoT, the data of a machine in factories can be monitored using sensors (Biswas et al., 2022). Sensor data is an important tool for predictive maintenance.

In this study, voltage, rotation, pressure, vibration data for 100 machines are used. However, these data are not sufficient for the predictive model. The model, age, maintenance times and failure times of the machines are also important. For example: a machine may operate in high vibration due to its age, but this does not mean that it will fail or a machine may change its trend after maintenance and models may produce incorrect results if the maintenance time is not known.

As a result, data from the machines, maintenance times, machine features are used to obtain an accurate model.

5.1.2 Data Understanding

In the data understanding step, data files, attributes, data numbers, data quality, data types are examined. Microsoft Azure predictive maintenance datasets are used for this study. The reason of the choosings this datasets, there are all datasets that describe the problem. For example: vibration, temperature and pressure are important sensor data but it is not enough to describe the all problem. Because the machines have breakdowns and after that the technicians fix or change them. Without these data, predictive models can't predict accurately.

There are 5 excel files for azure predictive maintenance data. In Table 5.1, All files are listed with their number of instances and number of attributes.

Table 5.1. Dataset files explanation

Excel File	Number of instances	Number of attributes
Machine sensor values dataset	876.101	6
Failure history	761	3
Maintenance history	3287	3
Error logs	3920	3
Machine features	100	3

Machine sensor values dataset contains vibration, rotation, voltage, pressure for 100 machines belongs to 2015 year. These values are very important to predict breakdown. In Table 5.2, sensor data example is shown. Machine id shows which has machines values.

Table 5.2. Example of machine sensor data

Datetime	MachineID	Rotation	Vibration	Pressure	Voltage
2015-01-01 06:00:00	1	418.504	45.087	113.077	176.217
2015-01-01 07:00:00	1	402.747	43.413	95.460	162.879
2015-01-01 08:00:00	1	527.349	34.178	75.237	170.989
2015-01-01 09:00:00	1	430.252	41.423	110.248	173.985
2015-01-01 10:00:00	1	425.124	47.785	102.859	168.135

Failure history shows component changing due to failure. This file contains component name, datetime and machine id. In Table 5.3, failure history data is shown. Each machine has component that can be replaced.

Table 5.3. Example of failure history

Datetime	MachineID	Failure component
2015-01-01 06:00:00	1	Com2
2015-03-27 07:00:00	100	Com3
2015-05-12 08:00:00	52	Com5
2015-07-07 09:00:00	1	Com6
2015-09-18 10:00:00	1	Com7

Also, maintenance history has same file format. Maintenance history shows replacement of component for planned maintenance or failure. Components that change due to an failure are also kept in the failure history file.

Error log file is the file where the errors that do not cause failure are collected. It contains datetime, machine id and error id. In Table 5.4, error log file data is shown.

Table 5.4. Example of error history

Datetime	MachineID	Error
2015-02-01 06:00:00	1	Error1
2015-05-27 07:00:00	100	Error3
2015-07-12 08:00:00	52	Error10
2015-09-07 09:00:00	1	Error6
2015-11-18 10:00:00	1	Error9

The machine features file is very important for predictive maintenance. Because generally, when a DSS system is to be designed, if there is no information about machine features, all of them are considered equal. In real-world problems, this is impossible. This dataset contains machine model and machine age. In Table 5.5, machine features examples are shown. Machine id is not a number. It is a categorical value such as machine one, machine two, machine three.

Table 5.5. Example of machine features

Model	MachineID	Age
Model1	1	18
Model2	2	12
Model3	3	11
Model3	4	5
Model5	5	15

As a result, when all files are looked at, there are categorical and continuous data needed for analysis. There are one discrete, four continuous and four categorical attributes in the all files. In Table 5.6, all attributes and their types are listed. Looking at the table, there are a hundred machine with four different model and their age ranges are between 0 and 20. There are approximately 8770 data for each machine. Model one has 16 machines, model two has 17 machines, model three has 35 machines, model

four has 32 machines. All machines have sensor measurements such as vibration, voltage, pressure and rotation. Component represents machine part. One machine contains component one, component two, component three and component four. Each component can cause breakdown.

Table 5.6. All dataset attributes types

Name	Data Type	Explanation
Machine ID	Categorical	A hundred machines
Model	Categorical	Four different model
Age	Discrete	Between [0-20]
Component	Categorical	Four different component
Error	Categorical	Five different error
Rotation	Continuous	-
Voltage	Continuous	-
Vibration	Continuous	-
Pressure	Continuous	-

In this section, the data is examined in general terms. Looking at these data, it can be said that categorical and numeric data are used and labeling should be done for breakdown prediction. In the following sections, data will be explained in more detail with operations such as labeling, data cleaning, exploratory data analysis.

5.1.3 Exploratory Data Analysis

Reseachers or data scientists use a variety of techniques such as histograms, scatter plots and box plots, to better understand the distribution and relationships between the different variables in the dataset. They may also use summary statistics, such as means, medians and standard deviations, to describe the data. In this study, exploratory data analysis is implemented using all datasets of Microsoft Azure predictive maintenance.

Google colab platform is used and selected python programming language for EDA. Google colab platform is a cloud-based platform that facilitates data science studies. Loading the libraries is very successful in managing the execution performance of the algorithms.

5.1.3.1 Descriptive Analysis

Descriptive analysis is a type of statistical analysis that is used to describe and summarize a dataset. It is often used to get a better understanding of the characteristics of a dataset and to identify patterns and trends within the data (Lawless, & Heymann, 2010).

In this study, all datasets are imported to Google colab environment. Descriptive analysis are performed for continuous attributes. In Table 5.7, descriptive analysis results are shown such as min, max, standart deviation.

Table 5.7. Descriptive analysis

	Volt	**Rotate**	**Pressure**	**Vibration**
Count	876100	876100	876100	876100
Std	15.509114	52.67388	11.048679	5.370361
Min	97.333604	138.4320	51.237106	14.877054
25%	160.30492	412.3057	93.498181	36.777299
75%	181.00449	482.1766	107.55523	43.784938
Max	255.12471	695.020984	185.951998	76.791072
Mean	170.780	446.59481	100.8610	40.386154

There are a variety of tools and techniques that can be used for descriptive analysis. Histograms and bar charts are some of these that help to understand the data more detailed. Histogram and bar charts are created for machine features, machine failure history, machine sensor values and machine maintenance history datasets. Bar chart are preferred for categorical and discrete data and histogram are preferred for continuous data for this study.

In Figure 5.1, machine age distribution is shown. According to this figure, there are a lot of old machines. The quantity of 14 years old machines are more than the others.

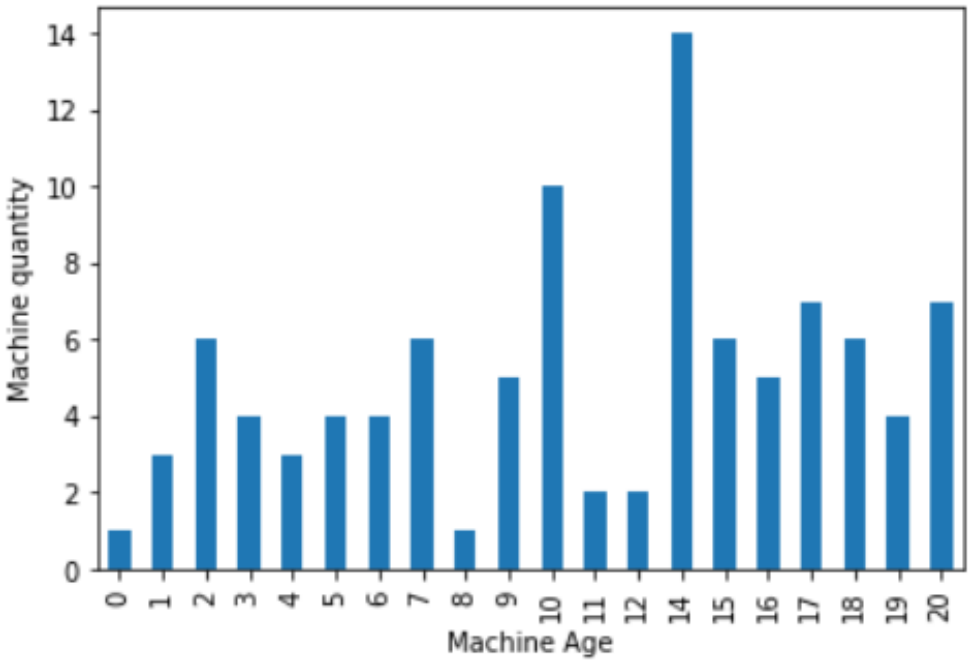

Figure 5.1 Machine age chart

In Figure 5.2, machine model distribution is shown. Model 3 and model 4 machines are more than others.

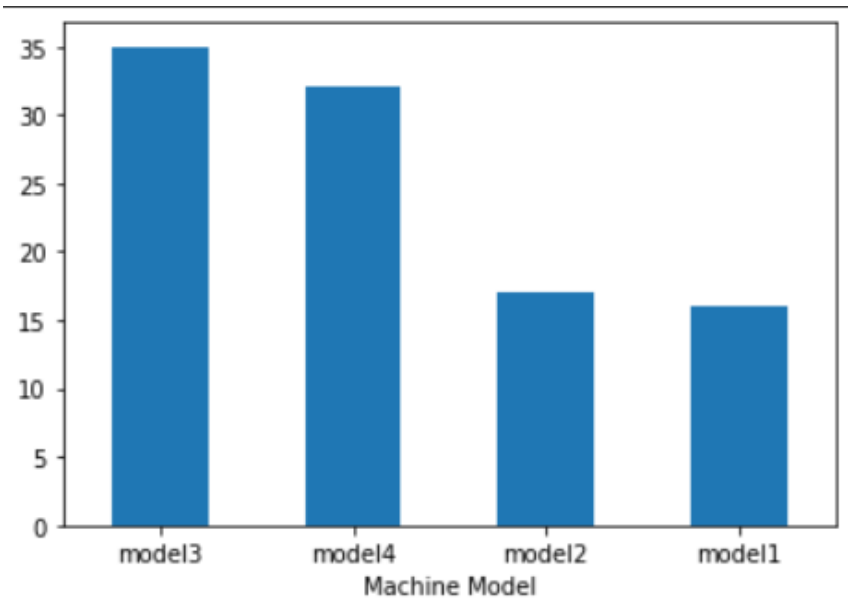

Figure 5.2 Machine model chart

In Figure 5.3, the distribution of component that have maintenance record is shown. Approximately, the maintenance records of all components are close to each other.

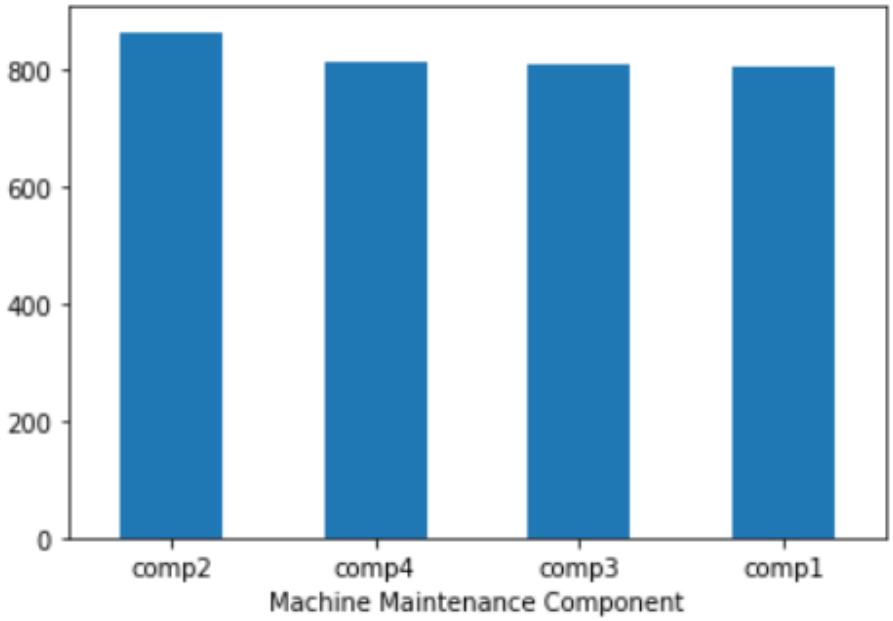

Figure 5.3 Machine maintenance chart

In Figure 5.4, the component distributions that cause the failure are shown. Looking at Figure 5.3, despite the close quantity of maintenance records, component 2 causes more failure.

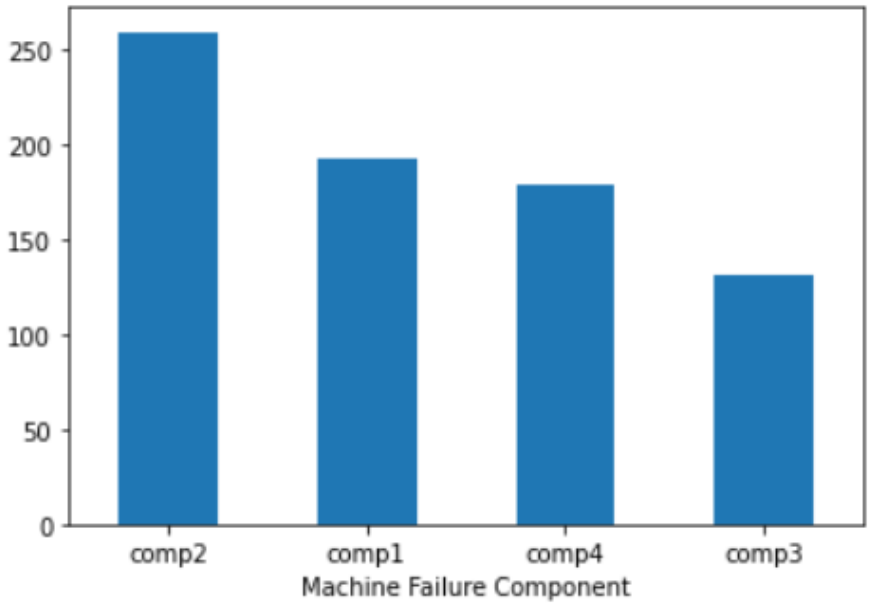

Figure 5.4 Machine failure chart

In Figure 5.5, voltage, rotate, vibration and pressure histograms are displayed in the same chart. Looking at the histograms, it can be said that all continuous attributes are normally distributed, normality test results are shown in the following stages.

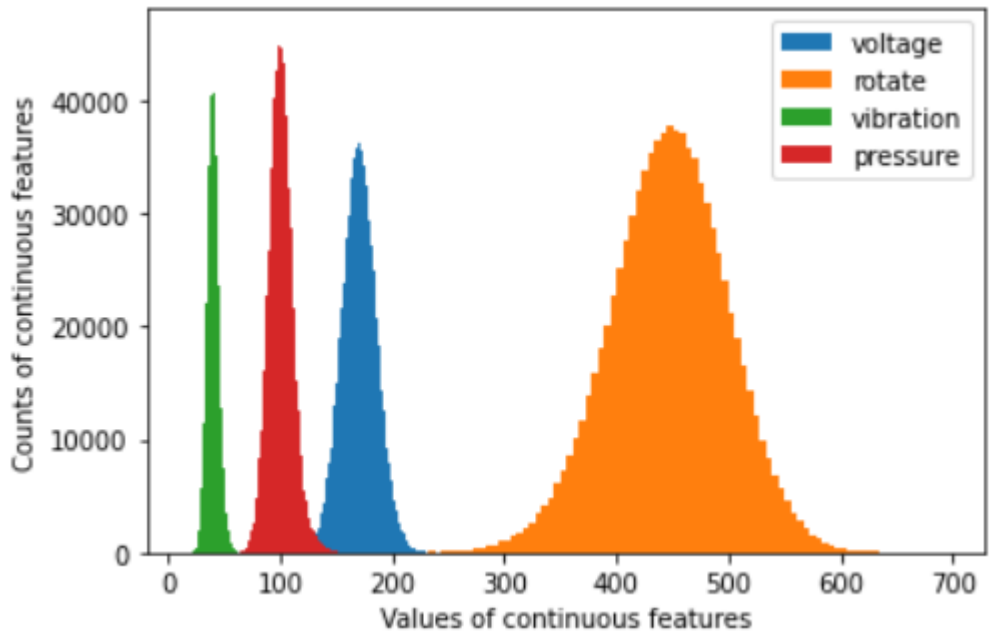

Figure 5.5 Continuous attributes histogram

In general, more information are gained about all attributes in the descriptive analysis phase. The distribution of the ages and models of the machines are examined. Components causing failure are examined. Normality is examined by looking at the histogram of continuous data. In the following stages within the scope of EDA, the subject of normality and correlation is examined.

5.1.3.2 Normality Analysis

The next step is normality test. A normality test is a statistical test that is used to determine whether a sample of data is normally distributed. If the data is normally distributed, it is more likely that the results of a statistical test will be reliable and interpretable. There are several different types of normality tests, including the shapiro-wilk test, the k-squared test and the Anderson-Darling test (Yap, & Sim,

2011). Also, the quantile-quantile plot can help to understand the data normality. For this dataset, quantile-quantile plots and shapiro wilk test are used.

In the Shapiro-wilk test, the p-value is checked. If the p-value is less than 0.05, the data is normally distributed. In Table 5.8, the shapiro-will results are shown.

Table 5.8. Shapiro-wilk results

Attribute	P-Value	Normal Distribution/Not
Voltage	2.962507176128713e-21	Normal distribution
Vibration	0.0	Normal distribution
Rotate	6.445972935894159e-44	Normal distribution
Pressure	0.0	Normal distribution

According to Shapiro-wilk test, all attributes are normal distribution. The another method is quantile-quantile plot (Q-Q plot) method. Q-Q plots are often used in statistical analysis to assess whether a set of data follows a particular distribution, such as the normal distribution. In Figure 6.6, Q-Q plot is shown for vibration,voltage,pressure and rotate.

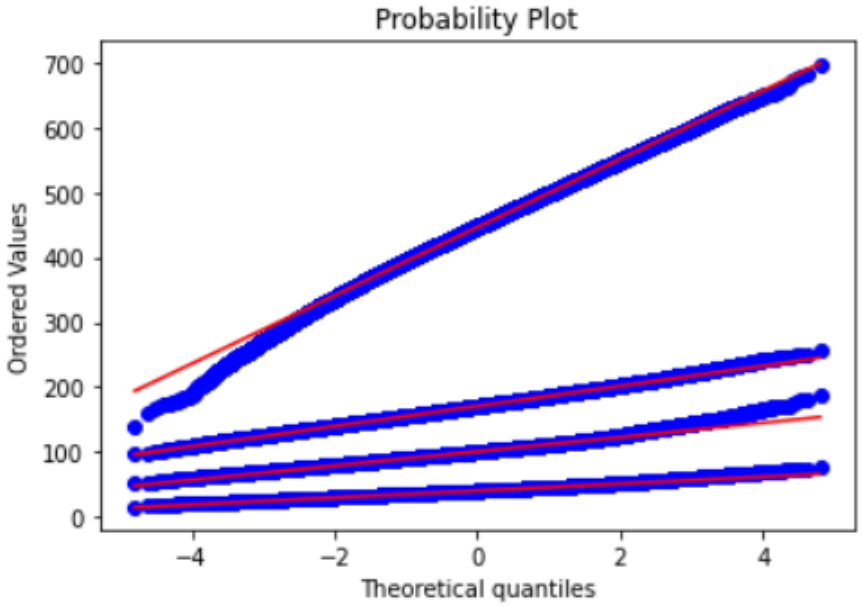

Figure 5.6 Q-Q plot

In this figure, the points in the Q-Q plot falls approximately on a straight line. It means all attributes data is normal distributed.

Normality tests are important because many statistical techniques assume that the data is normally distributed. If the data is not normally distributed, then these techniques may not be appropriate and could lead to incorrect conclusions.

5.1.3.3 Correlation Analysis

Correlation analysis is a statistical technique that is used to measure the strength and direction of the relationship between two variables. The strength of the relationship is measured using a correlation coefficient, which ranges from -1 to 1. A correlation coefficient of 1 indicates a strong positive relationship, a correlation coefficient of -1 indicates a strong negative relationship and a correlation coefficient of 0 indicates no relationship (Gogtay, & Thatte, 2017). Correlation matrix helps to identify relationships between variables and to visualize the strength and direction of these relationships.

There are many tests such as Pearson's correlation, Spearman's rank correlation and Kendall's rank correlation. The p-value is very important in these methods. The p-value is used to determine whether the results of a statistical test are statistically significant, meaning that they are unlikely to have occurred by chance. A small p-value (usually less than 0.05) indicates that the results are statistically significant. On the other hand, a large p-value (usually greater than 0.05) indicates that the results are not statistically significant (Gibbons, & Pratt, 1975).

At this stage, the relationship between numerical data is examined. First of all, the correlation coefficient between numerical data is examined by using the correlation matrix. Then, using the Pearson correlation method, correlation analyzes of numerical data are examined.

In Figure 5.7, correlation matrix is shown. According to the correlation matrix, there is no relationship between the attributes.



Figure 5.7 Correlation matrix

The relationship between attributes is significant or not is decided by applying correlation methods and looking at the p-value. In Table 5.9, Pearson correlation p-values are shown. *** means strong significance, ** means medium significance and * low significance. Looking this table, age has significance relation with pressure and vibration. Vibration has significance relation with voltage.

Table 5.9. Pearson correlation significance (p-values)

	volt	**rotate**	**pressure**	**vibration**	**age**
volt	1.0	0.15548	0.1172	0.02558*	0.53372
rotate	0.15548	1.0	0.52466	0.00373**	0.35435
pressure	0.1172	0.52466	1.0	0.18551	0.00037***
vibration	0.02558*	0.00373**	0.18551	1.0	0.0***
age	0.53372	0.35435	0.00037***	0.0***	1.0

At the basic level, the relation of the attributes with each other is examined. More detailed tests and analyzes are carried out in the following steps.

5.1.4 Data Preparation/Cleaning

Data preparation and cleaning are essential steps before model building. The goal of this section is to get the data into a form that is suitable for analysis. Identifying and handling missing or incorrect values, merging datasets, transforming, labelling, encoding, formatting the data and if it is necessary feature selection operations are explained.

5.1.4.1 Data Merging and Labelling

Data merging is the process of combining multiple datasets into a single dataset. There are five separate datasets here. These datasets need to be combined to make them ready before model building. Each dataset contains a datetime, machine id or both. The datasets are combined using datetime and machine id. Join operations on python dataframe are used as merging method.

In Table 5.10, Five dataset join results using python pandas library merge function are shown. M.id represent machine id.

Table 5.10. Single dataset

datetime	m.ID	volt	rotate	pressure	vibration	age
1.01.2015 06:00	1	176	419	113	45	18
1.01.2015 07:00	1	163	403	95	43	18
1.01.2015 08:00	1	171	527	75	34	18
1.01.2015 09:00	1	162	346	109	41	18
1.01.2015 10:00	1	158	435	112	26	18

The single dataset has three more column named failure,component and error id. These columns have empty values. Empty values must be filled with logical values. Labeling has been applied for failure, comp and error id. Failure column is set as class attribute. For this reason, this column is labeled as failure or not-failure (1-0). The

comp column contains records of the components' maintenance. non-maintenance times are labeled "not-maintenance". Similar to the comp column, empty values are labeled "not-error" for the error id column.

datetime	machineID	volt	rotate	pressure	vibration	model	age	failure	comp	errorID
1.01.2015 06:00	1	176	419	113	45	model3	18	not-failure	not-maintenance	not-error
1.01.2015 07:00	1	163	403	95	43	model3	18	not-failure	not-maintenance	not-error
1.01.2015 08:00	1	171	527	75	34	model3	18	not-failure	not-maintenance	not-error
1.01.2015 09:00	1	162	346	109	41	model3	18	not-failure	not-maintenance	not-error
1.01.2015 10:00	1	158	435	112	26	model3	18	not-failure	not-maintenance	not-error
2.01.2015 10:00	1	158	443	105	42	model3	18	not-failure	comp3	not-error
3.01.2015 10:00	1	158	443	105	42	model3	18	not-failure	comp1	not-error

Figure 5.8 Single Dataset with labelling

In Figure 5.8, The combined dataset with labelling is shown. In the feature selection section, this dataset can change, categorical data can be divided into fewer categories. some unnecessary columns can be removed and the dataset preparation process is completed before it comes to the encoding part. The results of these significance analyzes and dataset operations are shown in the feature selection section.

5.1.4.2 Feature Selection

In the data preparation phase of machine learning, feature selection plays a crucial role. It helps in reducing the complexity of the model, making it more interpretable, and also improves its generalization ability.

There are several statistical test methods that can be used for feature selection in machine learning. These methods are used to evaluate the significance of each feature in predicting the target variable and select the most relevant features for use in the model. Chi-squared test and ANOVA f-value are between these statistical methods

.

Chi-squared test measures the independence of two categorical variables and can be used to select features that are significantly correlated with the target variable (Pandis, 2016). ANOVA F-value measures the difference in mean between two or more groups and can be used to select features that have a significant effect on the target variable (Cuevas et al., 2004).

The dataset contains both categorical and numerical data. For this reason, ANOVA and chi-squared methods have been used for the attributes that affect the failure condition. Simple K Best algorithm of Python sklearn library, which can use ANOVA and chi-squared test methods as scoring function is used for this study.

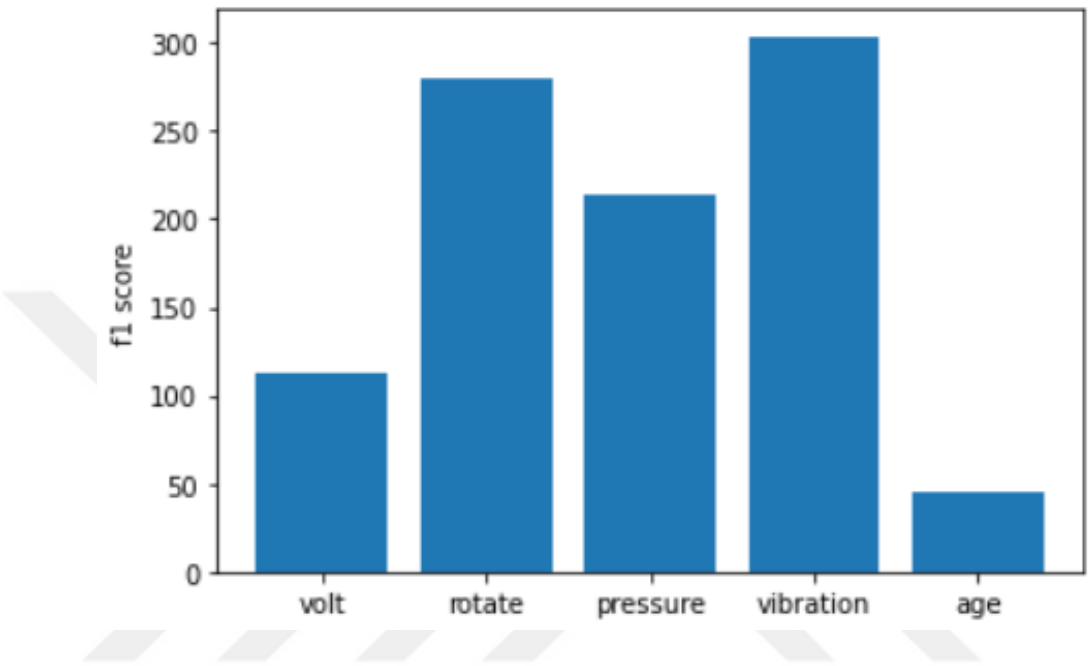

Figure 5.9 Simple k-best ANOVA scoring result

In Figure 5.9, feature selection score values are shown. According to this graph, vibration, rotate, pressure, volt and age attributes are related to failure condition, respectively. When the p-values of these values are also examined, all numerical attributes are statistically significant

After the numerical data are analyzed with the ANOVA test, the chi-squared method is started to be used to analyze the categorical data. Python spicy library is used for this. It is necessary to create a contingency table in the data set prepared for the application of the chi-squared test.

A contingency table is a type of table used to display and analyze the relationship between two categorical variables (Mosteller, 1968). It is used to summarize and compare the distribution of one variable by the levels of another variable. The rows of

the table represent one variable and the columns represent the other variable. The intersection of a row and column represents the count or percentage of observations that fall into that category. In Table 5.11, contingency table example is shown. Rows represent failure mode and columns represent maintenance components.

Table 5.11. Contingency example

target	comp1	comp2	comp3	comp4	not-maintenance
failure	268	342	217	277	18
not-failure	455	448	503	458	874.223

Contingency tables are prepared for categorical data and chi-squared test p-value and chi-statistic values are interpreted.

Table 5.12. Chi-squared results

Attribute	P-Value	Chi Statistics	Target Attribute
Machine ID	1.9733e-14	245.42	Failure
Error ID	0.022633	13.081	Failure
Component	0.0	3.2547e+05	Failure
Model	7.6749e-20	92.12	Failure

In Table 5.12, chi-squared results are shown. According to p-values and chi-statistics, all categorical values are significance for target attribute. Because p-values are smaller than 0.05 and chi statistics values are high.

As a result, when the categorical and numerical data are examined, it was seen that all of them have a significant effect on the failure. In the next step, the encoding processes for the selected attributes/features are explained.

5.1.4.3 Encoding

In the context of model building, encoding step is the process of converting data into a format that can be understood and processed by a computer. This typically involves converting raw data, such as text or numerical values, into a numerical representation that the model can understand and use to make predictions or decisions (Ul Hag et al., 2018).

There are a few different types of encoding that can be used for machine learning models, depending on the type of data being used and the requirements of the model. These are one-hot encoding, label encoding, normalization,

One-hot encoding is commonly used for categorical data and involves creating a separate binary column for each category in the data. Each row of the data will have a 1 in the column corresponding to its category and 0s in all other columns (Ul Hag et al., 2018).

Label encoding involves assigning a unique integer to each category in the data. This can be useful when the categories have some inherent order or ranking, but can cause problems if the model interprets the encoded values as having a numeric relationship (e.g., "1" being "greater than" "2").

Normalization scales numeric data to a common range, such as [0, 1] or [-1, 1]. Normalization can be useful when the scale of the data varies significantly, as it can help the model learn more effectively.

In this study, one-hot encoder method is used for categorical data. The dataset is prepared by applying the normalization process for all attributes for model building. The python sklearn library is used for this.

Table 5.13. Encoding result

Attribute	Category Number	Column examples
Machine ID	100	Machine_2, Machine_3, Machine_4…
Error ID	5	Error_2, Error_3, Error_4, Error_5
Component	4	Comp_2, Comp_3, Comp_4
Model	4	Model_2, Model_3, Model_4

The categorical data are created into columns for each different value with the one-hot encoder method. In Table 5.13, examples of columns created are shown. In order to avoid the dummy variable trap, one column should drop. The dummy variable trap occurs when all of the categories in a categorical feature are perfectly correlated, which can lead to multicollinearity and make it difficult to interpret the results of the model.

Dataset properties are shown in Table 5.14 after coding operations. First of all, the target attribute failure column is coded as non-failure is 0, failure is 1. According to number of failure and not-failure, this column is imbalanced. Category columns are dropped from the dataset since categorical data are created as columns.

Table 5.14. Dataset information

Properties	Explanation/Quantity
Number of instances	877209
Number of attributes	117
Number of continuous attributes	4
Number of discrete attributes	1
Number of binary attributes (without target attribute)	111
Target attribute name	failure
Number of failure (1)	1122
Number of not-failure (0)	876087

As the last step before model building, the normalization process is applied to the prepared dataset. In this way, it is aimed to increase the accurate of the models. Python min-max scaler algorithm is used for normalization.

5.1.5 Model Building and Training

Model building and training is a critical aspect of machine learning, as it determines the accuracy and effectiveness of the model in solving the problem it was designed to address. Careful planning and attention to detail are key to ensuring that the model is able to learn from the data and make accurate predictions or decisions. In the previous sections, the dataset are examined in detail and a usable dataset are created. In this section, algorithm selection for the model, training operations, selection of performance metrics and model results are explained.

5.1.5.1 Model Selection

Model selection is very important to choose the model accurate for the problem in manufacturing. Different models have different strengths and weaknesses so the models to be used for training should be determined initially by looking at the number of attributes in the dataset, the number of instances and the structure of the target attribute.

Failure column that is the target attribute consists of 0 and 1 values so the logistic regression model is chosen first. Support vector mahine is chosen as the second method due to the large number of columns and because it is a successful algorithm considering high dimension dataset. Finally, the random forest decision tree algorithm, which can give successful results in large datasets, has been selected.

5.1.5.2 Training

Model training is the process of teaching a machine learning model to make accurate predictions based on data. Training step goal is to minimize the error between the predicted outputs and the true labels. In this study, three models, one logistic regression, one support vector machine and one random forest decision tree are trained. The python sklearn library is used for training the models.

The dataset created in the data preparation step is split as a train and test set in the training part. Splitting data into a training set and a test set is an important step in the model training process because it allows to evaluate the performance of the model on unseen data.

Many models use k-fold cross validation method when training instead of the train test split. Cross-validation is a technique that helps to estimate the performance of a ML model by using multiple subsets of the data for training and testing. (Refaeilzadeh et al., 2009). It helps mitigate overfitting for accurate estimating of the model's performance on new, unseen data. The k-fold cross validation method are not preferred due to the data set here is imbalanced. Because using k-fold cross-validation for imbalanced data can be problematic because it may result in some folds having a very imbalanced class distribution. This can lead to poor performance on those folds, especially if the model is sensitive to imbalanced data.

When working with imbalanced data, it is important to use performance metrics that are appropriate for imbalanced datasets. This is standard metrics like accuracy can be misleading when the classes are imbalanced, as the model can achieve high accuracy by simply predicting the majority class all the time. For this reason, other performance metrics are used. these are f1-score, precision, recall and AUC value. The criterion of success is to predict the breakdown if evaluated in terms of the problem. not being able to predict a real breakdown is more important than estimating a breakdown at the wrong time. Therefore, recall and AUC values are more considerable than other metrics. Considering these metrics three type models are developed. The

training process is continued with different parameters when necessary by looking at the values of the performance metrics. In Figure 5.10, logistic regression training logic is shown. The model is based on weight update.

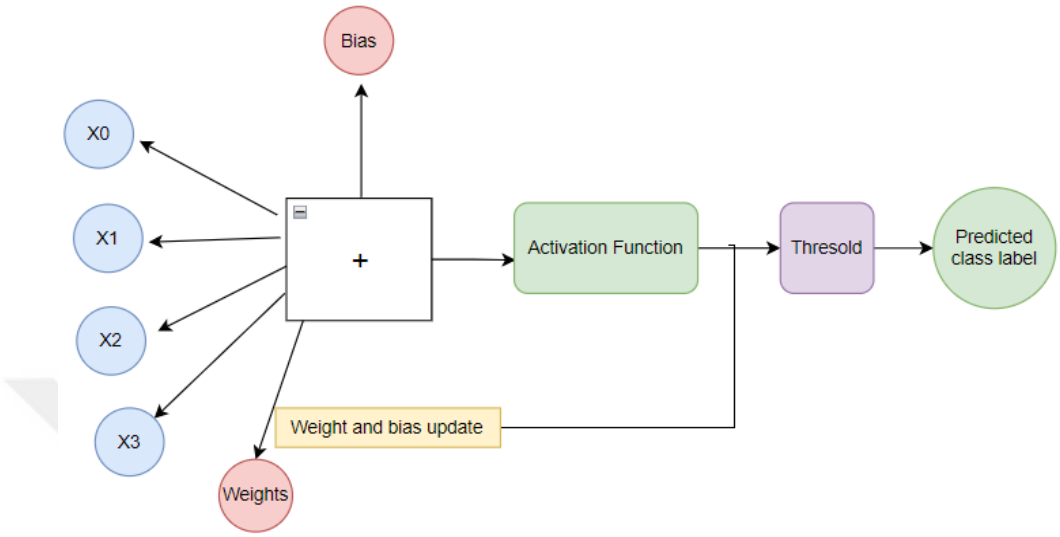

Figure 5.10 Logistic regression training logic

The activation function is the sigmoid function, which maps any input value to a value between 0 and 1. After sigmoid function prediction, the cross-entropy loss function is used as the cost function because it is well-suited for classification tasks and can effectively measure the difference between the predicted probability and the true label. The goal in training a logistic regression model is to minimize the cross-entropy loss. The weights are also updated to minimize cross-entropy.

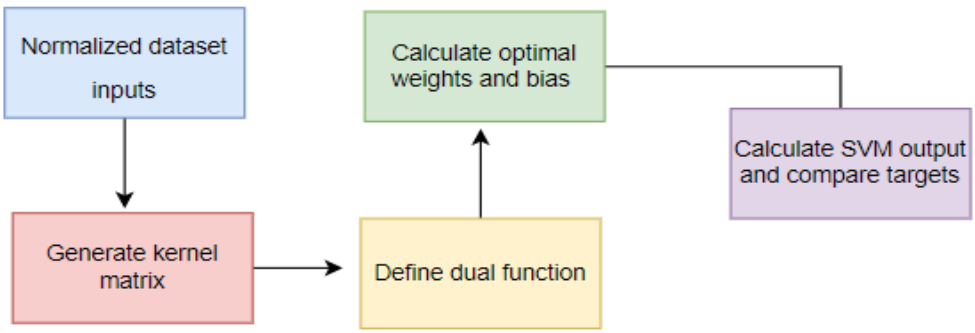

Figure 5.11 Support vector machine training logic

In Figure 5.11, support vector machine training logic is shown. A kernel is a function that calculates the similarity between two input vectors, represented by a scalar value. It maps the input data into a high-dimensional space, allowing for non-linear decision boundaries to be created in the original input space. The kernel matrix is a matrix of all the pairwise kernel values between the input vectors in the training set. The dual function in support vector machines is a mathematical function that represents the optimization problem being solved in the training process.

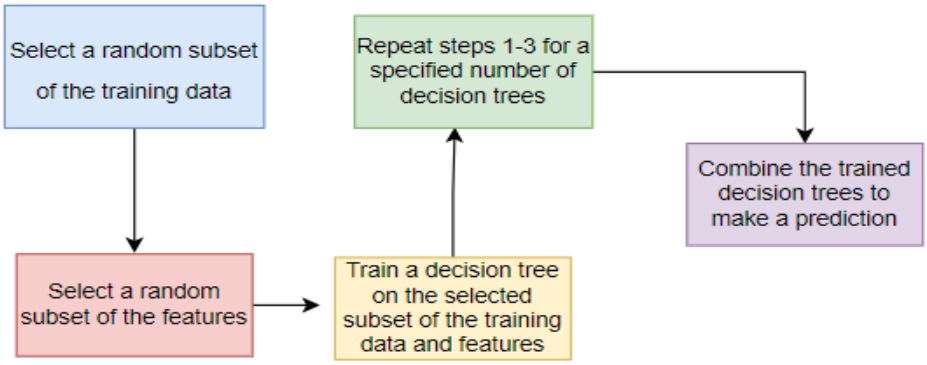

Figure 5.12 Random forest decision tree training logic

In Figure 5.12, random forest training logic is shown. In each iteration, a decision tree is trained on a different subset of the training data and a different subset of the features, which helps to reduce overfitting and improve the generalization of the model.

As a result, logistic regression, SVM and random forest models mentioned with 877229 instances and 117 attributes are trained using the prepared train set. Prediction results are taken for each model and the performance metrics are calculated using test dataset. Performance metrics are shown in detail in the model result section.

5.1.5.3 Model Results

The results of the trained models are interpreted in this section. Precision, recall, f1-score, AUC, accuracy values of each model are shown and confusion matrix is visualized.

The first model is logistic regression model. Python sklearn library is used. Some of logistic regression model parameters are solver and max iteration number. The "solver" parameter in the logistic regression model specifies the algorithm to use for optimization. The solver is responsible for finding the optimal coefficients for the features in the model. There are four type solver. These are Newton-CG, limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS), coordinate descent algorithm, stochastic average gradient descent algorithm. The logistic regression models are build with four type solver. The results are shown in Table 5.15.

Table 5.15. Logistic Regression models result

Solver	Accuracy	Precision	Recall	AUC
Newton CG	0.99	0.60	0.44	0.72
LBFGS	0.99	0.60	0.44	0.72
Coordinate descent	0.99	0.60	0.44	0.72
Average gradient descent	0.99	0.60	0.44	0.72

By looking at the Table 5.15, it can be said all solver type gives same performance results. Due to imbalanced data the accuracy results are quite high but recall and precision value aren't acceptable. In Figure 5.13, also the confusion matrix is shown. The model miss 197 breakdown from 354 real breakdown according to confusion matrix. Also, model predict 104 false breakdown from 262809 not-breakdown.

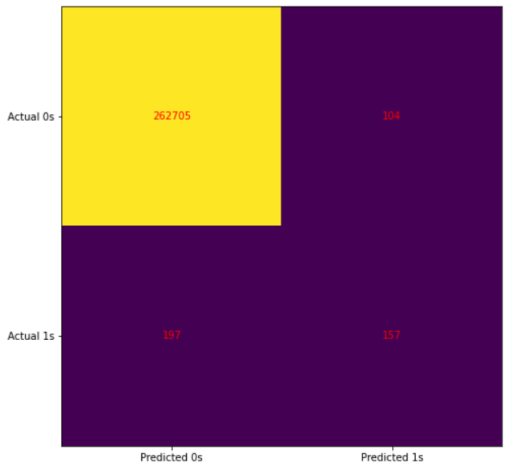

Figure 5.13 Logistic regression confusion matrix

In order to handle the imbalance data, the weighted logistic regression method, which is a variant of logistic regression in which the contribution of each data point to the model is weighted differently has also been tried. The model miss 7 breakdown from 354 real breakdown according to confusion matrix and recall value is 0.98, precision value 0.36, AUC is 0.98 so precision value can be ignored due to the problem nature. In Figure 5.14, weighted logistic regression confusion matrix is shown.

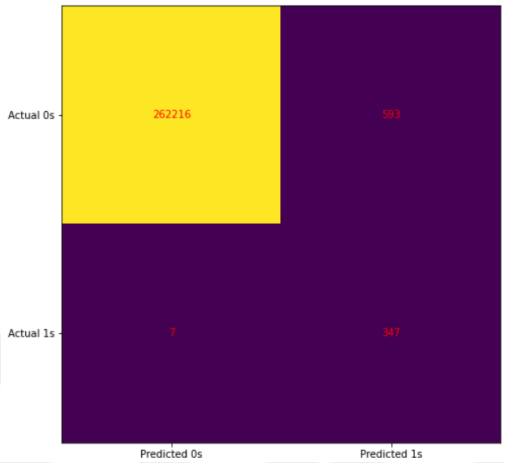

Figure 5.14 Weighted logistic regression confusion matrix

The second model is support vector model. The most important SVM parameter is kernel function. These are linear, polynominal, radial basis function and sigmoid function. The SVM models are build with four type solver. The results are shown in Table 5.16. According to the table most succesfull method is SVM with polynominal kernel function. But SVM is not successful enough in finding breakdowns.

Table 5.16. SVM Models Result

Kernel	**Accuracy**	**Precision**	**Recall**	**AUC**
Linear	0.99	0.53	0.36	0.68
Polynominal	0.99	0.57	0.38	0.69
Radial basis function	0.99	0.54	0.34	0.67
Sigmoid function	0.99	0.32	0.34	0.66

In Figure 5.15, SVM with polynominal kernel function confusion matrix is shown. According the confusion matrix the model doesn't find 216 real breakdown.

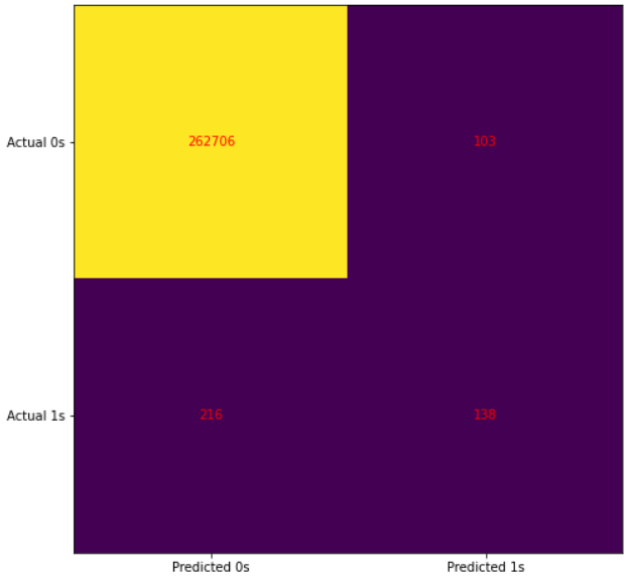

Figure 5.15 SVM with polynominal kernel function

The third model is random forest decision tree. In the sklearn library, the random forest classifier and random forest regressor classes are used to train. In Figure 5.16, random forest decision tree confusion matrix is shown. Precision and recall values of this model are average.

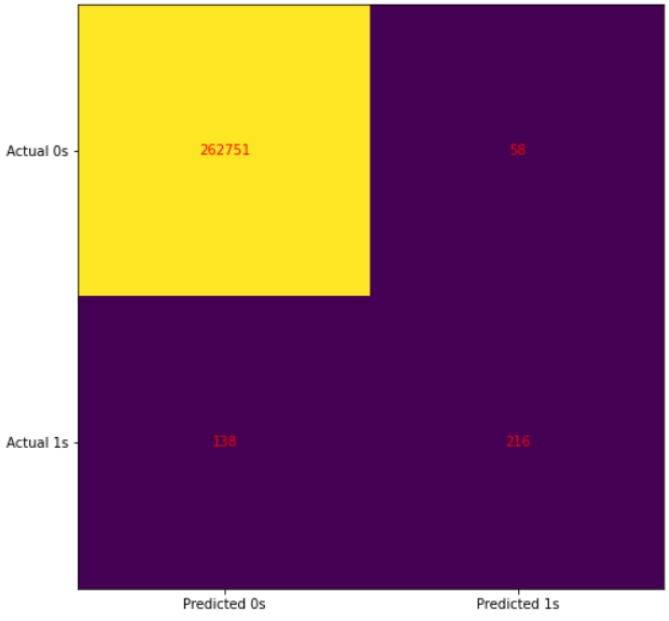

Figure 5.16 Random forest decision tree confusion matrix

As a result, using SVM, logistic regression and random forest decision tree algorithms, the most successful models were selected from the models created with different parameters. The successfull models are shown in Table 5.17.

Table 5.17. Final models result

Model	Accuracy	Precision	Recall	AUC
Weighted logistic regression	0.99	0.36	0.98	0.98
Support vector machine	0.99	0.57	0.38	0.69
Random forest decision tree	0.99	0.78	0.61	0.80

The weighted logistic regression model is very successful in predicting the breakdown considering the recall value and AUC value. Considering the precision values, the random forest algorithm is quite successful, but the recall and AUC values of this algorithm are lower than logistic regression. When all criteria are evaluated, the weighted logistic regression model is chosen as the model to be deployed.

5.1.6 Deployment

The deploy steps for this model are followed since the weighted logistic regression method is chosen as the model to be deployed. The model output is saved in a file with the extension .sav. After that, for the sensor data coming from the database, it is predicted using the sav file. The file with the .sav extension makes predictions when the new data saved and saves it to the database. For example: machine one is failure or not-failure. The machine condition is designed to be viewed on the website. In Figure 5.17, model deployment flow is shown.

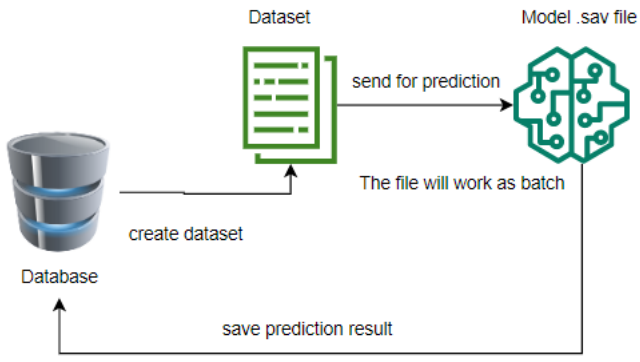

Figure 5.17 Model deployment

5.2 Software Design and Development

In this part, the design and development of the software developed as a web site is explained. The architecture of the software, database tables are explained and the function of the software is explained with UML diagrams.

5.2.1 Architectural View

The software is built according to the n-tier architecture. In the data layer, all database operations are performed using the entity framework (ORM). API is created to display the data on the website. The API is the structure that sends the data to the website and the data format is javascript object notation (JSON).

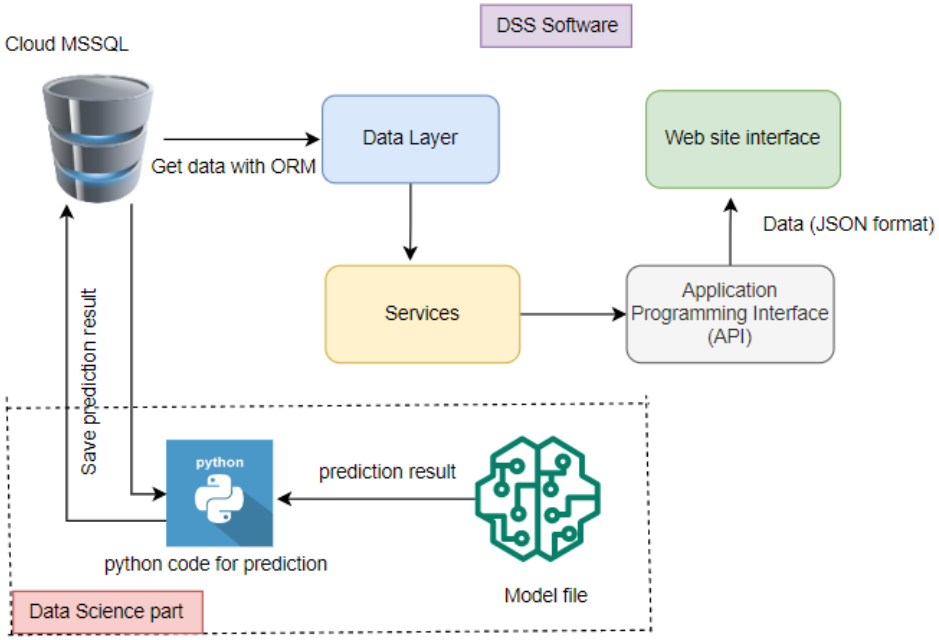

Figure 5.18 Architectural view

In Figure 5.18, the web site architecture is shown. The architecture of the data science part is also included. Here, the prediction results are saved to the database. Prediction data and sensor data are sent to the interface of the website in JSON format using the API. These data coming to the interface are presented to the user with visualizations such as charts and tables.

5.2.2 Class Diagram

A class diagram shows the structure of a system by displaying the system's classes, their attributes and the relationships between the classes. Class diagrams are a way to represent the object-oriented design of a system.

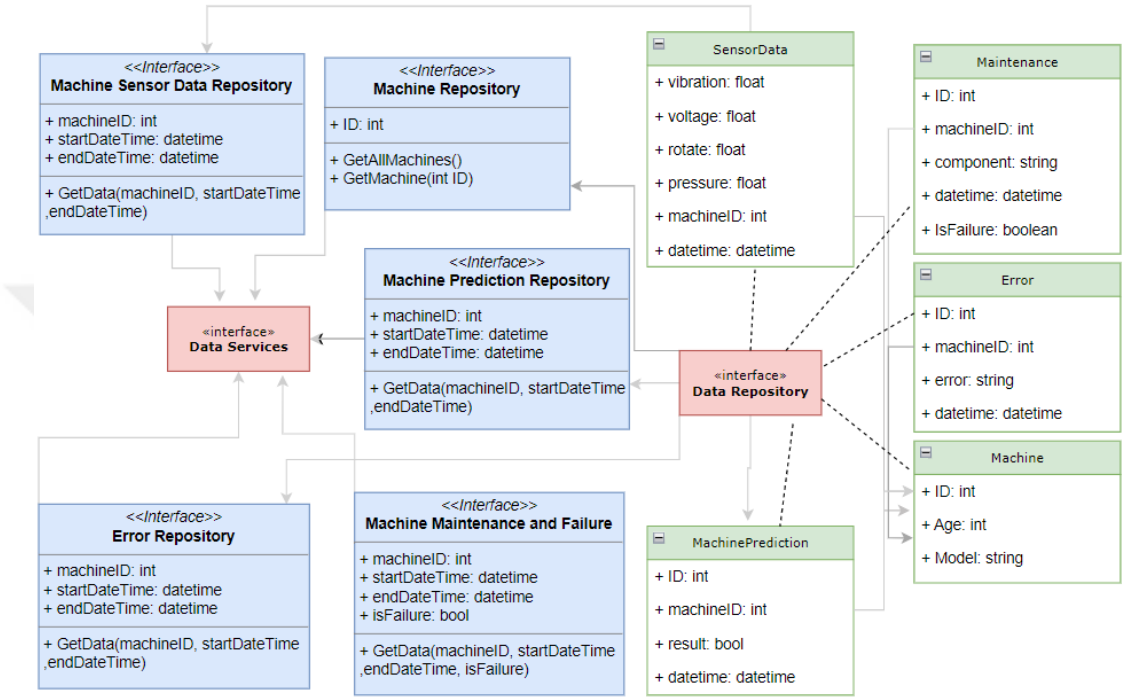

Figure 5.19 Class diagram

In Figure 5.19, class diagram is shown. There are 4 classes. These are machine table that contains machine features, machine sensor data table that contains each machine sensor data, machine error table that contains errors that don't cause failure and maintenance table that contains maintenance records with failure records. The interface section shows which methods are used. Each table has a separate interface. The data repository interface contains functions that will be common to all interfaces. Entity framework functions are used in this interface. For example: get all data, get first or default data.

The data service interface is the part where the data is called from the API. The API only communicates with this interface. In this way, the system is designed with object-oriented rules.

5.2.3 Use Case Diagram

A use case diagram shows the relationships between actors and use cases within a system. Use case diagrams are used to understand the requirements of a system and to identify the functionalities that the system must provide. They are an effective way to visualize the interactions between actors and the system and to understand how the system will be used.

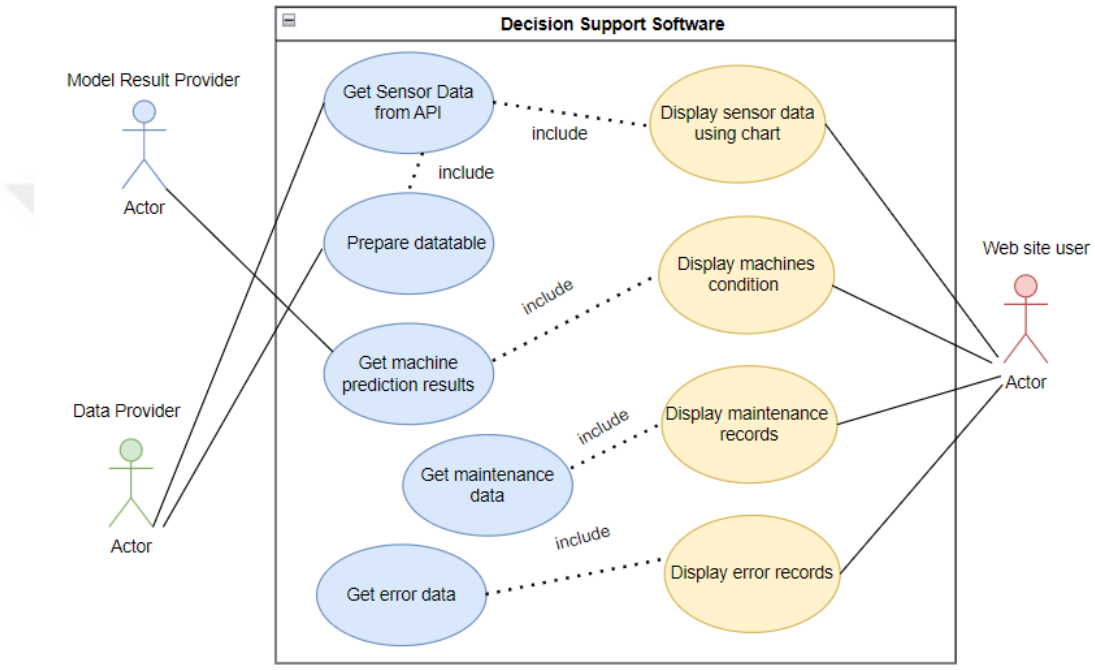

Figure 5.20 Use case diagram

In Figure 5.20, use case diagram is shown. Here the user can see all the data of the machine and list the prediction results. Data providers are responsible for getting the data to the API.

5.2.4 Activity Diagram

An activity diagram is to represent the flow of activities within a system. It is used to model the behavior of a system by showing the interactions between the various elements within the system.

In Figure 5.21, activity diagram is shown. The activities are drawn considering the interface, API and database parts.

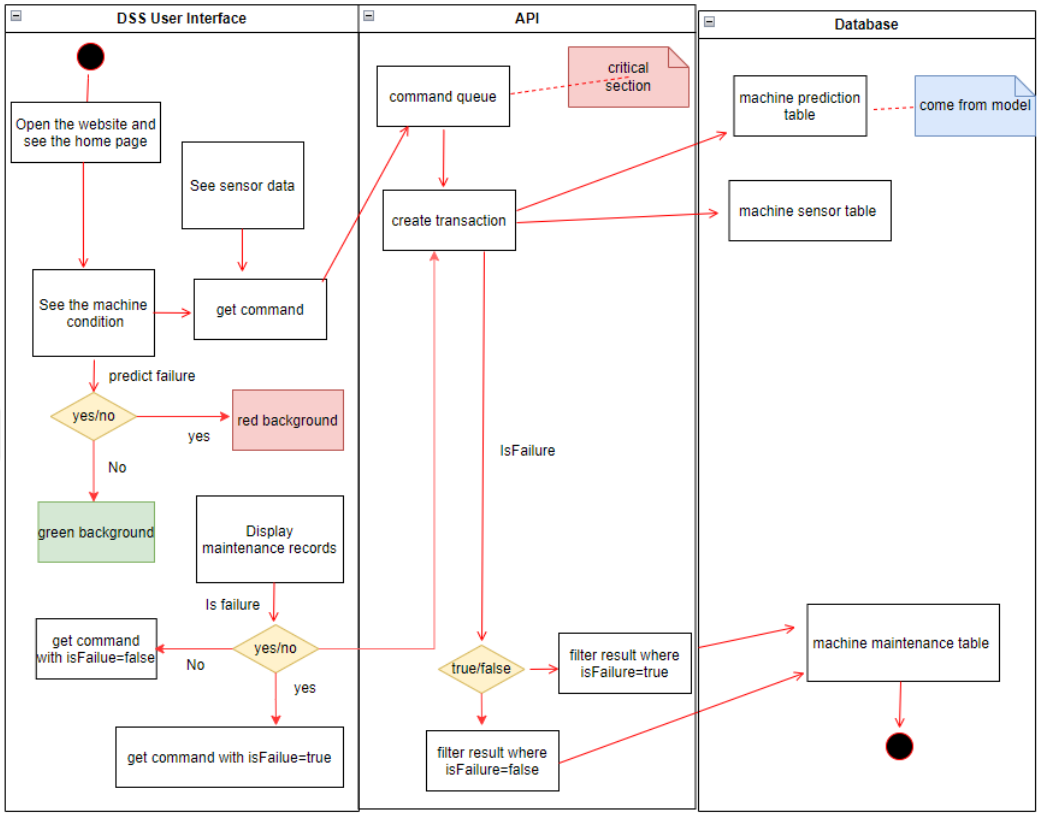

Figure 5.21 Activity diagram

5.2.5 Sequence Diagram

A sequence diagram is a type of diagram that represents the interactions between objects or components in a system over a period of time. It shows the objects and the messages that are exchanged between them, along with the sequence in which these messages are sent and received.

In the sequence diagram, functions, interfaces, models used by the functions are given. Basically, this diagram explains which functions and how they are used.

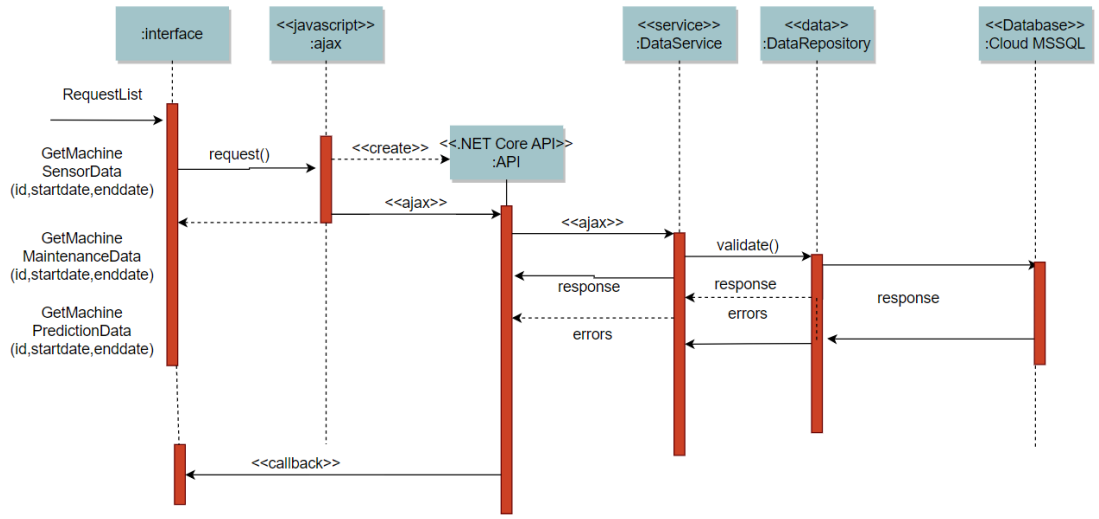

Figure 5.22 Sequence diagram

In Figure 5.22, sequence diagram is shown.

Creating UML diagrams is a very important step when implementing software. Projects are created and business codes are written according to these diagrams. The creation of the website is explained in more detail in the web site development part.

5.2.6 Web Site Development

Decision support system web site have implemented that displays sensor data and prediction results for machines, using a combination of HTML, bootstrap, .NET core API and cloud MSSQL. The web site is organized using an n-tier architecture, with separate layers for the data, services, API and web. Some steps are considered while developing the website. These steps are explained below.

- Setting up a cloud MSSQL instance on a cloud platform on Google cloud for providing the database for web site.
- Creating a new .NET core API project using visual studio or the dotnet command-line interface for providing the backend logic for web site.

- Installing the entity framework nuget package and configure it to connect to the cloud MSSQL database to allow the .NET core API to interact with the database and perform tasks such as querying and updating data.
- Setting up the HTML and javascript files for the web site. The website use MVC pattern. A few view files are created ve designed the interface.
- Implementing interfaces in the .NET Core API to define the contracts for the services and data access objects that will be used in the web site to provide consistent and standardized way of interacting with the data and services in the web site. In this web site, data service is created.
- Connectting the HTML and javascript files to the .NET core API using tools like the fetch API for making HTTP requests to the API to retrieve data from the database or update data in the database.
- Testing the web site locally to make sure everything is working correctly, including the connection to the Cloud MSSQL database and the use of Entity Framework and interfaces in the .NET Core API.
- Deploying the web site to a Google hosting platform

Database, .NET Core, HTML and javascript codes are created in the system. Store procedures are written in the database part, business codes are written in the .NET core part and chart visualizations are used in the front end for web site.

```
CREATE PROCEDURE [dbo].[GetMachineSensorData]
    @MachineID int,
    @StartDate datetime,
    @EndDate datetime
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;
    SELECT * from MachineSensorData
    where MachineID=@MachineId AND DateTime BETWEEN
    CONVERT(VARCHAR(255), @StartDate, 121)
    and CONVERT(VARCHAR(255), @EndDate, 121)
END

GO
```

Figure 5.23 SQL creating store procedure query

In Figure 5.23, a store procedure is created using the machine id, start date and end date parameters. This procedure lists the sensor data for the specified machine in the relevant date range.

The figure 5.24 illustrates the utilization of .NET Core codes in conjunction with a store procedure created within the database. This store procedure is specifically called utilizing .NET Core and the Entity Framework in order to facilitate the transmission of data from the back-end to the front-end of the application. The purpose of this operation is to ensure that the data is properly received by the back-end and then subsequently sent to the front-end to be displayed on the user interface for the end user to interact with and utilize. The use of .NET Core and the Entity Framework in conjunction with a store procedure allows for an efficient and streamlined process for transmitting data within the application.

```
public List<object> GetMachineSensorData(int machineID, DateTime startDate, DateTime endDate)
{
    CommonRepository.CommonRepository rep = new CommonRepository.CommonRepository(_lineDB);
    using (SqlConnection sql = new SqlConnection(_lineDB.Database.GetConnectionString()))
    {

        DataTable dt = new DataTable();
        SqlCommand cmd = new SqlCommand($@"exec GetMachineSensorData @MachineID={machineID},
        @StartDate='{startDate.ToString("yyyy-MM-dd HH:mm:ss")}',
        @EndDate='{endDate.ToString("yyyy-MM-dd HH:mm:ss")}',
        @IsDelete=0,@IsActive=1", sql);
        cmd.CommandType = CommandType.Text;

        SqlDataAdapter da = new SqlDataAdapter(cmd);
        da.Fill(dt);
        object data = new object();
        return ConvertDatatableOperations.ToAnonymousCollection(dt, data);

    }

}
```

Figure 5.24 .NET Core get data from SQL store procedure function

In Figure 5.25, data visualization code is shown. Highcharts is used for chart creation. Using javascript, machine sensor data such as vibration and voltage are visualized.

```
<script>
Highcharts.chart(containerName, {
    chart: {
        type: 'spline'
    },
    title: {
        text: machineName
    },
    yAxis: {
        title: {
            text: 'Voltage'
        },
        min: 0
    },
    tooltip: {
        headerFormat: '<b>{series.name}</b><br>',
        pointFormat: '{point.x:%e. %b}: {point.y:.2f} m'
    },
    plotOptions: {
        series: {
            marker: {
                enabled: true,
                radius: 2.5
            }
        }
    },
    series: machineSensorData
});

</script>
```

Figure 5.25 Highchart sensor data visualization code

5.2.7 Web Site User Interface (UI)

The user interface that has been designed for users to interact with is focused on providing information about the current state of various machines being used in an industrial setting. Specifically, if a breakdown is expected to occur with a particular machine, the machine will be shown in red on the interface, indicating that immediate attention is required. On the other hand, if a machine is functioning normally, it will be displayed in green.

In addition to the current state of each machine, the user interface also displays the last recorded vibration value for each machine. This allows users to quickly gain an understanding of the machine's current condition, as vibration levels can be an important indicator of the machine's health. To provide even more detailed information, the user interface also includes the capability for users to access measurement charts by clicking on the individual machines. These charts provide detailed data on various performance metrics and allow users to more closely monitor the health and condition of the machines.

As per the Figure 5.26, it represents the home page of the user interface, which provide the snapshot of the current state of all machines, where users can quickly see which machines are functioning normally, and which ones require attention.

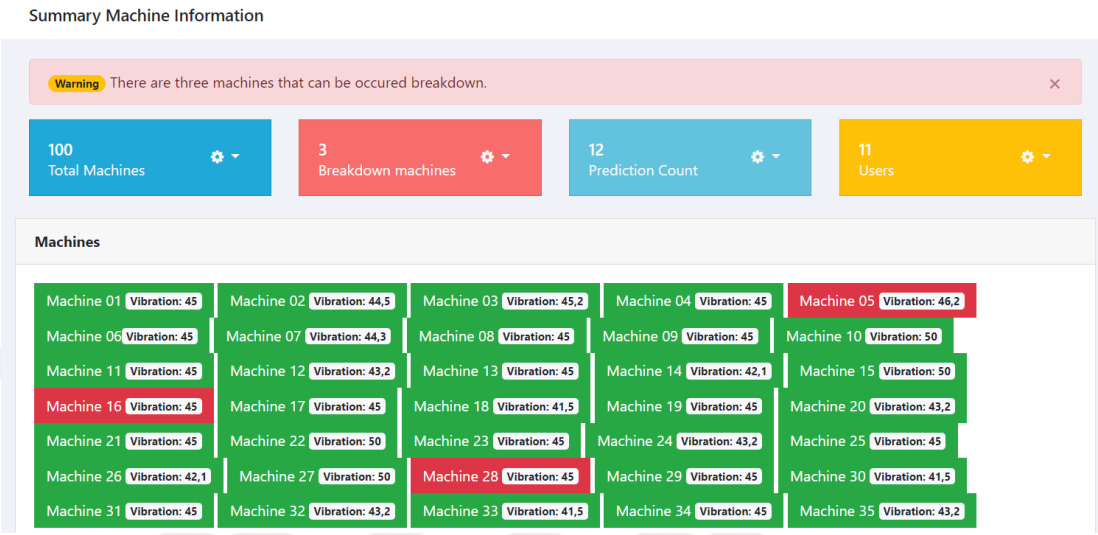

Figure 5.26 Home page

In Figure 5.27, sensor data page is shown. This page can be accessed by clicking on the machines on the homepage or from the menu. The graph of each measurement type is shown separately.



Figure 5.27 Sensor data page

CHAPTER SIX
CONCLUSION

In conclusion, the decision support system using machine learning algorithms has been successfully implemented and deployed.

In this study, a dataset was created from the Azure predictive maintenance dataset, containing sensor data such as vibration, voltage, rotation and pressure for a hundred machines in 2015, as well as machine failures and part replacements.

An exploratory analysis was conducted using Python to understand the relationships between attributes and the dataset was prepared by combining and transforming multiple datasets. The resulting dataset contained 117 attributes and approximately one million data points, with a target variable of 0 (not-failure) or 1 (failure).

Three algorithms were used to build models: random forest decision tree, logistic regression and SVM. The performance of the models was evaluated using the recall and AUC metrics and the weighted logistic regression model was chosen due to its higher recall and AUC of 98%.

The model was integrated into a website to serve as a DSS for predictive maintenance, using technologies such as Google Colab, Python, Cloud SQL, .NET Core API and Entity Framework.

This study has demonstrated the feasibility of developing a predictive maintenance decision support system using sensor data, failure records, component replacements and machine features in a production environment. This system can help optimize maintenance schedules and improve the overall efficiency of the manufacturing process.

REFERENCES

Alasadi, S. A., & Bhaya, W. S. (2017). Review of data preprocessing techniques in data mining. Journal of Engineering and Applied Sciences, 12(16), 4102-4107.

Alzubi, J., Nayyar, A., & Kumar, A. (2018, November). Machine learning from theory to algorithms: an overview. In Journal of physics: conference series (Vol. 1142, No. 1, p. 012012). IOP Publishing.

Bi, Q., Goodman, K. E., Kaminsky, J., & Lessler, J. (2019). What is machine learning? A primer for the epidemiologist. American journal of epidemiology, 188(12), 2222-2239.

Biedermann, H., & Kinz, A. (2019). Lean Smart Maintenance—Value Adding, Flexible and Intelligent Asset Management. BHM Berg-und Hüttenmännische Monatshefte, 164(1), 13-18.

Biswas, S., Wardat, M., & Rajan, H. (2022, May). The art and practice of data science pipelines: A comprehensive study of data science pipelines in theory, in-the-small and in-the-large. In Proceedings of the 44th International Conference on Software Engineering (pp. 2091-2103).

Buathong, P., & Vilasdaechanont, A. (2020, March). Development of Decision Support System for Solving the Machine Selection Problem in an Intermittent Manufacturing System Design. In IOP Conference Series: Materials Science and Engineering (Vol. 784, No. 1, p. 012022). IOP Publishing.

Connelly, L. (2020). Logistic regression. Medsurg Nursing, 29(5), 353-354.

Cuevas, A., Febrero, M., & Fraiman, R. (2004). An anova test for functional data. Computational statistics & data analysis, 47(1), 111-122.

Dalenogare, L. S., Benitez, G. B., Ayala, N. F., & Frank, A. G. (2018). The expected contribution of Industry 4.0 technologies for industrial performance. International Journal of Production Economics, 204, 383-394.

Davis, J., Edgar, T., Porter, J., Bernaden, J., & Sarli, M. (2012). Smart manufacturing, manufacturing intelligence and demand-dynamic performance. Computers & Chemical Engineering, 47, 145-156.

Dhar, V. (2013). Data science and prediction. Communications of the ACM, 56(12), 64-73.

Durgesh, K. S., & Lekha, B. (2010). Data classification using support vector machine. Journal of theoretical and applied information technology, 12(1), 1-7.

El Naqa, I., & Murphy, M. J. (2015). What is machine learning?. In machine learning in radiation oncology (pp. 3-11). Springer, Cham.

Flavio Trojan, Rui F. M. Marçal Journal of Business and Economics, ISSN 2155-7950, USA July 2017, Volume 8, No. 7, pp. 560-572 DOI: 10.15341/jbe(2155-7950)/07.08.2017/005

Gibbons, J. D., & Pratt, J. W. (1975). P-values: interpretation and methodology. The American Statistician, 29(1), 20-25.

Gogtay, N. J., & Thatte, U. M. (2017). Principles of correlation analysis. Journal of the Association of Physicians of India, 65(3), 78-81.

Harimoorthy, K., & Thangavelu, M. (2021). Multi-disease prediction model using improved SVM-radial bias technique in healthcare monitoring system. Journal of Ambient Intelligence and Humanized Computing, 12(3), 3715-3723.

Hilbe, J. M. (2011). Logistic Regression. International encyclopedia of statistical science, 1, 15-32.

Ito, Y. (1991). Representation of functions by superpositions of a step or sigmoid function and their applications to neural network theory. Neural Networks, 4(3), 385-394.

Jagtap, S., Skouteris, G., Choudhari, V., Rahimifard, S., & Duong, L. N. K. (2021). An Internet of Things Approach for Water Efficiency: A Case Study of the Beverage Factory. Sustainability, 13(6), 3343.

Kelleher, John D., Brian Mac Namee and Aoife D'Arcy (2015). Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples and Case Studies. The MIT Press.

Kolberg, D., & Zühlke, D. (2015). Lean automation enabled by industry 4.0 technologies. IFAC-PapersOnLine, 48(3), 1870-1875.

Kusiak, A. (1990). Intelligent Manufacturing.

Kuzlu, M., Fair, C., & Guler, O. (2021). Role of artificial intelligence in the Internet of Things (IoT) cybersecurity. Discover Internet of things, 1(1), 1-14.

Lawless, H. T., & Heymann, H. (2010). Descriptive analysis. In Sensory evaluation of food (pp. 227-257). Springer, New York, NY.

Li, B. H., Hou, B. C., Yu, W. T., Lu, X. B., & Yang, C. W. (2017). Applications of artificial intelligence in intelligent manufacturing: a review. Frontiers of Information Technology & Electronic Engineering, 18(1), 86-96.

Lombardi, M., Pascale, F., & Santaniello, D. (2021). Internet of things: A general overview between architectures, protocols and applications. Information, 12(2), 87.

Mahesh, B. (2020). Machine learning algorithms-a review. International Journal of Science and Research (IJSR).[Internet], 9, 381-386.

Mosteller, F. (1968). Association and estimation in contingency tables. Journal of the American Statistical Association, 63(321), 1-28.

Pandis, N. (2016). The chi-square test. American journal of orthodontics and dentofacial orthopedics, 150(5), 898-899.

Pavlidis, P., Wapinski, I., & Noble, W. S. (2004). Support vector machine classification on the web. Bioinformatics, 20(4), 586-587.

Prajwala, T. R. (2015). A comparative study on decision tree and random forest using R tool. International journal of advanced research in computer and communication engineering, 4(1), 196-199.

Provost, F., & Fawcett, T. (2013). Data science and its relationship to big data and data-driven decision making. Big data, 1(1), 51-59.

Psarommatis, F., & Kiritsis, D. (2021). A hybrid Decision Support System for automating decision making in the event of defects in the era of Zero Defect Manufacturing. Journal of Industrial Information Integration, 100263.

Refaeilzadeh, P., Tang, L., & Liu, H. (2009). Cross-validation. Encyclopedia of database systems, 5, 532-538.

Raschka, S. (2018). Model evaluation, model selection and algorithm selection in machine learning. arXiv preprint arXiv:1811.12808.

Shukla, K., Nefti-Meziani, S., & Davis, S. (2022). A heuristic approach on predictive maintenance techniques: Limitations and scope. Advances in Mechanical Engineering, 14(6), 16878132221101009

Sibagariang, R., & Riandari, F. (2019). Decision Support System for Determining the Best Wood For the Production Cabinet Using Bayes Method: Decision Support System for Determining the Best Wood For the Production Cabinet Using Bayes Method. Jurnal Mantik, 3(3), 99-103.

Sprague, R. H. and E. D. Carlson (1982). Building effective decision support systems. Englewood Cliffs, N.J., Prentice-Hall. ISBN 0-13-086215-13

Torgo, L., & Ribeiro, R. (2009, October). Precision and recall for regression. In International Conference on Discovery Science (pp. 332-346). Springer, Berlin, Heidelberg.

Ul Haq, I., Gondal, I., Vamplew, P., & Brown, S. (2018, November). Categorical features transformation with compact one-hot encoder for fraud detection in distributed environment. In Australasian Conference on Data Mining (pp. 69-80). Springer, Singapore.

Utomo, D. T., & Pratikto, P. B. S. (2020). Preliminary study of web based decision support system to select manufacturing industry suppliers. Journal of Southwest Jiaotong University, 55(2).

Vartak, M., Subramanyam, H., Lee, W. E., Viswanathan, S., Husnoo, S., Madden, S., & Zaharia, M. (2016, June). ModelDB: a system for machine learning model management. In Proceedings of the Workshop on Human-In-the-Loop Data Analytics (pp. 1-3).

Vásquez, F., Cravero, A., Castro, M., & Acevedo, P. (2021). Decision Support System Development of Wildland Fire: A Systematic Mapping. Forests, 12(7), 943.

Vigni, M. L., Durante, C., & Cocchi, M. (2013). Exploratory data analysis. In Data handling in science and technology (Vol. 28, pp. 55-126). Elsevier.

Villamil, S., Hernández, C., & Tarazona, G. (2020). An overview of internet of things. Telkomnika, 18(5), 2320- 2327.

Wakiru, J., Pintelon, L., Muchiri, P. N., Chemweno, P. K., & Mburu, S. (2020). Towards an innovative lubricant condition monitoring strategy for maintenance of ageing multi-unit systems. Reliability Engineering & System Safety, 204, 107200.

Wang, H., Xu, Q., & Zhou, L. (2015). Large unbalanced credit scoring using lasso-logistic regression ensemble. PloS one, 10(2), e0117844.

Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. Neurocomputing, 415, 295-316.

Yap, B. W., & Sim, C. H. (2011). Comparisons of various types of normality tests. Journal of Statistical Computation and Simulation, 81(12), 2141-2155.

Zhong, R. Y., Xu, X., Klotz, E., & Newman, S. T. (2017). Intelligent manufacturing in the context of industry 4.0: a review. Engineering, 3(5), 616-630.