Predictive Maintenance: Machine Learning Approaches for Detecting and Classifying Types of Machine Failures

A report authored by

Imane Ouada

supervised by

Madam Chaimaa ELALAMI

29 July, 2023

Acknowledgment

I sincerely thank everyone who contributed to this report's completion. Special gratitude goes to Madam Chaimaa ELAlami, my supervisor, for their invaluable guidance. I express my gratitude to M. Salah Ezzahar for supplying the essential information that enabled me to select this project. Heartfelt thanks to my family and friends for constant encouragement.

Abstract

This report presents a Predictive Maintenance project conducted at OCP Jorf Lasfar, focusing on enhancing equipment reliability through Machine type of failure classification. The project involves different steps, including Exploratory Data Analysis (EDA), data cleaning, and visualization. Several machine learning algorithms are evaluated, and the XGBoost classifier emerges as the optimal model. The outcome of the project is MechAlert, an innovative application designed to predict potential machine failures.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In today's dynamic industrial landscape, maintaining the smooth functioning of critical machinery and equipment is crucial to the success of any organization. Traditional maintenance approaches often rely on reactive measures, leading to unexpected breakdowns, costly repairs, and significant downtime. To address these challenges, a paradigm shift towards predictive maintenance has gained prominence. This proactive strategy leverages technology to detect potential failures before they occur, enabling timely interventions and optimized maintenance schedules.

In the domain of predictive maintenance, machine learning (ML) has emerged as a revolutionary tool. Through the analysis of data from sensors, historical maintenance records, and operational parameters, ML algorithms have the ability to reveal anomalies that might signal the emergence of particular failure types. This groundbreaking approach not only improves the dependability and lifespan of industrial equipment but also mitigates operational disruptions and lowers maintenance expenses.

As part of my internship project in OCP Group, I focused on developing a machine learning-based application for the detection of machine failures. The objective was to create a predictive maintenance solution that could identify potential failures in real-time and provide actionable insights to prevent costly breakdowns.

Chapter 2

Company Profile

The OCP Group, also known as the **Office Chérifien des Phosphates**, is a global leader in the phosphate industry. Founded in 1920, the OCP Group is headquartered in Morocco and has established itself as the world's largest exporter of phosphates and phosphate-based products.

The company's primary focus is on the extraction, processing, and marketing of phosphate rock, which is a key ingredient in the production of fertilizers. The OCP Group controls extensive phosphate reserves in Morocco, making it a major player in the global phosphate market. The company also operates subsidiaries and joint ventures in various countries, including Brazil, India, and China, to ensure a wide reach and enhance its global presence.

In addition to phosphate rock, the OCP Group has diversified its operations to include other value-added products such as phosphoric acid and fertilizers. They are involved in the entire production chain, from mining and processing to the development of innovative solutions for sustainable agriculture.

2.1 Corporate timeline

The following table provides a chronological overview of key milestones in the history of the OCP Group, showcasing significant events and achievements that have shaped the company's growth.

Year	Milestone
1920	Office Chérifien des Phosphates founded
1921	Launch of mining production in Khouribga
1931	Launch of mining production in Youssoufia
1965	Launch of chemical production in Safi
1976	Acquisition of 65% of Phosboucraa
1980	Launch of mining production at Benguerir site
1984	Launch of chemical production at Jorf Lasfar
1996	Construction of purified phosphoric acid plant launched in Jorf Lasfar
2002	OCP becomes sole owner of Phosboucraa
2006	Office Chérifien des Phosphates becomes OCP
2008	Transformation of Office Chérifien des Phosphates into a Corporation (OCP S.A)
2011	Joint Venture with Jacobs Engineering
2013	Joint Venture with DuPont de Nemours
2014	Launch of the Slurry Pipeline between Khouribga and Jorf Lasfar
2016	Creation of OCP Africa
2018	Inauguration of Mohammed VI Polytechnic University and Joint Venture with IBM

Table 2.1: Corporate Timeline.

2.2 Jorf Lasfar Chemical Hub

Jorf Lasfar Chemical Hub situated on the Atlantic coast, 20 km southwest of El Jadida, began its production in 1986. This new unit allowed the OCP Group to double its phosphate valorization capacity, with the first site, Morocco-Phosphore I-II, located in the Safi region. The strategic location of the hub was carefully considered, taking into account:

- Proximity to mining areas, ensuring a steady supply of phosphate (from Khouribga).
- Presence of a deep-water port.
- Availability of both seawater and freshwater.
- Availability of land for future expansions.

The Chemical Hub comprises six departments:

✦ **Morocco Phosphore Jorf Lasfar (CIJ) Direction**: Responsible for Maroc Phosphore IIIIV plants, which commenced operations in 1986, producing merchant phosphoric acid and fertilizers.

- ✦ **Euro Maroc Phosphore (CIE) Direction**: Operates the EMAPHOS plant in partnership with PRAYON (Belgium) and C.F BUDENHEIM (Germany), which started its operations in 1998 for purified phosphoric acid production.
- ✦ **Indo Maroc Phosphore (CII) Direction**: Manages the IMACID plant in partnership with the BIRLA and TATA Group (India), which began production of merchant phosphoric acid in 1999. Other functional and technical departments associated with the Chemical Hub include CI/AP, CI/CR, CI/CS, and CI/CT.
- ✦ **PACKPHOS (PMP)**: In partnership with the Pakistani group FAUJI, started its operations in 2007.
- ✦ **BUNGE**: In partnership with the Brazilian group BUNGE FERTILIZI ZANTES, started its operations in 2009.
- ✦ **JFC**: In partnership with the American group Jacobs Engineering SA, with their first site being ODI-P1.

Chapter 3

Project Context

3.1 Predictive Maintenance

Predictive Maintenance is a proactive maintenance strategy that uses data-driven techniques and technologies to monitor the condition and performance of equipment in real-time or near real-time. This involves collecting and analyzing data from sensors, IoT devices, and other monitoring tools to identify patterns and anomalies that could indicate potential failures. It is a crucial aspect of modern industrial operations, aimed at maximizing the uptime of critical assets, minimizing downtime, reducing maintenance costs, and optimizing overall operational efficiency.

3.2 Importance in the Industry

1. **Cost Efficiency**: Predictive Maintenance helps industries move away from traditional reactive maintenance practices, where maintenance is performed only after a failure occurs. By identifying potential issues before they escalate into major breakdowns, companies can avoid costly emergency repairs and reduce overall maintenance expenses.

2. **Minimized Downtime**: Unplanned equipment failures can lead to significant downtime, disrupting production schedules and causing financial losses. Predictive Maintenance enables proactive planning of maintenance activities, ensuring that maintenance interventions are scheduled during planned downtimes or low-demand

periods, thus minimizing the impact on productivity.

3. **Increased Asset Lifespan**: Regular monitoring and early detection of equipment anomalies can prevent further deterioration and extend the operational life of machinery and assets. This optimizes the return on investment for expensive equipment and infrastructure.

4. **Enhanced Safety**: Malfunctioning equipment can pose safety risks to workers and the environment. By predicting and preventing failures, Predictive Maintenance enhances workplace safety and reduces the likelihood of accidents.

5. **Improved Product Quality**: Well-maintained equipment operates more reliably and consistently, leading to better product quality and customer satisfaction. Predictive Maintenance helps in maintaining optimal operating conditions, thereby ensuring consistent production output.

6. **Reduced Unplanned Downtime**: Unplanned downtime can result in production delays and missed deadlines. By predicting potential failures and scheduling maintenance activities accordingly, industries can minimize unplanned downtime and maintain a smooth production flow.

7. **Competitive Advantage**: Companies that adopt Predictive Maintenance gain a competitive edge in the market. They can offer better service levels, higher reliability, and more efficient operations, which attract customers and improve their reputation within the industry.

8. **Sustainable Practices**: By optimizing maintenance schedules and reducing unnecessary interventions, Predictive Maintenance promotes sustainable practices by minimizing energy consumption, reducing waste, and extending the useful life of equipment.

3.3 Project description

The project aims to develop a machine learning-based classification system that can accurately predict and classify different types of machine failures. By utilizing historical data from sensors and monitoring systems, the proposed model will be able to identify patterns and early warning signs of potential machine failures.

Chapter 4

Project Steps

4.1 Exploratory Data Analysis (EDA)

4.1.1 Dataset Overview

The dataset consists of 10 000 data points stored as rows with 10 features in columns

- **UID**: unique identifier ranging from 1 to 10000
- **Type** : consisting of a letter L, M, or H for low (50
- **ProductID**: a combination of Type and the product 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 power of 2860 W, overlaid with a normally distributed noise
- **Torque [Nm]**: torque values are normally distributed around 40 Nm with an f = 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. and a **machine failure** label that indicates, whether

the machine has failed in this particular data point for any of the following failure modes are true

- **Target** : Failure or Not.
- **Failure Type** : Type of Failure.

4.2 Descriptive Statistics

	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]
count	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000
mean	300.004930	310.005560	1538.776100	39.986910	107.951000
std	2.000259	1.483734	179.284096	9.968934	63.654147
min	295.300000	305.700000	1168.000000	3.800000	0.000000
25%	298.300000	308.800000	1423.000000	33.200000	53.000000
50%	300.100000	310.100000	1503.000000	40.100000	108.000000
75%	301.500000	311.100000	1612.000000	46.800000	162.000000
max	304.500000	313.800000	2886.000000	76.600000	253.000000

Figure 4.1: descriptive statistics

	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]
skewness	0.114257	0.015025	1.992872	-0.009515	0.027288
kurtosis	-0.836144	-0.500084	7.388649	-0.013834	-1.166754

Figure 4.2: Skewness and Kurtosis

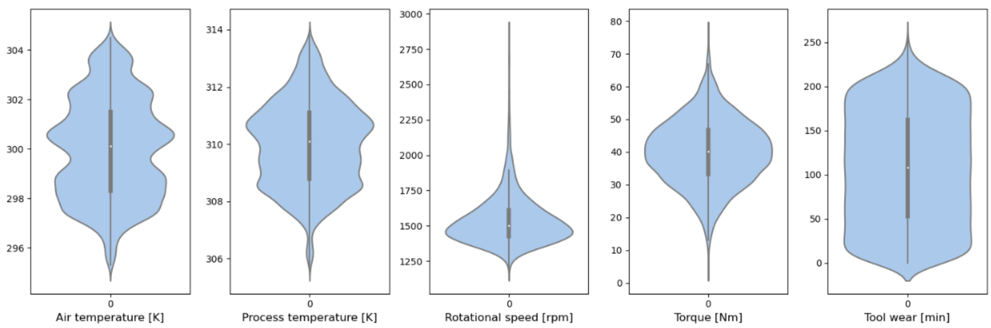

Figure 4.3: ViolinPlot

From the tables and the violin plots above, we can draw the following conclusions related to central tendency, variability, skewness, range, percentiles, kurtosis and outliers

- Air Temperature :

1. **Central Tendency**: The mean temperature (300.0049 K) and the median (300.1 K) are relatively close, indicating that the data is approximately symmetrically distributed around the central value. This suggests that the dataset may follow a roughly normal distribution.
2. **Variability**: The standard deviation (2.0002 K) is relatively small compared to the range of temperatures (295.3 K to 304.5 K). This suggests that the temperatures are relatively close to the mean, indicating less variability in the dataset.
3. **Range**: The temperature data spans a range from a minimum of 295.3 K to a maximum of 304.5 K. This gives an idea of the spread of temperatures within the dataset.
4. **Skewness**: The skewness value is positive (0.114), indicating that the distribution is slightly positively skewed. This means that the distribution has a longer tail on the right side, and the majority of data is concentrated on the left side of the central peak.
5. **Kurtosis**: A kurtosis value of -0.836 indicates that distribution is platykurtic.

- Process temperature :

  1. **Central Tendency**: The mean (average) temperature is approximately 310.01 K, and the median (50th percentile) temperature is 310.1 K. This suggests that the data is roughly symmetrically distributed around the central value.
  2. **Variability**: The standard deviation of 1.4837 K indicates the amount of variability or dispersion in the data.
  3. **Range**: The temperature range is from a minimum of 305.7 K to a maximum of 313.8 K. This shows the full extent of the data.
  4. **Percentiles**: The 25th percentile (Q1) is 308.8 K, and the 75th percentile (Q3) is 311.1 K. These percentiles help us understand the spread of the data and identify the values below which certain percentages of the data fall. For instance, Q1 indicates that 25% of the temperatures are below 308.8 K, and Q3 shows that 75% of the temperatures are below 311.1 K.

5. **Kurtosis**: A kurtosis value of -0.50 indicates that distribution is platykurtic.

- Rotational Speed rpm :

1. **Central Tendency**: The mean rotational speed is approximately 1538.78 rpm, and the median (50th percentile) rotational speed is 1503 rpm. The mean is larger than the median, suggesting that the data is right skewed.
2. **Variability**: The standard deviation of 179.28 rpm indicates the amount of variability or dispersion in the data.
3. **Range**: The rotational speed range spans from a minimum of 1168 rpm to a maximum of 2886 rpm. This shows the full extent of the data.
4. **Percentiles**: The 25th percentile (Q1) is 1423 rpm, and the 75th percentile (Q3) is 1612 rpm. These percentiles help us understand the spread of the data and identify the values below which certain percentages of the data fall. For instance, Q1 indicates that 25% of the rotational speeds are below 1423 rpm, and Q3 shows that 75% of the rotational speeds are below 1612 rpm.
5. **Skewness**: The skewness value is significantly positive (1.993), indicating that the distribution is positively skewed. The data is concentrated on the left side of the central peak, with a long tail on the right side.
6. **Kurtosis**: A kurtosis value of 7.388 indicates that the distribution is leptokurtic.
7. **Outliers**: Notably, there are outliers in the violinplot with values greater than 2250 and we choose not to remove them from the dataset. Many of these outliers are associated with anomalies in the machine or rare occurrences that are our aim. Removing them could lead to a loss of critical insights into potential issues.

- Torque :

1. **Central Tendency**: The mean torque is approximately 39.99 Nm, and the median (50th percentile) torque is 40.1 Nm. The mean and median are quite close, indicating that the data is approximately symmetrically distributed around the central value.

2. **Variability**: The standard deviation of 9.97 Nm indicates the amount of variability or dispersion in the data.

3. **Range**: The torque range extends from a minimum of 3.8 Nm to a maximum of 76.6 Nm. This demonstrates the full extent of the data.

4. **Percentiles**: The 25th percentile (Q1) is 33.2 Nm, and the 75th percentile (Q3) is 46.8 Nm. These percentiles help us understand the spread of the data and identify the values below which certain percentages of the data fall. For instance, Q1 indicates that 25% of the torque measurements are below 33.2 Nm, and Q3 shows that 75% of the torque measurements are below 46.8 Nm.

5. **Skewness**: The skewness value is very close to zero (-0.0095), indicating that the distribution is almost symmetric. The data is distributed fairly evenly around the central peak.

6. **Kurtosis**: A kurtosis value of -0.0138 indicates that the distribution is platykurtic.

7. **Outliers**: Notably, there are outliers in the violin plot with values greater than 70 Nm and values less than 10 Nm and we choose not to remove them from the dataset. All these outliers are associated with anomalies in the machine or rare occurrences that are our aim. Removing them could lead to a loss of critical insights into potential issues.

- Tool wear :

1. **Central Tendency**: The mean time duration is approximately 107.95 minutes, and the median (50th percentile) time duration is 108 minutes. The mean and median are quite close, indicating that the data is approximately symmetrically distributed around the central value.

2. **Variability**: The standard deviation of 63.65 minutes indicates the amount of variability or dispersion in the data.

3. **Range**: The time duration range spans from a minimum of 0 minutes to a maximum of 253 minutes. This shows the full extent of the data.

4. **Percentiles**: The 25th percentile (Q1) is 53 minutes, and the 75th percentile (Q3) is 162 minutes. These percentiles help us understand the spread of the data and identify the values below which certain percentages of the data fall. For instance, Q1 indicates that 25

5. **Skewness**: The skewness value is positive (0.027), indicating that the distribution is slightly positively skewed. This means that the distribution has a longer tail on the right side, and the majority of data is concentrated on the left side of the central peak.

6. **Kurtosis**: A value of -1.166 indicates that the distribution is platykurtic

4.3 Distribution of Unique Values

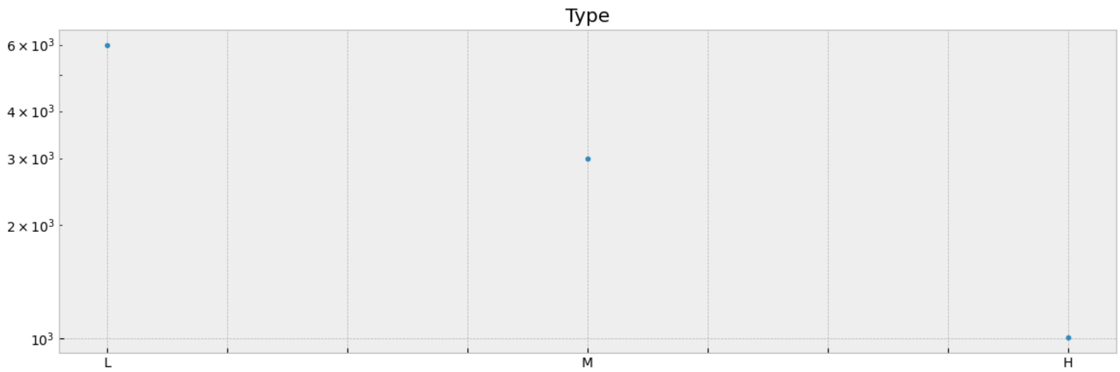

Figure 4.4: Distribution of values of the variable Type

The provided non-numerical features have been categorized into three groups: "L," "M," and "H," with approximately 6,000, 3,000, and 1,000 occurrences, respectively. This indicates that the "L" category is the most frequent, followed by "M," while "H" is the least common. The distribution highlights imbalanced data and the presence of rare categories. The bar plot indicates that the "Rotational Speed" and "Torque" features have a relatively high number of unique values, approximately around 1,000. This suggests that these two features have a broad range of data points, possibly representing continuous or fine-grained measurements with considerable variability.

On the other hand, the "Tool Wear" feature exhibits a lower number of unique values, falling . This indicates that the "Tool Wear" data may be more discrete or categorized.

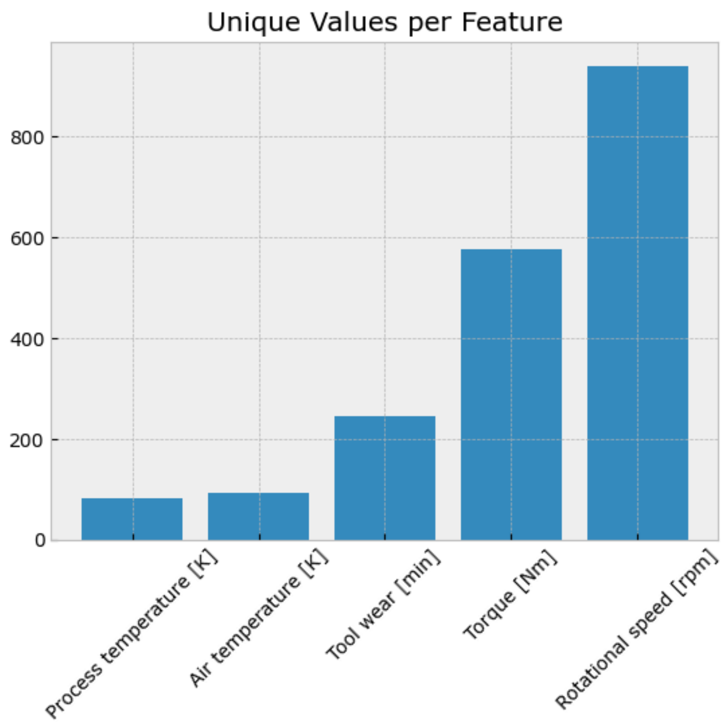

Figure 4.5: Unique values per feature

Both "Process Temperature" and "Air Temperature" features have a relatively limited number of unique values, approximately around around 100. This suggests that these two features have a more constrained range.

4.4 Pairplot analysis

In this pair plots analysis, we explored the relationships between various numerical variables in our dataset. The plot displays a grid of scatter plots, where each cell represents the correlation between two different variables. The diagonal cells show density plots for each individual variable, while the off-diagonal cells depict the scatter plots.

Upon initial examination, we observed a positive linear relationship between "Process Temperature" and Air Temperature," indicating that as Process Temperature increases, so does the the air Temperature. Additionally, we noticed a clustering pattern in the "Torque" and "Process Temperature" scatter plot.

Furthermore, the pair plot revealed some interesting insights between "Torque" and "Rotational Speed." The scatter plot displayed a negative exponential like relationship, where rotational speed decreases at an exponential rate as the torque increases.

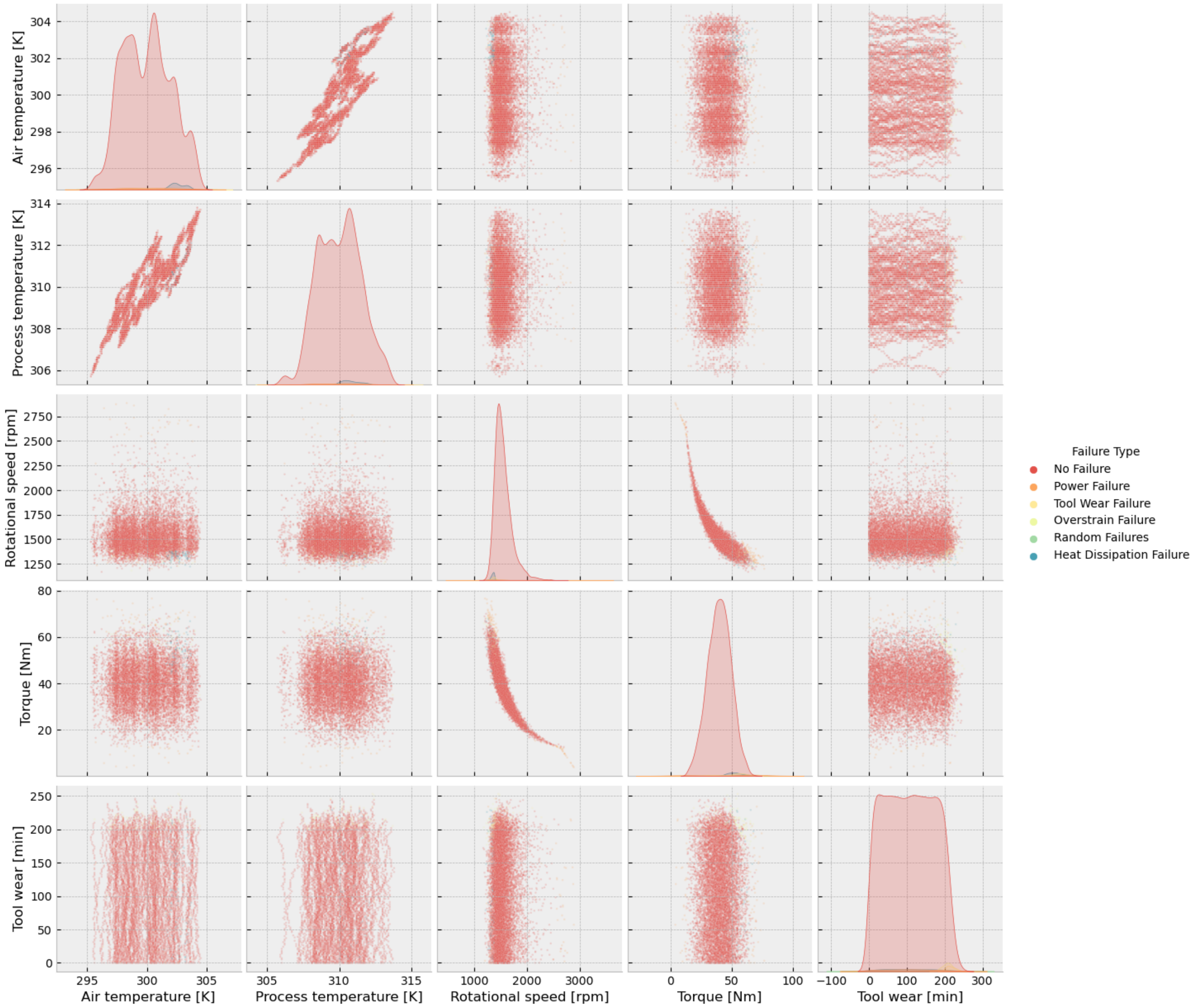

Figure 4.6: Pairplot

4.5 3D PCA visualization

Just glancing at the plot, we can see that there are mainly 3 clusters of Failure Type.

It appears that the data points are arranged in a V-shaped pattern, with the lower part representing a cluster of non-failure type, and the sides representing Tool Wear Failure . Additionally, there's a small cluster between the sides that represents 'Power Failure'.

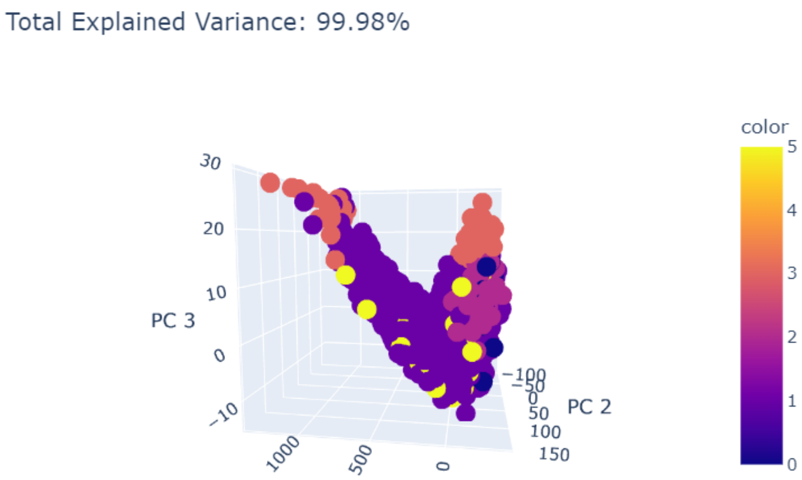

Figure 4.7: 3D PCA visualization

Chapter 5

Machine Learning Algorithms

In our project, we have explored multiple machine learning algorithms to detect failure types. Among the algorithms we experimented with are K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Decision Tree, Random Forest, and XGBoostClassifier.

5.1 K-nearest neighbors

The k-nearest neighbors algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. While it can be used for either regression or classification problems, it is typically used as a classification algorithm, working off the assumption that similar points can be found near one another.

The general principle of the k-nearest neighbors (KNN) algorithm revolves around the idea that data points with similar features tend to belong to the same class or have similar output values. KNN makes predictions based on the assumption that neighboring data points in the feature space share some underlying patterns or characteristics. To classify or predict a new data point, KNN identifies the 'k' closest data points from the training dataset using a chosen distance metric, where 'k' is a hyperparameter set by the user. The algorithm then takes a majority vote (for classification) or calculates the average (for regression) of the output labels or target values of these 'k' neighbors to determine the prediction for the new data point. KNN's simplicity and effectiveness make it a widely used algorithm, but it can be sensitive to the choice of distance metric and the value of 'k', requiring careful consideration during model selection and tuning.

5.2 Support Vector Machine

Support Vector Machine is a supervised machine learning algorithm used for classification and regression tasks. SVM finds the optimal hyperplane that best separates data points of different classes, maximizing the margin between them. The data points closest to the hyperplane are known as support vectors, hence the name. SVM is widely used for its ability to handle high-dimensional data and its effectiveness in handling both linear and non-linear classification problems.

The general principle of Support Vector Machine (SVM) centers on finding an optimal decision boundary, known as the hyperplane, that effectively separates data points belonging to different classes in a feature space. The primary objective of SVM is to maximize the margin between the hyperplane and the nearest data points of each class. This margin represents the region of confidence, ensuring a clear distinction between classes and enhancing the generalization of the model to unseen data. SVM is commonly used for binary classification, where it aims to identify the best hyperplane that correctly divides the data into two classes, but it can also be extended to handle multi-class classification tasks. To deal with non-linearly separable data, SVM employs the kernel trick, which implicitly maps the original feature space into a higher-dimensional space, where linear separation becomes feasible. SVM's ability to handle complex datasets and its strong theoretical foundation make it a popular choice in various machine learning applications.

5.3 Decision Trees

A decision tree is a non-parametric supervised learning algorithm, which is utilized for both classification and regression tasks. It has a hierarchical, tree structure, which consists of a root node, branches, internal nodes and leaf nodes.

The general principle of decision trees revolves around creating a hierarchical and tree-like model that systematically partitions the data into subsets based on the values of input features. The decision tree starts with a root node, representing the entire dataset, and iteratively selects the most significant feature to split the data into two or more branches.

This process is based on selecting the feature that best separates the data and maximizes information gain (for classification) or minimizes impurity (for regression). Each internal node of the tree represents a decision based on a feature, and each leaf node represents a predicted output value or class label. The construction of the tree continues until a stopping criterion is met, such as reaching a maximum depth or having a minimum number of samples in the leaf nodes. Decision trees are interpretable, easy to visualize, and capable of capturing complex relationships in the data, making them valuable tools for various machine learning applications. However, they can be prone to overfitting, which can be addressed through techniques like pruning and using ensemble methods like random forests.

5.4 Random Forests

Random Forests are an ensemble learning method in machine learning that operates on the principle of combining multiple decision trees to make accurate predictions.

The general principle of Random Forests is based on the concept of "wisdom of the crowd." Instead of relying on a single decision tree, this technique constructs a large number of diverse decision trees, each trained on different subsets of the data and using random feature subsets. During prediction, the individual trees vote on the outcome, and the final prediction is determined by the majority vote. This process reduces overfitting and improves generalization, as the ensemble model tends to be more robust and less sensitive to noise in the data. Random Forests excel in handling high-dimensional datasets, providing feature importance rankings, and delivering robust and accurate predictions for both classification and regression tasks.

5.5 Gradient Boosting

Boosting algorithms combine weak learners, i.e. learners slightly better than random, into a strong learner in an iterative way. Gradient boosting is a boosting-like algorithm for regression. Given a training dataset $D = \{x_i, y_i\}_1^N$, the goal of gradient boosting is to find an approximation, $F(x)$, of the function $F^*(x)$, which maps instances x to their output values y, by minimizing the expected value of a given loss function, $L(y, F(x))$. Gradient boosting builds an additive approximation of $F^*(x)$ as a weighted sum of functions

$$F_m(x) = F_{m-1}(x) + \rho_m h_m(x)$$

where ρ_m is the weight of the m^{th} function $h_m(x)$. these functions are the models of the ensemble (decision trees). The approximation is constructed iteratively. First, a constant approximation of $F^*(x)$ is obtained as:

$$F_0(x) =_\alpha \sum_{i=1}^{N} L(y_i, \alpha)$$

Subsequent models are expected to minimize

$(\rho_m, h_m(x)) =_{\rho,h} \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) + \rho h(x_i))$

However, instead of solving the optimization problem directly, each h_m can be seen as a greedy step in a gradient descent optimization for F^*. For that, each model, h_m, is trained on a new dataset $D = \{x_i, r_{mi}\}_{i=1}^{N}$, where the pseudo- residuals, r_{mi}, are calculated by:

$$r_{mi} = [\frac{\partial L(y_i, F(x))}{\partial F(x)}]_{F(x)=F_{m-1}(x)}$$

The value of ρ_m is subsequently computed by solving a line search optimization problem.

This algorithm can suffer from over-fitting if the iterative process is not properly regularized. For some loss functions (e.g. quadratic loss), if the model h_m fits the pseudo-residuals perfectly, then in the next iteration the pseudo-residuals become zero and the process terminates prematurely. To control the additive process of gradient boosting, several regularization parameters are considered. The natural way to regularize gradient boosting is to apply shrinkage to reduce each gradient decent step $F_m(x) = F_{m-1}(x) + \nu\rho_m h_m(x)$ with $\nu = (0, 1.0]$.

The value of ν is usually set to 0.1. In addition, further regularization can be achieved by limiting the complexity of the trained models. For the case of decision trees, we can limit the depth of the trees or the minimum number of instances necessary to split a node. Contrary to random forest, the default values for these parameters in gradient boosting are set to harshly limit the expressive power of the trees (e.g. the depth is generally limited to 35). Finally, another family of parameters also included in the different versions of gradient boosting are those that randomize the base learners, which can further improve the generalization of the ensemble , such as random subsampling without replacement.

5.5.1 XGBoost

XGBoost is a decision tree ensemble based on gradient boosting designed to be highly scalable. Similarly to gradient boosting, XGBoost builds an additive expansion of the objective function by minimizing a loss function. Considering that XGBoost is focused only on decision trees as base classifiers, a variation of the loss function is used to control the complexity of the trees.

$$L_{xgb} = \sum_{i=1}^{N} L(y_i, F(x_i)) + \sum_{m=1}^{M} \Omega(h_m)$$
$$\Omega(h) = \gamma T + \frac{1}{2}\lambda \|w\|^2$$

where T is the number of leaves of the tree and ware the output scores of the leaves. This loss function can be integrated into the split criterion of decision trees leading to a pre-pruning strategy. Higher values of γ result in simpler trees. The value of γ controls the minimum loss reduction gain needed to split an internal node. An additional regularization parameter in XGBoost is shrinkage, which reduces the step size in the additive expansion. Finally, the complexity of the trees can also be limited using other strategies as the depth of the trees, etc. A secondary benefit of tree complexity reduction is that the models are trained faster and require less storage space. Furthermore, randomization techniques are also implemented in XGBoost both to reduce overfitting and to increment training speed. The randomization techniques included in XGBoost are: random subsamples to train individual trees and column subsampling at tree and tree node levels. In addition, XGBoost implements several methods to increment the training speed of decision trees not directly related to ensemble accuracy. Specifically, XGBoost focuses on reducing the computational complexity for finding the best split, which is the most time-consuming part of decision tree construction algorithms. Split finding algorithms usually enumerate all possible candidate splits and select the one with the highest gain. This requires performing a linear scan over each sorted attribute to find the best split for each node. To avoid sorting the data repeatedly in every node, XGBoost uses a specific compressed column based structure in which the data is stored pre-sorted. In this way, each attribute needs to be sorted only once. This column based storing structure allows to find the best split for each considered attributes in parallel. Furthermore, instead of scanning all possible candidate splits, XGBoost implements a method based on percentiles of the data where only a subset of candidate splits is tested and their gain is computed using aggregated statistics.

Chapter 6

Results

We present the comprehensive results of our model evaluation, obtained through using either K-fold cross-validation or a hold-out 20% testing set . To optimize the performance of our model, we utilized a systematic Grid Search technique to fine-tune hyperparameters.

Algorithm	Cross-Validation	Parameters	Accuracy
KNN	K fold	metric : manhattan, n_neighbors: 9, weights: distance	0.97
Decision Tree	K fold	default	0.971
Random Forest	K fold	max_depth: 3, n_estimators: 139	0.969
Linear SVM	hold out	default	0.98
Rbf SVM	hold out	default	0.9685
XGBoost	hold out	default	0.985

Table 6.1: Results Table

Chapter 7

Developing MechAlert

MechAlert is a user-friendly tool that uses machine learning to predict machinery maintenance needs. Users can input data or upload a CSV file for analysis. By employing the XGBoost Classifier, MechAlert helps users make informed decisions on preventive maintenance. This minimizes equipment failures, reduces downtime, and boosts efficiency. The tool was designed using Streamlit, ensuring a smooth and accessible experience for all users. Additionally, MechAlert is hosted on Render, providing a reliable and scalable platform for seamless accessibility and usage.

In the following section, we present images of the app's user interface to provide a visual representation of its features and functionalities.

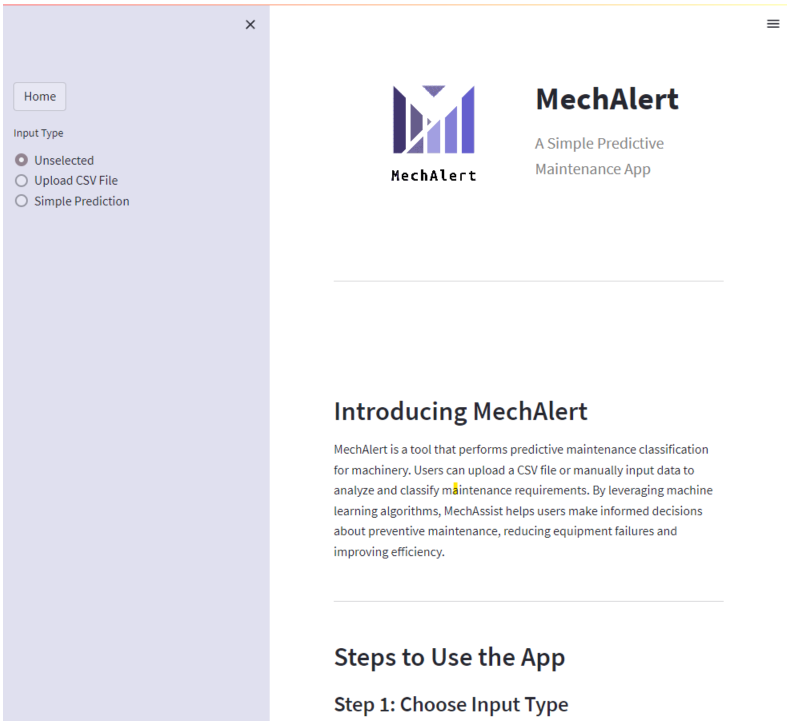

Figure 7.1: MechAlert

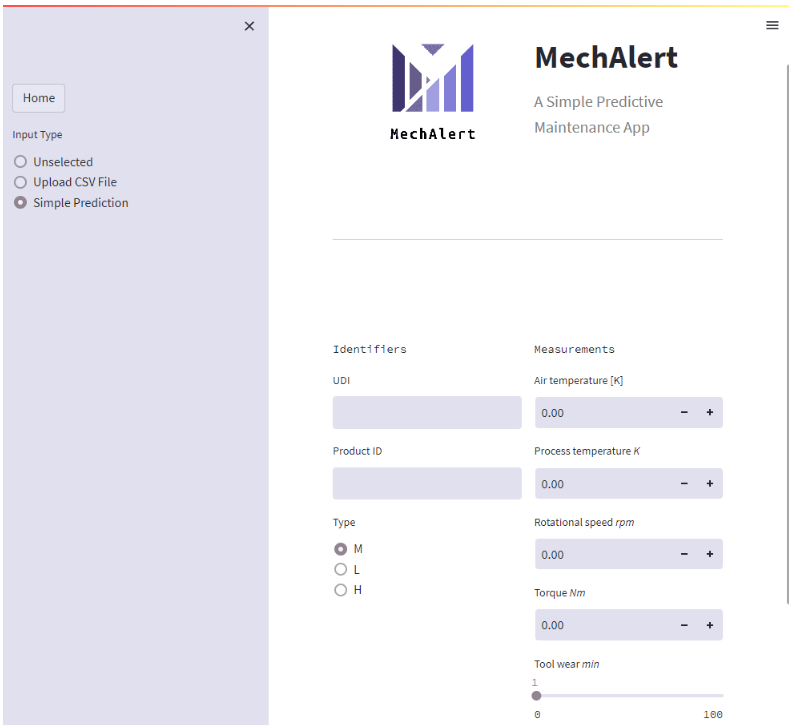

Figure 7.2: MechAlert

Chapter 8

Conclusion

In conclusion, the summer internship project focused on the development of MechAlert, a predictive maintenance system for OCP Jorf Lasfar. The primary objective was to enhance equipment reliability and optimize maintenance practices using data analytics and machine learning techniques.

Predictive maintenance was identified as a crucial strategy in the industry, offering numerous benefits such as cost efficiency, minimized downtime, increased asset lifespan, improved safety, and enhanced product quality.

The project's success was achieved through a systematic approach, starting with exploratory data analysis (EDA) and data cleaning to ensure the dataset's integrity. Descriptive statistics and data visualization provided valuable insights into the data's distribution, variability, and potential patterns, guiding the subsequent analysis.

Multiple machine learning algorithms, including K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Decision Tree, Random Forest, and XGBoost Classifier, were evaluated to detect failure types accurately. Among these, the XGBoost Classifier proved to be the optimal choice due to its superior performance in predicting potential machine failures.

MechAlert, the final outcome of the project, is a cutting-edge application capable of predicting and classifying different types of machine failures. By leveraging historical data, sensor readings, and machine parameters, MechAlert identifies early warning signs and offers actionable insights for timely maintenance interventions.