A Course Project report
On

Predictive Maintenance for Industrial IoT

In partial fulfillment of the requirements for the award of

BACHELOR OF TECHNOLOGY

in

Computer Science and Engineering with Data Science(CSD)

Arigala Likhith Kumar	21E51A6703
Raolapalli Pushkara Kaushik Naidu	21E51A6744
Thatipampula Naveen	21E51A6761
V Ramesh	21E51A6762

Under the guidance of

Mrs.CH.SHANTHI PRIYA

Assistant Professor

Department of ET, HITAM

HYDERABAD INSTITUTE OF TECHNOLOGY AND MANAGEMENT

Autonomous, Approved by AICTE, Accredited by NAAC A+, NBA.

Basuragadi (Village), Medchal (Mandal), Medchal - Malkajgiri (Dist.), Hyderabad, TS- 501401.

2023–2024

CERTIFICATE

This is to certify that the project entitled " **Predictive Maintenance for Industrial IoT** " is a bonafide work carried out by **Arigala Likhith Kumar** bearing Roll No.21E51A6703, **Raolapalli Pushkara Kaushik Naidu** bearing Roll No.21E51A6744, **Thatipampula Naveen** bearing Roll No.21E51A6761 and **V Ramesh** bearing Roll No.21E51A6762 in partial fulfillment of the requirements for the degree BACHELOR OF TECHNOLOGY in CSD by the Jawaharlal Nehru Technological University, Hyderabad, during the academic year 2023-2024. The matter contained in this document has not been submitted to any other University or institute for the award of any degree or diploma.

Internal supervisor	**Head of the Department**
Mrs.CH.SHANTHI PRIYA	**Dr. M.V.A.NAIDU**
Assistant Professor	HOD
Department of ET	Department of ET

HYDERABAD INSTITUTE OF TECHNOLOGY AND MANAGEMENT

Autonomous, Approved by AICTE, Accredited by NAAC A+, NBA.

HYDERABAD, TELANGANA STATE

Department of Computer Science and Engineering with Data Science(CSD)

DECLARATION

We "Arigala Likhith Kumar , Raolapalli Pushkara Kaushik Naidu , Thatipampula Naveen , V Ramesh" student of Bachelor of Technology in CSD, session: 2023-2024, Hyderabad Institute of Technology and Management, Dundigal, Hyderabad, Telangana State, hereby declare that the work presented in this Project entitled "**Predictive Maintenance for Industrial IoT**" is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of engineering ethics. It contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Arigala Likhith Kumar (21E51A6703)
Raolapalli Pushkara Kaushik Naidu (21E51A6744)
Thatipampula Naveen (21E51A6761)
V Ramesh (21E51A6762)

ACKNOWLEDGEMENT

An endeavor of a long period can be successful only with the advice of many well- wishers.

We would like to thank our chairman Sri **Mr.ARUTLA PRASHANTH** for providing all the facilities to carry the internship work successfully.

We would like to thank our Principal **Dr.S.ARVIND** who has inspired us a lot through their speeches and provided this opportunity to carry out our project work successfully.

We are very thankful to our Head of the Department, **Dr. M.V.A.Naidu** and B.Tech Project work coordinator **Mrs.CH.SHANTHI PRIYA**

We would like to specially thank my internal supervisor, **Ms. RICHA TIWARI** for technical guidance, constant encouragement, and enormous support provided to us for carrying out our internship work.

Arigala Likhith Kumar (21E51A6703)
Raolapalli Pushkara Kaushik Naidu (21E51A6744)
Thatipampula Naveen (21E51A6761)
V Ramesh (21E51A6762)

DECLARATION

We hereby declare that the course project on **“ Predictive Maintenance for Industrial IoT”** **was** submitted to **Hyderabad Institute of Technology and Management. Hyderabad (AUTONOMOUS)** as part of Project-based learning in the academic initiative, is a result of work done by us in III B.Tech II semester. It is further declared that the project report or any part thereof has not been previously submitted to any other university or institute.

Arigala Likhith Kumar (21E51A6703)
Raolapalli Pushkara Kaushik Naidu (21E51A6744)
Thatipampula Naveen (21E51A6761)
V Ramesh (21E51A6762)

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

Industry 4.0 enables technological trends like Big Data Analytics and Machine Learning techniques to converge into and merge with traditional manufacturing processes, resulting in smart manufacturing. Smart manufacturing techniques leverage the use of Industrial Internet of things (IIoT) technology using IoT sensors that are fitted on physical assets to enhance manufacturing processes. IoT Sensors enable smart manufacturing facilities capable of autonomously exchanging information, which can be used to drive business decisions more accurately. Businesses that adopt Smart manufacturing techniques lead to a competitive advantage for these firms as they can bring in higher profit margins, reduced maintenance costs, energy savings, and better-quality products. This Project proposes an architecture for IIoT based predictive maintenance. This project focuses on creating a predictive maintenance model for industrial machines in the automobile industry. By leveraging IoT technology, the model aims to anticipate sudden breakdowns, making production and maintenance processes smarter. The study showcases how implementing such a model can lead to improved efficiency and cost savings in manufacturing operations.

Keywords:
Industry 4.0, Smart Manufacturing, Industrial Internet of Things (IIoT), Predictive Maintenance, IoT Sensors, Automobile Industry

CHAPTER-1

INTRODUCTION

In today's highly competitive business environment, organizations want to increase production and operational efficiency, employee safety and drive financial results. Many manufacturing organizations are yet to leverage the use of big data and machine learning analytics due to lack of smart analytic tools and capabilities. The transition and advancements in manufacturing technologies have defined the industrial revolution. Industry 1.0 began in the 1760s, and it led to the advent of mechanization, steam power, and weaving loom. In the year 1870, mass production driven by electricity and assembly lines provided momentum for Industry 2.0. The third industrial revolution 3.0, beginning in the 1960s was the era of Automation and digital technologies – electronic chips, mainframes, PCs and the Internet. All this paved the way for the fourth industrial revolution (4.0) in the 21st century by the confluence of emerging technological breakthroughs, from artificial intelligence, 3D printing, nanotechnology, machine learning to quantum computing.

The current study aims to deduce an approach for Predictive Maintenance using IoT sensors that will aid manufacturing industries to predict machine failures before asset breakdown occurs. where the solution is developed using Machine Learning's (ML) binary classification Support Vector Machine, Classification and Regression Trees (C&RT), and Boosted Classification Trees.

Maintenance is a large part of any industry and depending on the industry can represent between 15 and 60 percent of total costs. In the United States, industries spend more than $200 billion each year on maintenance. Recent surveys have shown that there is a need for better maintenance implementation. Improper and unnecessary maintenance actions make up almost 33% of maintenance costs

In order to achieve both high reliability and low cost, a predictive maintenance strategy would need to be implemented. Predictive maintenance is condition-based and maintenance is performed only when the component requires it. Predictive maintenance also allows for predictions to be made on when a component will fail which allows maintainers to be prepared for that failure before it happens. Implementing predictive maintenance can provide a company with an optimal maintenance strategy at a low cost with the highest reliability.

Internet of Things

The Internet of Things (IoT) refers to a network of interconnected objects or things. These objects are able to communicate and interact with each other regardless of physical location. In the maintenance world, these objects are embedded with sensors that monitor the condition of the object. Over the years, the IoT has increased in popularity in maintenance. It provides a framework for users to connect and collect data from all the components and systems that they are monitoring. This data is collected and analyzed in real time to identify and extract meaningful relationships in the data. The collected data can also be compared to previous trends that were calculated with historical data. In the IoT framework, not only are the physical components and systems connected, but also every user in the maintenance process is connected. This means that a user does not need to be physically with the asset to know its condition. The information a user would need is presented to them with real-time dashboards

and alerts.

The IoT has changed the way users think about data. If an IoT framework is being implemented, then a large amount of data is continuously being collected in real-time from the network of sensors. This data being collected from the components and systems is full of valuable insights and useful relationships but it also presents some difficulties. One problem is how to manage and extract value from the IoT data. Previously, data could be stored locally but this approach has its limits on storage and efficiency. In order to quickly store, manage, and analyze IoT data, cloud computing will need to be used.

This work is placed in the field of Data Science with application to the area of predictive maintenance. The need to have a way to determine whether or not a particular machine will fail, as well as the nature of the failure, is essential for generation 4.0 industries. The main reason lies behind the following consideration: the repair or replacement of a faulty machine generally requires costs that are much higher than those required for the replacement of a single component. Therefore, the installation of sensors that monitor the state of the machines, collecting the appropriate information, can lead to great savings for industries.

FIG 1.0 - Industrial IOT

CHAPTER -2

EXISTING METHOD

In industries, machinery requires continual maintenance for proper working. Industries follow various maintenance approaches for their machinery, depending on their specific needs and constraints. Here is a summary of some common maintenance approaches:

Reactive Maintenance:

This approach involves waiting for a piece of equipment to fail and then repairing or replacing it. Reactive maintenance is typically the least expensive option in the short term, but it can lead to increased downtime and higher costs in the long run

Preventive Maintenance:

This approach involves performing maintenance tasks on a fixed schedule, such as replacing parts or conducting inspections, regardless of whether or not the equipment is showing signs of wear or malfunction. Preventive maintenance can help reduce the likelihood of unexpected failures but can also lead to unnecessary maintenance and downtime.

Traditionally, the two most common strategies for maintenance management were either reactive or preventive. Reactive maintenance is a failure-based strategy where maintenance actions are performed only after the component has failed. Preventive maintenance is a time-based or usage-based strategy where maintenance actions are performed after a set amount of time. Both types of strategies have their strengths and weaknesses. Reactive maintenance can save money in the short-term, but long-term can lead to higher repair costs and longer down-times. Preventive maintenance results in higher reliability but can be more costly if maintenance is performed before it is actually needed.

FIG 2.0 - Maintenance Approaches

CHAPTER-3

PROPOSED SYSTEM

Modern manufacturing processes face huge downtime caused due to mechanical failures in Industrial Machines. Conventionally preventive maintenance is being used by various companies to manage and handle these failures. Preventive maintenance is the process of checking, testing, and analyzing the equipment at regular intervals to determine its proper functioning. These frequent checking processes require huge costs leading to greater investment in maintenance. Whereas predictive maintenance is the concept through which an equipment's shutdown period can be determined through its behavior. This approach decreases the number of frequent checks to maintain machines. The research paper has been proposed with an aim to build a system that can reduce the downtime cost in the manufacturing processes in the industries with the use of predictive maintenance. The predictive maintenance will be condition-based depending on various factors such as volume flow, temperatures, vibrations, power consumption.

Predictive Maintenance:

This approach involves using data and analytics to predict when a piece of equipment is likely to fail and then taking action to prevent the failure from occurring. Predictive maintenance can help minimize downtime and reduce maintenance costs, but it requires significant data collection and analysis investment.

Nowadays, many companies still follow a periodic (PM) or condition-based maintenance (CBM) approach. While in PM, industrial machines are maintained at regular intervals, CBM involves defining threshold values for particular sensors, which trigger maintenance operations when exceeded. PM often leads to a waste of personnel and material since, in many cases, maintenance is not necessary and could be postponed.

Among all the maintenance approaches, predictive maintenance using machine learning is a powerful tool that can help organizations improve equipment reliability and reduce costs associated with unplanned downtime. Predictive maintenance is a technique used to predict equipment failure and prevent unplanned downtime by analyzing data from various sources, including sensors, historical maintenance records, and other operational data. Machine learning is an important tool used in predictive maintenance to automatically identify patterns and anomalies in this data, which can help identify potential issues before they become critical. Machine learning models can be trained using historical data to identify patterns that indicate equipment failure is likely to occur. These models can then predict when a piece of equipment is likely to fail and alert maintenance teams to take action before the failure occurs. Using machine learning in predictive maintenance can lead to significant cost savings by reducing downtime, improving equipment reliability, and extending equipment lifespan. It also enables maintenance teams to focus their efforts on the most critical issues rather than spending time on routine maintenance tasks.

CHAPTER-4

BLOCK DIAGRAM

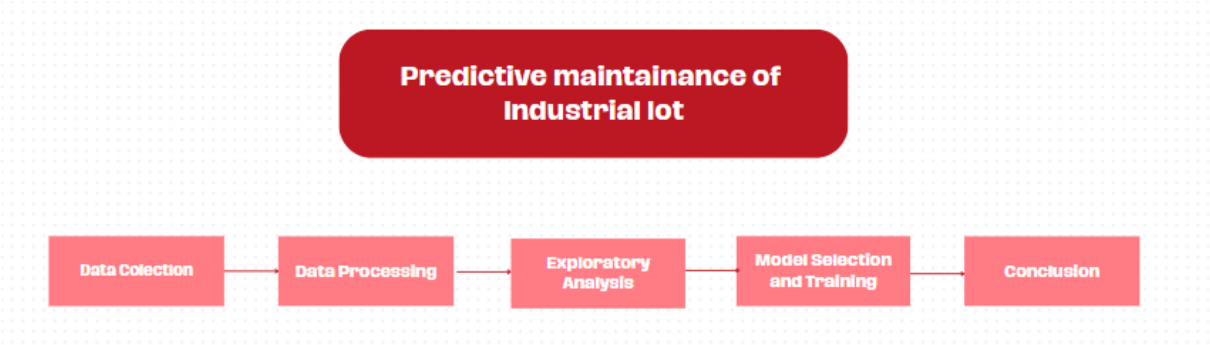


FIG 4.0 - Block Diagram

The image is a flowchart illustrating the process of predictive maintenance for an industrial IoT (Internet of Things) system. The steps involved in the process are as follows:

- **Data Collection**
- **Data Processing**
- **Exploratory Analysis**
- **Model Selection and Training**
- **Conclusion**

CHAPTER – 5
METHODOLOGY

5.1 Data Collection

Here we use the AI4I Predictive Maintenance Dataset from the UCI Repository to carry out an analysis that aims to respond to the needs just reported. In particular, the work is presented through a lineup that characterizes a typical Machine Learning application. In the first place the dataset is explored to obtain a deeper knowledge that can guide in fully understanding the ground truth. Then, some preprocessing techniques are applied to prepare the data for the algorithms we will use to make our predictions. We consider two main tasks: the first consists in establishing whether a generic machine is about to suffer a failure while the second concerns the determination of the nature of the fault. Finally, a comparison is provided between the results obtained by the latter, evaluating both their performance through appropriate metrics, and their interpretability.

Since real predictive maintenance datasets are generally difficult to obtain and in particular difficult to publish, the data provided by the UCI repository is a synthetic dataset that reflects real predictive maintenance encountered in industry to the best of their knowledge. The dataset consists of 10 000 data points stored as rows with 14 features in columns:

- UID: unique identifier ranging from 1 to 10000;
- Product ID: consisting of a letter L, M, or H for low (60% of all products), medium (30%) and high (10%) as product quality variants and a variant-specific serial number;
- Air temperature [K]: generated using a random walk process later normalized to a standard deviation of 2 K around 300 K;
- Process temperature [K]: generated using a random walk process normalized to a standard deviation of 1 K, added to the air temperature plus 10 K;
- Rotational speed [rpm]: calculated from a power of 2860 W, overlaid with a normally distributed noise; Torque [Nm]: torque values are normally distributed around 40 Nm with a standard deviation of 10 Nm and no negative values;
- Tool wear [min]: The quality variants H/M/L add 5/3/2 minutes of tool wear to the used tool in the process; Machine failure: label that indicates whether the machine has failed in this particular data point for any of the following failure modes are true. The machine failure consists of five independent failure modes:
- Tool wear failure (TWF): the tool will be replaced of fail at a randomly selected tool wear time between 200 - 240 mins;
- Heat dissipation failure (HDF): heat dissipation causes a process failure, if the difference between air- and process temperature is below 8.6 K and the tools rotational speed is below 1380 rpm;
- Power failure (PWF):the product of torque and rotational speed (in rad/s) equals the power required for the process. If this power is below 3500 W or above 9000 W, the process fails;
- Overstrain failure (OSF): if the product of tool wear and torque exceeds 11,000 minNm for the L product variant (12,000 M, 13,000 H), the process fails due to overstrain;
- Random failures (RNF): each process has a chance of 0,1 % to fail regardless of its process parameters. If at least one of the above failure modes is true, the process fails and the ’machine failure’ label is set to 1. It is therefore not transparent to the machine learning method, which of the failure modes has caused the process to fail.

5.2 Data Processing

Data Cleaning: Data cleaning is performed as the raw data may contain irrelevant or undesired elements. It manages the feeding of missing data and noisy values through procedures such as outlier detection and others.

Data Transformation: Data Normalization is performed to scale the data values into the desired range. In this phase, attributes are selected as per suitability, and new attributes from existing or known attributes are generated.

Data Processing: In this stage, machine learning algorithms are applied to process the data. Random Forest, SVM, and Decision Tree are used for predicting the data using values such as pressure, volume, temperature and others

Target anomalies: In this section we observe the distribution of the target to find any imbalances and correct them before dividing the dataset. The first anomaly respect to dataset's description is that when the failure is random (RNF), the Machine Failure feature is not set to 1.

Outliers inspection: The goal of this section is to check if the dataset contains any outlier, which are usually misleading for machine learning algorithms. We begin by looking at a statistical report of the numerical features.

Resampling with SMOTE: SMOTE (Synthetic Minority Over-sampling Technique) is a popular method for handling class imbalance in machine learning datasets. It works by generating synthetic samples for the minority class rather than duplicating existing ones, thus helping to balance the class distribution. Here's how you can perform resampling using SMOTE:

Features scaling and Encoding: In order to make data exploitable for the algorithms we will run, we apply two transformations:

- First, we apply a label encoding to the categorical columns, since Type is an ordinal feature and Cause must be represented in one column. The mapping follows this scheme: Type: {L=0, M=1, H=2} Cause: {Working=0, PWF=1, OSF=2, HDF=3, TWF=4}
- Secondly we perform the scaling of the columns with StandardScaler. This is particularly useful for the good working of methods that rely on the metric space, such as PCA and KNN. It has been also verified that using StandardScaler leads to slightly better performances than using MinMaxScaler.

5.3 Exploratory Data Analysis

EDA helps determine how best to manipulate data sources to get the answers you need, making it easier for data scientists to discover patterns, spot anomalies, test a hypothesis, or check assumptions.

EDA is primarily used to see what data can reveal beyond the formal modeling or hypothesis testing task and provides a better understanding of data set variables and the relationships between them. It can also help determine if the statistical techniques you are considering for data analysis are appropriate. Originally developed by American mathematician John Tukey in the 1970s, EDA techniques continue to be a widely used method in the data discovery process today.

The main purpose of EDA is to help look at data before making any assumptions. It can help identify obvious errors, as well as better understand patterns within the data, detect outliers or anomalous events, and find interesting relations among the variables.

5.3.1 Principal component analysis

Principal component analysis, or PCA, is a dimensionality reduction method that is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set.

Reducing the number of variables of a data set naturally comes at the expense of accuracy, but the trick in dimensionality reduction is to trade a little accuracy for simplicity. Because smaller data sets are easier to explore and visualize, and thus make analyzing data points much easier and faster for machine learning algorithms without extraneous variables to process.

So, to sum up, the idea of PCA is simple: reduce the number of variables of a data set, while preserving as much information as possible.

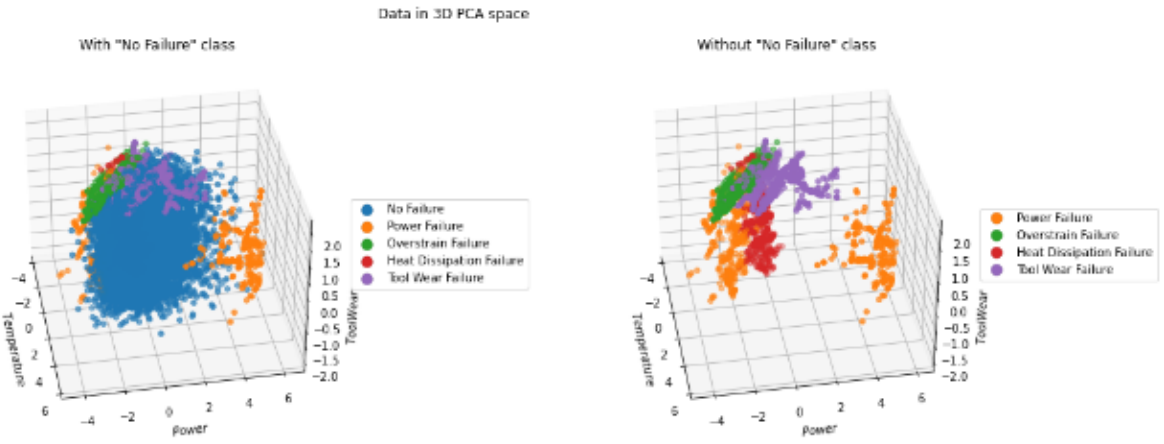


FIG 5.3 - EDA Data Visualization

5.3.2 Correlation Heatmap

Correlation heatmaps are a type of plot that visualize the strength of relationships between numerical variables. Correlation plots are used to understand which variables are related to each other and the strength of this relationship. A correlation plot typically contains a number of numerical variables, with each variable represented by a column. The rows represent the relationship between each pair of variables. The values in the cells indicate the strength of the relationship, with positive values indicating a positive relationship and negative values indicating a negative relationship.

Correlation heatmaps can be used to find potential relationships between variables and to understand the strength of these relationships. In addition, correlation plots can be used to identify outliers and to detect linear and nonlinear relationships. The color-coding of the cells makes it easy to identify relationships between variables at a glance. Correlation heatmaps can be used to find both linear and nonlinear relationships between variables.

5.4 Model Selection and Training

The important machine learning stage of model selection entails selecting the best model and algorithm for a certain task. To make precise predictions on unknown data, it is crucial to find a balance between model complexity & generalization. Model selection involves selecting potential candidates, assessing each model's performance, and selecting the model with the best results.

Assessing the problem's complexity, data quality and availability, interpretability, model assumptions, scalability, efficiency, regularization, domain knowledge, resource restrictions, and the possible advantages of ensemble approaches are all factors that should be taken into account when choosing a model. These factors aid in ensuring that the chosen model complies with the limits and needs of the issue.

There are many methods for choosing a model, such as train-test split, cross-validation, grid searches, random search, Bayesian optimization, model averaging, information criteria, expertise in the domain, and model performance comparison. These methods make it possible to thoroughly assess, tune hyperparameters, and compare various models to get the best fit

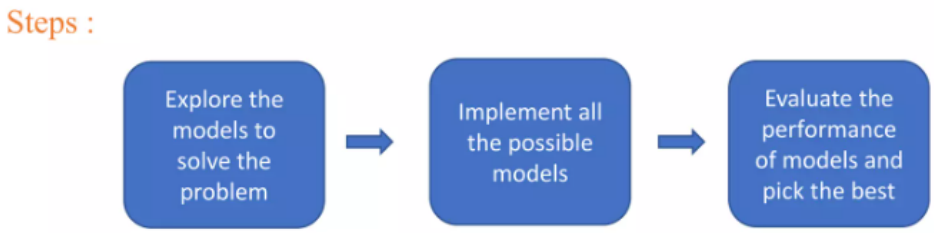


FIG 5.4 - Model Selection and Training

CHAPTER – 6

MACHINE LEARNING ALGORITHMS

6.1 Machine Learning for Predictive Maintenance

Predictive maintenance with machine learning emerges as a viable solution to mitigate the impact of unexpected machine failures . AI-driven anomaly detection and fault prediction in predictive maintenance can yield substantial benefits, including:

- Increasing runtime between 10 to 20%.
- Reducing maintenance costs by up to 10%.
- Minimizing the time needed for maintenance scheduling by up to 50%.

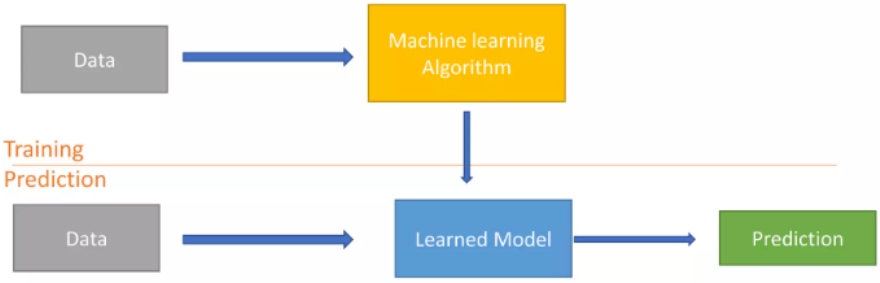


FIG 6.1.1 - Machine Learning Process

Classification algorithms are part of data mining and use supervised machine learning methods to make predictions about data. The aim of this project is to find the best model for binary classification of the dataset to predict whether or not there will be Machine Failure.

The classification techniques we choose to implement are the following:

- Logistic Regression
- K-nearest neighbors (K-NN)
- Support Vector Machine
- Random Forest
- XGBoost

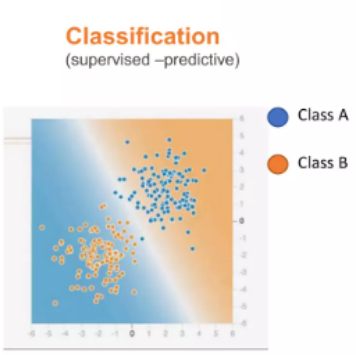


FIG 6.1.2 Classification Algorithm

6.2 Logistic regression

Logistic regression is used for binary classification where we use sigmoid function, that takes input as independent variables and produces a probability value between 0 and 1.

Key Points:

- Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value.
- It can be either Yes or No, 0 or 1, True or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.
- In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).

Logistic Regression Equation:

The Logistic regression equation can be obtained from the Linear Regression equation. The mathematical steps to get Logistic Regression equations are given below:

We know the equation of the straight line can be written as:

$$y= b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$

In Logistic Regression y can be between 0 and 1 only, so for this let's divide the above equation by (1-y):

$$\frac{y}{1-y}$$; 0 for y= 0, and infinity for y=1

But we need range between -[infinity] to +[infinity], then take logarithm of the equation it will

become:

$$log\left[\frac{y}{1-y}\right] = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$

The above equation is the final equation for Logistic Regression.

6.3 K-Nearest Neighbor(KNN) Algorithm

The **K-Nearest Neighbors (KNN) algorithm** is a supervised machine learning method employed to tackle classification and regression problems. Evelyn Fix and Joseph Hodges developed this algorithm in 1951, which was subsequently expanded by Thomas Cover. The article explores the fundamentals, workings, and implementation of the KNN algorithm.

Distance Metrics Used in KNN Algorithm

As we know that the KNN algorithm helps us identify the nearest points or the groups for a query point. But to determine the closest groups or the nearest points for a query point we need some metric. For this purpose, we use below distance metrics:

Euclidean Distance

This is nothing but the cartesian distance between the two points which are in the plane/hyperplane. Euclidean distance can also be visualized as the length of the straight line that joins the two points which are into consideration. This metric helps us calculate the net displacement done between the two states of an object.

$$\text{distance}(x, X_i) = \sqrt{\sum_{j=1}^{d}(x_j - X_{i_j})^2]}$$

6.4 Support Vector Machine

SVMs are commonly used within classification problems. They distinguish between two classes by finding the optimal hyperplane that maximizes the margin between the closest data points of opposite classes. The number of features in the input data determine if the hyperplane is a line in a 2-D space or a plane in a n-dimensional space. Since multiple hyperplanes can be found to differentiate classes, maximizing the margin between points enables the algorithm to find the best decision boundary between classes. This, in turn, enables it to generalize well to new data and make accurate classification predictions. The lines that are adjacent to the optimal hyperplane are known as support vectors as these vectors run through the data points that determine the maximal margin.

Mathematical intuition of Support Vector Machine

Consider a binary classification problem with two classes, labeled as +1 and -1. We have a training dataset consisting of input feature vectors X and their corresponding class labels Y.

The equation for the linear hyperplane can be written as:

$$w^T x + b = 0$$

The vector W represents the normal vector to the hyperplane. i.e the direction perpendicular to the hyperplane. The parameter **b** in the equation represents the offset or distance of the hyperplane from the origin along the normal vector **w**.

The distance between a data point x_i and the decision boundary can be calculated as:

$$d_i = \frac{w^T x_i + b}{||w||}$$

where ||w|| represents the Euclidean norm of the weight vector w. Euclidean norm of the normal vector W

For Linear SVM classifier :

$$\hat{y} = \begin{cases} 1 & : \ w^T x + b \geq 0 \\ 0 & : \ w^T x + b < 0 \end{cases}$$

Optimization:

For Hard margin linear SVM classifier:

$$\underset{w,b}{\text{minimize}} \frac{1}{2} w^T w = \underset{W,b}{\text{minimize}} \frac{1}{2} \|w\|^2$$
$$\text{subject to } y_i(w^T x_i + b) \geq 1 \ for \ i = 1, 2, 3, \cdots, m$$

The target variable or label for the ith training instance is denoted by the symbol ti in this statement. And ti=-1 for negative occurrences (when yi= 0) and to=1 positive instances (when yi = 1) respectively. Because we require the decision boundary that satisfy the constraint:

$$t_i(w^T x_i + b) \geq 1$$

For Soft margin linear SVM classifier:

$$\underset{w,b}{\text{minimize}} \ \frac{1}{2} w^T w + C \sum_{i=1}^{m} \zeta_i$$
$$\text{subject to } y_i(w^T x_i + b) \geq 1 - \zeta_i \ \ and \ \zeta_i \geq 0 \ \ for \ i = 1, 2, 3, \cdots, m$$

Dual Problem: A dual Problem of the optimization problem that requires locating the Lagrange multipliers related to the support vectors can be used to solve SVM. The optimal Lagrange multipliers α(i) that maximize the following dual objective function

$$\underset{\alpha}{\text{maximize}} : \frac{1}{2} \sum_{i \to m} \sum_{j \to m} \alpha_i \alpha_j t_i t_j K(x_i, x_j) - \sum_{i \to m} \alpha_i$$

where,

- αi is the Lagrange multiplier associated with the ith training sample.
- K(xi, xj) is the kernel function that computes the similarity between two samples xi and xj. It allows SVM to handle nonlinear classification problems by implicitly mapping the samples into a higher-dimensional feature space.
- The term ∑αi represents the sum of all Lagrange multipliers.

The SVM decision boundary can be described in terms of these optimal Lagrange multipliers and the support vectors once the dual issue has been solved and the optimal Lagrange multipliers have been discovered. The training samples that have i > 0 are the support vectors, while the decision boundary is supplied by:

$$w = \sum_{i \to m} \alpha_i t_i K(x_i, x) + b$$
$$t_i(w^T x_i - b) = 1 \iff b = w^T x_i - t_i$$

6.5 Random Forest

Machine learning, a fascinating blend of computer science and statistics, has witnessed incredible progress, with one standout algorithm being the Random Forest. Random forests or Random Decision Trees is a collaborative team of decision trees that work together to provide a single output. Originating in 2001 through Leo Breiman, Random Forest has become a cornerstone for machine learning enthusiasts. In this article, we will explore the fundamentals and implementation of Random Forest Algorithm.

The random Forest algorithm works in several steps which are discussed below

- **Ensemble of Decision Trees:** Random Forest leverages the power of ensemble learning by constructing an army of Decision Trees. These trees are like individual experts, each specializing in a particular aspect of the data. Importantly, they operate independently, minimizing the risk of the model being overly influenced by the nuances of a single tree.
- **Random Feature Selection:** To ensure that each decision tree in the ensemble brings a unique perspective, Random Forest employs random feature selection. During the training of each tree, a random subset of features is chosen. This randomness ensures that each tree focuses on different aspects of the data, fostering a diverse set of predictors within the ensemble.
- **Bootstrap Aggregating or Bagging:** The technique of bagging is a cornerstone of Random Forest's training strategy which involves creating multiple bootstrap samples from the original dataset, allowing instances to be sampled with replacement. This results in different subsets of data for each decision tree, introducing variability in the training process and making the model more robust.
- **Decision Making and Voting:** When it comes to making predictions, each decision tree in the Random Forest casts its vote. For classification tasks, the final prediction is determined by the mode (most frequent prediction) across all the trees. In regression tasks, the average of the

individual tree predictions is taken. This internal voting mechanism ensures a balanced and collective decision-making process.

6.6 XGBoost

XGBoost is an optimized distributed gradient boosting library designed for efficient and scalable training of machine learning models. It is an ensemble learning method that combines the predictions of multiple weak models to produce a stronger prediction. XGBoost stands for "Extreme Gradient Boosting" and it has become one of the most popular and widely used machine learning algorithms due to its ability to handle large datasets and its ability to achieve state-of-the-art performance in many machine learning tasks such as classification and regression.

One of the key features of XGBoost is its efficient handling of missing values, which allows it to handle real-world data with missing values without requiring significant pre-processing. Additionally, XGBoost has built-in support for parallel processing, making it possible to train models on large datasets in a reasonable amount of time.

XGBoost can be used in a variety of applications, including Kaggle competitions, recommendation systems, and click-through rate prediction, among others. It is also highly customizable and allows for fine-tuning of various model parameters to optimize performance.

XgBoost stands for Extreme Gradient Boosting, which was proposed by the researchers at the University of Washington. It is a library written in C++ which optimizes the training for Gradient Boosting.

Before understanding the XGBoost, we first need to understand the trees especially the decision tree**:**

Decision Tree:A Decision tree is a flowchart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

A tree can be *"learned"* by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called recursive partitioning. The recursion is completed when the subset at a node all has the same value of the target variable, or when splitting no longer adds value to the predictions.

Bagging:A Bagging classifier is an ensemble meta-estimator that fits base classifiers each on random subsets of the original dataset and then aggregate their individual predictions (either by voting or by averaging) to form a final prediction. Such a meta-estimator can typically be used as a way to reduce the variance of a black-box estimator (e.g., a decision tree), by introducing randomization into its construction procedure and then making an ensemble out of it.
Each base classifier is trained in parallel with a training set which is generated by randomly drawing, with replacement, N examples(or data) from the original training dataset, where N is the size of the original training set. The training set for each of the base classifiers is independent of each other. Many of the original data may be repeated in the resulting training set while others may be left out.

Bagging reduces overfitting (variance) by averaging or voting, however, this leads to an increase in bias, which is compensated by the reduction in variance though.

6.7 Evaluation Metrics in Machine Learning

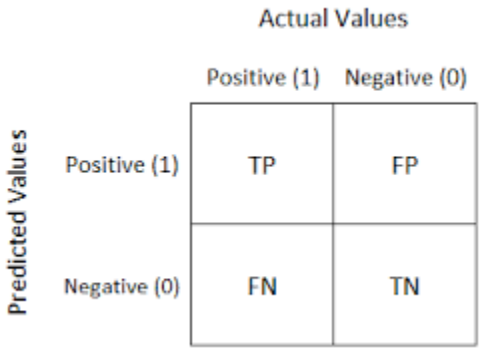
Classification metric

Confusion Matrix

A **confusion matrix** is a matrix that summarizes the performance of a machine learning model on a set of test data. It is a means of displaying the number of accurate and inaccurate instances based on the model's predictions. It is often used to measure the performance of classification models, which aim to predict a categorical label for each input instance.

The matrix displays the number of instances produced by the model on the test data.

- **True positives (TP):** occur when the model accurately predicts a positive data point.
- **True negatives (TN)**: occur when the model accurately predicts a negative data point.
- **False positives (FP)**: occur when the model predicts a positive data point incorrectly.
- **False negatives (FN)**: occur when the model mis predicts a negative data point.



Accuracy:

Model accuracy is not a wholly informative evaluation metric for classifiers. For instance, imagine we run a classifier on a data set of 100 instances. The model's confusion matrix shows only one false negative and no false positives; the model correctly classifies every other data instance. Thus the model has an accuracy of 99%. Though ostensibly desirable, high accuracy is not in itself indicative of excellent model performance. For instance, say our model aims to classify highly contagious diseases. That 1% misclassification poses an enormous risk. Thus, other evaluation metrics can be used to provide a better picture of classification algorithm performance.

Precision and recall:

Precision is the proportion of positive class predictions that actually belong to the class in question.[2] Another way of understanding precision is that it measures the likelihood a randomly chosen instance belongs to a certain class.[3] Precision may also be called *positive predicted value* (PPV).

Recall denotes the percentage of class instances detected by a model.[4] In other words, it indicates the proportion of positive predictions for a given class out of all actual instances of that class.[5] Recall is also known as *sensitivity* or *true positive rate* (TPR)

F1 score:

Precision and recall can share an inverse relationship at times. As a model increases recall by returning more actual class instances (i.e. true positives), the model will inevitably misclassify non-instances (i.e. false positives) as well, thereby decreasing precision.[6] The F1 score attempts to combine precision and recall to resolve this tradeoff.

The F1 score—also called F-score, F-measure, or the harmonic mean of precision and recall—combines precision and recall to represent a model's total class-wise accuracy. Using these two values, one can calculate the F1 score with the equation, where P denotes precision (PPV) and R denotes recall (sensitivity):

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} = \frac{60+100}{60+100+20+20} = \frac{160}{200} = 0.8$$

$$Recall = \frac{TP}{TP+FN} = \frac{60}{60+20} = \frac{60}{80} = 0.75$$

$$Precision = \frac{TP}{TP+FP} = \frac{60}{60+20} = \frac{60}{80} = 0.75$$

$$Specificity = \frac{TN}{TN+FP} = \frac{100}{100+20} = \frac{100}{120} \approx 0.833$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} = 2 \times \frac{0.75 \times 0.75}{0.75+0.75} = 0.75$$

FIG 6.7 - Formulas for Metrics

CHAPTER – 7

RESULTS

7.1 Model Evaluation & Comparison

FIG 7.1.1 - Confusion Matrices

Validation scores:

	KNN	SVC	RFC	XGB
ACC	0.959	0.966	0.977	0.987
AUC	0.954	0.987	0.997	0.999
F1	0.902	0.916	0.943	0.969
F2	0.928	0.931	0.954	0.970

Test scores:

	KNN	SVC	RFC	XGB
ACC	0.966	0.973	0.972	0.983
AUC	0.954	0.992	0.997	0.998
F1	0.916	0.934	0.931	0.958
F2	0.927	0.941	0.945	0.956

FIG 7.1.2 -Metric values

By comparing the results obtained, we see that K-NN is the model that performs the worst and its accuracy is a little lower than Logistic Regression. Despite this, we cannot exclude it a priori, as it still reaches high values for the metrics and, moreover, gives an immediate response. So, we can use it whenever we need to get an idea quickly about the situation and then apply other models when we have more time.

All other models perform better than the benchmark and they obtain high values for the chosen metrics both for validation and test set. SVC and RFC's performances are very similar to each other and XGB performs better than them. If we look at the training phase, SVC and RFC take the same time, while XGB takes more than four times as much as them. So, since the improvement obtained with XGB is only 1.5%, one can choose which model he prefers according to his needs. While the best parameters for multiclass K-NN and SVC are the same as binary classification, for XGB and RFC the Gridsearch for the two types of task returns different parameters. Moreover, in the transition from binary to multiclass problem, the estimated training time remains the same for all models, except for XGB that triples it. In order to understand how features contribute to predictions, let's look at the Permutation Feature Importances for each model

7.2 Failure Type Classification tree

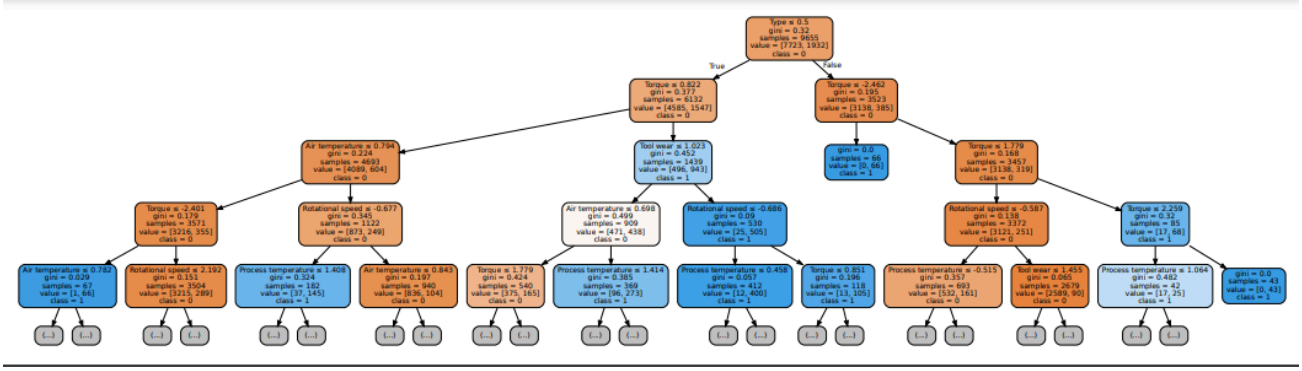


FIG 7.2 - Failure Type Classification tree

Decision Tree:A Decision tree is a flowchart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

The image depicts a decision tree used for classification purposes. Here's a detailed description of the tree:

1. Root Node:
   Type <= 0.15 is the condition at the root.
   The tree splits into two branches based on this condition (True/False).

2. First Split (True Branch):
   Air temperature <= 0.758 is the next condition.
   Further splits into two branches based on this condition.

3. Second Split (Air temperature <= 0.758):
   True Branch:
     Torque <= 3.401 leads to another split.
     Final leaf nodes: (I = 1.0) or (L = 1.0).
   False Branch:
     Rotational speed <= 0.677 leads to another split.
     Final leaf nodes: (I = 1.0) or (L = 1.0).

4. Second Split (Air temperature > 0.758):

Rotational speed <= 0.581 leads to further splits.
Final leaf nodes: (I = 1.0) or (L = 1.0).

5. First Split (False Branch):
Torque <= 3.252 is the next condition.
Further splits into two branches.

6. Second Split (Torque <= 3.252):
True Branch:
Tool wear <= 0.445 leads to further splits.
Final leaf nodes: (I = 1.0) or (L = 1.0).
False Branch:
Rotational speed <= 0.648 leads to further splits.
Final leaf nodes: (I = 1.0) or (L = 1.0).

7. Second Split (Torque > 3.252):
Rotational speed <= 0.587 leads to further splits.
Final leaf nodes: (I = 1.0) or (L = 1.0).

Each internal node of the tree represents a test on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label. The value in each node indicates the number of samples that reach that node, divided into respective classes.

The decision tree's structure shows a hierarchical division of the data based on the conditions set at each node, leading to final classification decisions at the leaves

Here we show the decision paths of one of the trees that make up the Random Forest for both tasks, truncated at depth=4. However this depth is enough to verify that trees require to be deep because the decision boundaries are complex themselves and they are not overfitting. This is evident if one looks at the multi-class tree, where some kinds of failure do not appear before depth four, but also in the binary classification tree by looking at the evolution of the gini score while following most of the paths. A further remark can be made about the feature Type being the origin node of both graphs and separating the majority class (Low quality) from the other two at the first step. It appears just one time more in the upper side of the trees and shows sporadically again at the lowest floors, where its impact is scarce.

CHAPTER – 8

APPLICATIONS

The IoT methodology can be used to develop predictive maintenance tools for the advancement of predictive maintenance in industries such as general machinery, petroleum and petrochemicals, and water treatment. In the area of general machinery, components have usually been repaired on a time-based schedule which leads to unnecessary maintenance repairs and increase in maintenance costs. The petrochemicals industry is facing large growth and with that a need for efficient maintenance practices. One important area of the water resource industry is the desalination of water. In order to keep the cost of desalination down, unscheduled maintenance needs to decrease. Predictive maintenance can address all of these problems and lead to a reduction in unscheduled and unnecessary maintenance.

8.1 General Machinery

Industries related to general machinery include aviation, aerospace, steel industries, paper industries, and other heavy industries. These industries have a lot of maintenance issues related to gears, bearings, shafts, and structure. Previously, these industries have relied on a preventive maintenance strategy based on time. Due to this, maintenance is one of the largest cost drives in these industries with maintenance representing up to 60% of total costs [Mobley, 2002]. Implementing predictive maintenance would help to reduce these costs by decreasing the amount of unnecessary repairs. A demonstration for predictive maintenance in general machinery has been created using a gearbox,

8.2 Petroleum and PetroChemicals

Industries related to this area include energy, gas, oil, and nuclear. The products of this industry provide many of the needed materials for our daily lives from propane to nuclear energy. This area is also experiencing a large growth and with that an increase in maintenance needs. Implementing predictive maintenance can help to address the maintenance issues this industry will face while reducing cost. Two problematic areas in this industry are control valves and process automation. A demonstration related to this industry could be built around a control valve, shown in Figure 3. The control valve would be fitted with various types of sensors. The goal of this demonstration would be to train and educate users on new monitoring techniques and how they could benefit their company. Similar to the demonstration for general machinery, there will be three screens for the user to view during the demonstration. The user will be able to see on the screens how different monitoring techniques can improve the data collected and provide more accurate models.

8.3 Water Treatment

The desalination and purification of water is becoming a more pressing issue as the human population increases and water resources decrease. Water desalination has the potential to be a solution to water shortage problems, but currently it is a very expensive process. Establishing a predictive maintenance program can help to reduce some of these costs by reducing unnecessary downtimes and increasing reliability and availability. For this demonstration, a pump will be used to demonstrate predictive maintenance, shown in Figure 3. Sensors will be used to monitor pressure, temperature, and motor amperage. The goal of this demonstration will be to show how predictive maintenance can ensure that unscheduled maintenance is not a part of the high cost of water desalination. This demonstration would be similar to the setup of the general machinery demonstration.

CHAPTER – 9

ADVANTAGES

1. Longer lifespan of equipment

Well-maintained equipment performs better and lasts longer. Data-driven predictive maintenance reduces wear and tear, minimizes malfunctions, and allows maintenance teams to act quickly when anomalies occur. Predictive maintenance utilizes machine learning (ML) algorithms to evaluate the series of interrelated data points collected by building systems and identifies potential issues that may impact future equipment performance. This extends the life of your equipment, ensures you get the most value out of your equipment, and reduces the frequency of costly capital expenses.

2. Lower maintenance costs

According to the PRSM 2012, HVAC Benchmarking Report, reactive service calls after equipment breaks are, on average, three times as expensive as proactive calls. That's an average of around $400 more per call. When applying a predictive maintenance strategy, these savings are even more substantial by helping to optimize maintenance frequency. When done right, predictive maintenance can all but eliminate unplanned reactive maintenance while still reducing preventive maintenance costs.

3. Less downtime

Predictive maintenance empowers engineering teams to avoid unplanned downtime while also reducing planned downtime. In addition to shortening the downtime of equipment for maintenance, predictive maintenance allows engineering teams to better plan downtimes around usage to minimize disruptions.

4. Enhanced routine maintenance activities

Armed with data in real time, the ML technology behind predictive maintenance can also enhance routine maintenance activities. It does this by providing data-driven fault detection and diagnostics capabilities with an advanced and increasingly accurate assessment of equipment performance. This helps mitigate the time technicians and contractors spend locating and assessing issues by providing root-cause insights into the problem.

5. Budget control

The ability to plan repairs and maintenance in advance gives real estate operations teams enhanced control over maintenance budgets. Equipment service requests can be scheduled at opportune times to maximize cost savings.

CHAPTER – 10

CONCLUSION

According to the analyses carried out and the results obtained, it is possible to make some conclusive considerations related to this project.

We decided to tackle two tasks: predict whether a machine will fail or not and predict the type of failure that will occur. Before developing the models we did data preprocessing to ensure the validity of the assumptions of applicability of the models and ensure the best performances. Briefly, in the preprocessing phase we have deleted some ambiguous samples, we applied a label encoding to the categorical columns and then we performed the scaling of the columns with StandardScaler. We also noticed the presence of some data points which at first we referred to as outliers but later turned out to be part of the natural variance of the data and played an important role in the classification task. Then we ran PCA and found that most of the variance is explained by the first three components, that can be represented as the following features: combination of the two Temperatures, Machine Power (product of Rotational Speed and Torque) and Tool Wear. In accordance with this, we found that these are the features that contribute the most in the predictions when applying the models. Contrary to logical predictions, we demonstrated that the machine's type does not affect the presence of failure.

At the end, we can conclude that for both tasks the chosen models perform very well. For both tasks the best model is XGBoost and the worst is KNN; however the response time of KNN is instant while XGBoost takes more time and this further increases when we proceed with the multi-class classification task. The choice of the model depends on the needs of the company: for faster application one can use KNN while if one cares more about accuracy one can use XGBoost.

CHAPTER – 10

REFERENCES

Gian Antonio Susto et al., “Machine Learning for Predictive Maintenance: A Multiple Classifier Approach,” IEEE Transactions on Industrial Informatics, vol. 11, no. 3, pp. 812-820, 2015.
https://ieeexplore.ieee.org/document/6879441

Ameeth Kanawaday, and Aditya Sane, “Machine Learning for Predictive Maintenance of Industrial Machines Using IoT Sensor Data,” in 2017 8th IEEE International Conference on Software Engineering and Service Science, pp. 87-90, 2017.
https://ieeexplore.ieee.org/document/8342870

Marina Paolanti et al., “Machine Learning Approach for Predictive Maintenance in Industry 4.0,” IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications, pp. 1-6, 2018.
https://ieeexplore.ieee.org/document/8449150

Marwin Zufle et al., “A Predictive Maintenance Methodology: Predicting the Time-to- Failure of Machines in Industry 4.0,” IEEE 19th International Conference on Industrial Informatics (INDIN), pp. 1-8, 2021.
https://ieeexplore.ieee.org/abstract/document/9557387