Annexure-I: Title Page

Predictive Maintenance Classification Using Machine Learning: A Synthetic Data Approach

A Project Report

Submitted in partial fulfilment of the requirements for the award of degree of

Bachelor of Technology

(Computer Science Engineering)

Submitted to

LOVELY PROFESSIONAL UNIVERSITY

PHAGWARA, PUNJAB

From 1st Sep 2024 to 25th Oct 2024

SUBMITTED BY

Name of student: Kedhareswer Naidu

Roll Number: RK21UWA32

Reg Number: 12110626

Faculty: Sajjad Manzoor Mir

Section: K21UW

Annexure-II: Student Declaration

To whom so ever it may concern

I, **Kedhareswer Naidu, 12110626**, hereby declare that the work done by me on **"Predictive Maintenance Classification Using Machine Learning: A Synthetic Dataset Approach"** from Sep 2024 to Oct 2024, is a record of original work for the partial fulfilment of the requirements for the award of the degree, Bachelor of Technology.

Name of the student: Kedhareswer Naidu

Registration Number: 12110626

Dated: 25TH October 2024

ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to the teacher and instructor of the course Machine Learning who provided me the golden opportunity to learn a new technology.

I would like to also thank my own college Lovely Professional University for offering such a course which not only improve my programming skill but also taught me other new technology.

Then I would like to thank my parents and friends who have helped me with their valuable suggestions and guidance for choosing this course.

Finally, I would like to thank everyone who have helped me a lot.

Dated: 25TH October 2024

Table of Contents

ABSTRACT:

In industrious environments, undramatic machine down-time endures enormous costs on operations and productivity losses. **Predictive maintenance with machine learning** would, therefore, thus provide an upstream approach whereby failures are predicted even before it happens and intervenes in time. This project concerns itself with the classification of maintenance needs done on a synthetic dataset based on *operational features and failure modes*. The dataset comprises 10,000 data instances with 14 features that simulate *real sensor data for temperature, rotational speed, torque, product quality metrics, and failure types.*

We used multiple machine learning models, such as *Random Forest*, *Decision Tree*, and *Support Vector Machines (SVM)*, to identify trends and predict system failures. The key to our approach was quite holistic exploratory data analysis (EDA) to capture the feature correlation and tackle the data imbalances so prevalent in predictive maintenance datasets. We ensured maximization of accuracy and reliability in failure prediction by evaluation across many metrics.

Our experiments demonstrate how the more complex ensemble methods- Random Forest and SVM- will closely approximate near perfect classification accuracy for failure and are interpretative.

OBJECTIVE:

This project aims to design a machine learning model that will predict machine failure based on different operational and environmental factors for the purposes of timely intervention on maintenance to minimize unplanned downtimes. Specific objectives are,

1. Failure Prediction: Classify whether a machine is likely to fail based on operational features such as temperature, rotational speed, torque, and tool wear.

2. Identify the exact failure mode when a failure occurred in order to aim maintenance at specific failure modes.

3. Optimization of Maintenance: Predicting failures allows for scheduling maintenance only when necessary, reducing unnecessary servicing while avoiding breakdowns.

4. Cost Reduction: By preventing unexpected failures, companies can reduce the financial burden associated with downtime, emergency repairs, and machine replacement.

5. Data-Driven Insights: The analysis of features such as air temperature, process temperature, rotational speed, torque, and tool wear provides insights into the operational parameters that most influence machine performance and failure rates.

Overall, the goal is to create a machine learning model that can predict failures effectively, improving operational efficiency and reducing maintenance costs

INTRODUCTION:

1. Problem Statement:

In industrial environments, machine failures cause much operational downtime, high repair costs, and low productivity. In an effort to prevent such machine failures, routine maintenance often proves to result in unwarranted service interventions, while reactive maintenance results in expensive emergency repairs. Predictive maintenance, on the other hand, actually predicts when such failures are expected to occur so that companies may plan maintenance only at critical times.

Development of the Machine Learning Model This project aims to develop a model to predict whether a machine is going to fail or not based on sensor reading values and other operating parameters. Moreover, for if it predicts failure, it can classify the type of failure to make diagnosis and rectification efficient.

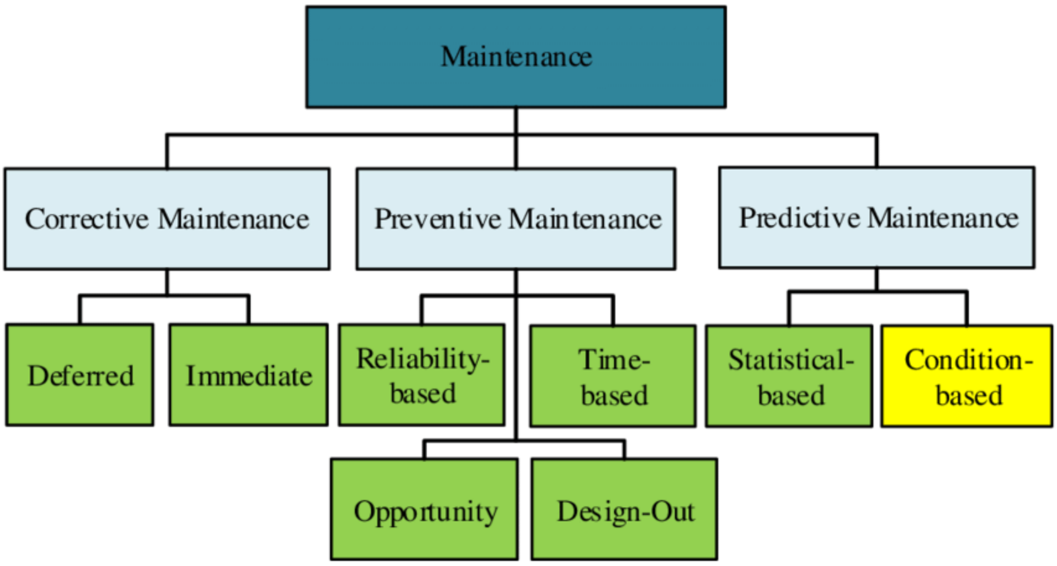
Records There are 10,000 records with 14 attributes: air temperature, process temperature, rotational speed, torque, and tool wear.
Targets There are two targets:
(1) Failure or not, and
(2) Failure Type.
In other words, the models are supposed to classify machine failures with a reasonable degree of accuracy without suffering from data leakage thereby providing actionable insights in addition to process improvement in the maintenance process.

2. Background:

Unforeseen failures in the machine cause enormous losses financially, lower productivity, and even a danger to safety when in industrial settings. Conventional maintenance approaches of this nature involve repair maintenance-ongoing correcting of a machine after it has broken down-involving massive costs due to forced idle time-and preventive maintenance-replacement at set intervals without consideration for the need based on the status of the machine-tend to lack optimum efficiency during work.

Predictive Maintenance is an *emerging approac*h whereby it uses data-driven insights to predict the failure of equipment even before it happens. These pattern analyses identify, across *historical maintenance records* and the *operational conditions* compiled from *machine sensors*, potential early failure indicators. Predictive maintenance will be able to identify and pin down problems most likely to occur and determine the equipment's remaining useful life, enabling maintenance at the most opportune time-before it breaks down. This saves money, prevents unexpected breakdowns, and prolonged equipment life.

Predictive Maintenance Through Machine Learning: Represents the technology used as the heart of predictive maintenance, allowing for the accurate analysis of extensive data streams generated by machine sensors and operational environments. Such patterns can be discovered by complex algorithms; therefore, ML models become able to classify potential failures, assess the RUL, and select failure modes-existing, electrical failure, mechanical wear-but primarily based on historical and current data. However, collection of heavy, quality datasets for PdM from the real environment is very difficult due to proprietary restrictions, variability in machinery, and data privacy issues. It has led to the adoption of synthetic datasets, like in this project, as a practical means to simulate the machine behaviors, failures, and operational conditions.



Project Context and Scope: In this project, we utilize a synthetic dataset that mimics real-world predictive maintenance scenarios. This dataset includes features such as air and process temperatures, rotational speed, torque, tool wear, and failure modes, designed to simulate realistic operational characteristics. The goal is to leverage this dataset to develop and evaluate machine learning models capable of classifying machine failures and identifying failure types. By understanding the patterns and trends that lead to failures, the project seeks to provide a foundation for real-time predictive maintenance systems in industrial applications.

By applying various machine learning models to classify failures and conducting exploratory data analysis (EDA) to better understand feature relationships, this project aims to demonstrate the effectiveness of predictive maintenance solutions. Although synthetic data is used, the insights gained can inform future developments in predictive maintenance models and set the stage for transitioning to real-world datasets, ultimately enabling more resilient, efficient, and proactive maintenance strategies in industrial environments.

THEORETICAL BACKGROUND:

1. What is **Predictive Maintenance (PdM)?**

Predictive Maintenance or PdM describes a data-driven methodology of monitoring equipment conditions to predict possible equipment failures before they actually happen.

Since it is not time-based like preventive maintenance, PdM relies on data provided by sensors and other sources to predict the health status of equipment to enable scheduled maintenance just in time before breakdown, thus optimizing schedules for maintenance, time waste, and cost savings.

Predictive maintenance leverages data from multiple sources, including sensor data (e.g., temperature, vibration, pressure), operational conditions (e.g., load, speed), and historical maintenance logs. The quality and relevance of data are critical, as different sensor readings reflect different aspects of the equipment's condition and potential failure modes.

2. What is **Failure Classification and Failure Modes?**

Binary Classification: Failure classification in PdM makes a prediction of whether a failure will occur or not based on features derived from machinery data. Typically, the result of a binary classification would be "Failure" or "No Failure".

3. **What are Machine Learning Algorithms for Predictive Maintenance?**

Heavy reliance in predictive maintenance would be placed on the machine learning algorithms to classify potential failure with good accuracy from the sensor and operational data. The project preferred classification models as most suitable for high-dimensional, noisy, and often imbalanced data seen in maintenance datasets.

- **Random Forest (RF):** An ensemble method which generates multiple decision trees and merges them into a very robust generalised classification of phenomena. RF is very respected in the field of PdM since it can be used for discovery of a non-linear relationship as well as carry resilience against overfitting.

- **Support Vector Machine (SVM):** SVM is also efficient in a high-dimensional space and helpful when data are not linearly separable. SVM finds out the optimal hyperplane for proper separation of classes, which is perfectly suited to complex datasets with likely overlap between classes "Failure" and "No Failure".

- **XGBoost or Extreme Gradient Boosting**: An efficient and scalable gradient-boosting algorithm, it is highly favorable for predictive maintenance tasks since it achieves a high performance in handling complex data patterns and builds reductions through iterative improvements.

- **Deep learning models:** For time series scenarios, the LSTM networks or GRUs are usually set to represent dependencies across the time horizon in sequential data. They were not applied in this project since there was no temporal data, although they are widely used in predictive maintenance to monitor the degradation of equipment over time.

4. Synthetic Data and Its Challenges?

- **Synthetically generated data:** Predictive maintenance data is generally not readily available in the real world due to limitations placed by proprietary restrictions and privacy restrictions. The synthetically generated data used in this project is patterned after actual industrial conditions, but it allows for models to be developed without access to sensitive data.

- **Limitations:** Although realistic conditions may be simulated with synthetic data, it is unlikely to very well capture real-world complications concerning operational issues, like variation of machine conditions, the practices of maintenance, or unexpected operation scenarios. Models thus trained on synthetic data have to be adapted when applied to real-world settings.

Hardware & Software Requirements:

Hardware:

Predictive maintenance projects can be resource-intensive, especially for model training on large datasets. The requirements may vary based on the size of the dataset and complexity of the model, but here are some general recommendations:

1. **Processor (CPU)**:
   - Minimum: Quad-core processor (Intel i5 or AMD Ryzen 5)
   - Recommended: Multi-core processor (Intel i7 or AMD Ryzen 7 and above)
2. **Graphics Processing Unit (GPU)** (for deep learning models or larger datasets):
   - Minimum: NVIDIA GTX 1050 (2 GB VRAM)
   - Recommended: NVIDIA RTX 2060 (6 GB VRAM) or higher for better performance, such as the NVIDIA RTX 3090 (24 GB VRAM) for very large datasets or complex neural networks.
3. **RAM**:
   - Minimum: 8 GB
   - Recommended: 16 GB or higher
   - Sufficient memory helps handle large datasets and improves model training and analysis speed, especially when working with high-dimensional sensor data.

4. **Storage**:
   - Minimum: 256 GB SSD (for faster data access and processing)
   - Recommended: 512 GB SSD or higher, especially if working with larger datasets or multiple versions of synthetic datasets.

Software:

1. **Operating System**:
   - Windows 10 or higher, macOS, or Linux (Ubuntu is widely used in data science and machine learning)

2. **Programming Language**:
   - **Python 3.8 or higher**: Python is widely used in machine learning due to its

comprehensive libraries for data processing, model building, and visualization.

3. **Integrated Development Environment (IDE)**:
   - **Jupyter Notebook** (recommended for interactive data exploration and EDA)
   - **Anaconda** (a popular Python distribution that includes Jupyter Notebook, making it easier to manage packages and dependencies)
   - **VS Code, PyCharm, or Spyder** (optional; these are more suited for code management in larger projects)

4. **Key Python Libraries**:
   - **Data Processing and Analysis**:
     - pandas: for data manipulation and analysis
     - numpy: for numerical computations
   - **Exploratory Data Analysis and Visualization**:
     - matplotlib and seaborn: for data visualization
     - plotly: for interactive plots (optional)
   - **Machine Learning and Model Training**:
     - scikit-learn: for traditional machine learning algorithms, data preprocessing, and evaluation metrics

5. **Additional Tools for Documentation and Reporting**:
   - **Microsoft Excel or Google Sheets**: for simple data exploration, report organization, and result presentation
   - **Markdown** (e.g., in Jupyter Notebooks or GitHub): for documentation and creating a clear project report
   - **PowerPoint or Google Slides**: for presenting findings and sharing results with stakeholders

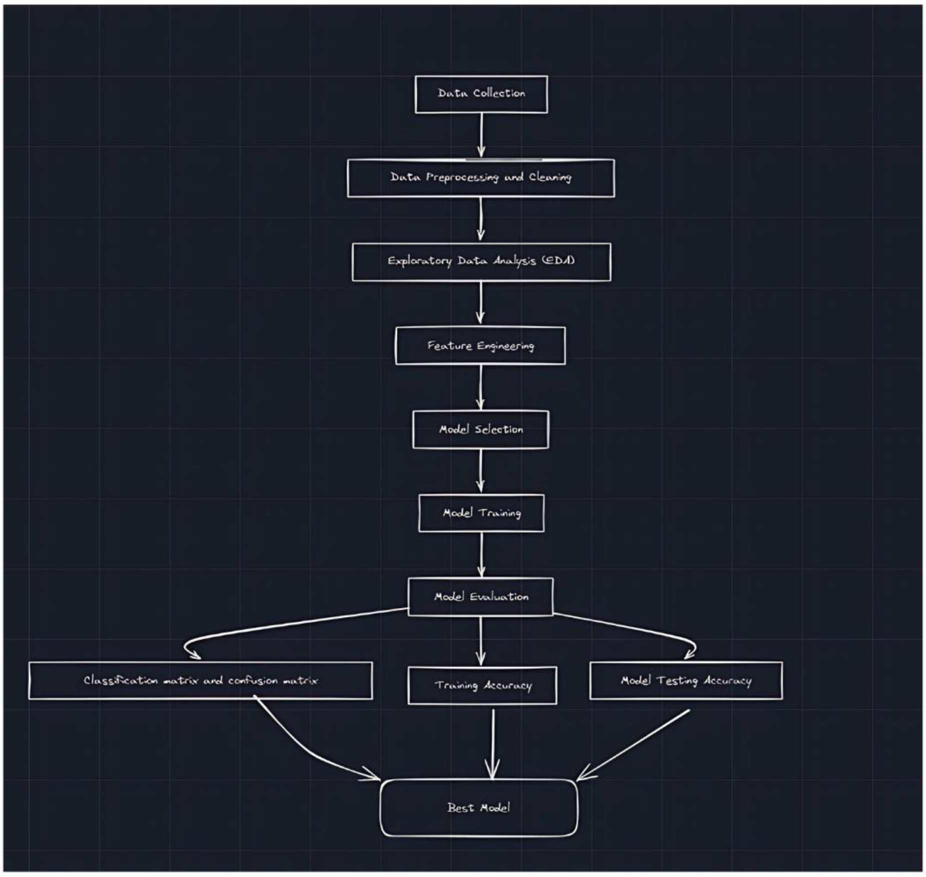
METHODOLOGY:

Import Required Libraries

```
import statistics
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import missingno as msno
import category_encoders as ce
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
```

```
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import warnings
warnings.filterwarnings("ignore")
```

FLOWCHART:



Flow Chart-1: Project Flow

Data Collection and EDA:

Dataset: [AI4I 2020 Predictive Maintenance Dataset - UCI Machine Learning Repository]

The AI4I 2020 Predictive Maintenance Dataset is a synthetic dataset that reflects real predictive maintenance data encountered in industry.

Dataset Characteristics Multivariate, Time-Series	**Subject Area** Computer Science
Associated Tasks Classification, Regression, Causal-Discovery	**Feature Type** Real
# Instances 10000	**# Features** 6

Since real predictive maintenance datasets are generally difficult to obtain and in particular difficult to publish, we present and provide a synthetic dataset that reflects real predictive maintenance encountered in industry to the best of our knowledge.

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

(1) UID: unique identifier ranging from 1 to 10000
(2) productID: consisting of a letter L, M, or H for low (50% of all products), medium (30%), and high (20%) as product quality variants and a variant-specific serial number
(3) air temperature [K]: generated using a random walk process later normalized to a standard deviation of 2 K around 300 K
(4) 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.
(5) rotational speed [rpm]: calculated from powepower of 2860 W, overlaid with a normally distributed noise
(6) torque [Nm]: torque values are normally distributed around 40 Nm with an Ïf = 10 Nm and no negative values.
(7) 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.

Important : There are two Targets - Do not make the mistake of using one of them as feature, as it will lead to leakage.

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

Acknowledgements
UCI: https://archive.ics.uci.edu/ml/datasets/AI4I+2020+Predictive+Maintenance+Dataset

Importing Data:

```
df=pd.read_csv("predictive_maintenance.csv")
df = df.drop(["UDI","Product ID"],axis=1)
df.sample(6).style.set_properties(
    **{
        'background-color': 'Brown',
        'color': 'white',
        'border-color': 'White'
    })
```

	Type	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]	Target	Failure Type
7542	M	300.400000	311.600000	1401	49.900000	79	0	No Failure
3570	H	301.800000	310.600000	1441	46.300000	100	0	No Failure
3412	L	301.300000	310.300000	1313	55.600000	127	0	No Failure
6367	L	300.000000	309.600000	1571	33.100000	65	0	No Failure
3679	L	302.200000	311.500000	1725	34.000000	170	0	No Failure
6327	L	300.600000	310.000000	1425	49.500000	174	0	No Failure

Analysing Numerical Columns:

```
[8]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 9 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Type                        10000 non-null  object
 1   Air temperature [°C]        10000 non-null  float64
 2   Process temperature [°C]    10000 non-null  float64
 3   Rotational speed [rpm]      10000 non-null  int64
 4   Torque [Nm]                 10000 non-null  float64
 5   Tool wear [min]             10000 non-null  int64
 6   Target                      10000 non-null  int64
 7   Failure Type                10000 non-null  object
 8   Temperature difference [°C] 10000 non-null  float64
dtypes: float64(4), int64(3), object(2)
memory usage: 703.3+ KB
```

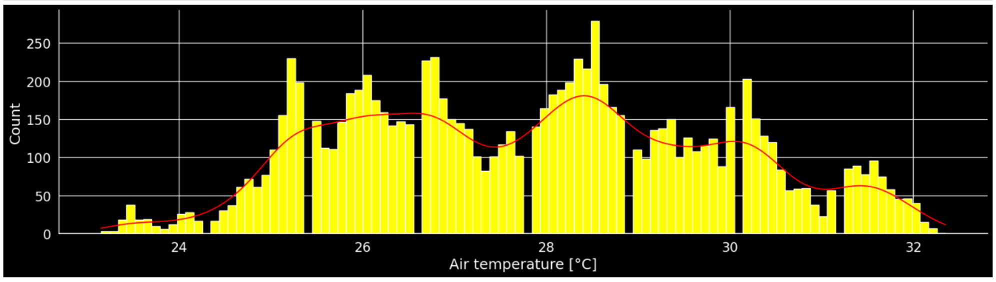
```
[9]: df.describe().style.background_gradient(cmap="magma")
```

[9]:

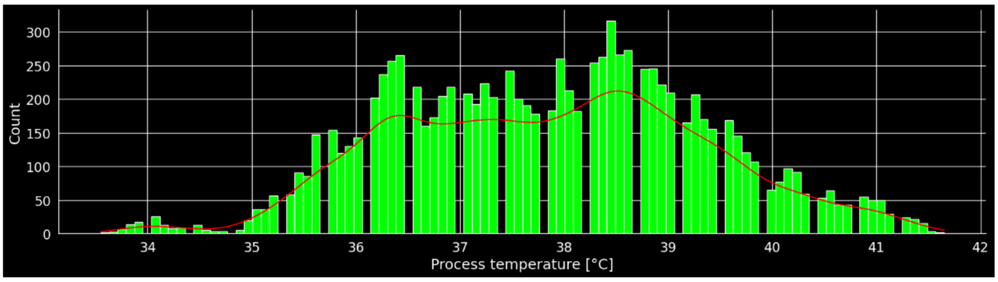
	Air temperature [°C]	Process temperature [°C]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]	Target	Temperature difference [°C]
count	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000
mean	27.854930	37.855560	1538.776100	39.986910	107.951000	0.033900	10.000630
std	2.000259	1.483734	179.284096	9.968934	63.654147	0.180981	1.001094
min	23.150000	33.550000	1168.000000	3.800000	0.000000	0.000000	7.600000
25%	26.150000	36.650000	1423.000000	33.200000	53.000000	0.000000	9.300000
50%	27.950000	37.950000	1503.000000	40.100000	108.000000	0.000000	9.800000
75%	29.350000	38.950000	1612.000000	46.800000	162.000000	0.000000	11.000000
max	32.350000	41.650000	2886.000000	76.600000	253.000000	1.000000	12.100000

Count Plots:

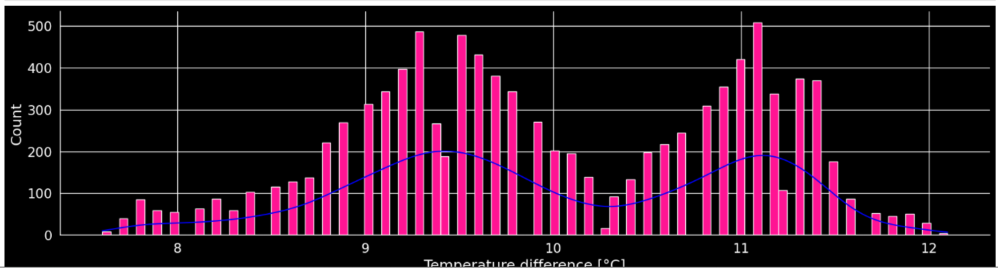
```
[13]: sns.displot(data=df, x="Air temperature [°C]", kde=True, bins = 100,color = "red", facecolor = "yellow",height = 5, aspect = 3.5)
plt.show()
```




```
[14]: sns.displot(data=df, x="Process temperature [°C]", kde=True, bins = 100,color = "red", facecolor = "lime",height = 5, aspect = 3.5)
plt.show()
```




```
[15]: sns.displot(data=df, x="Temperature difference [°C]", kde=True, bins = 100,color = "blue", facecolor = "DeepPink",height = 5, aspect = 3.5)
plt.show()
```



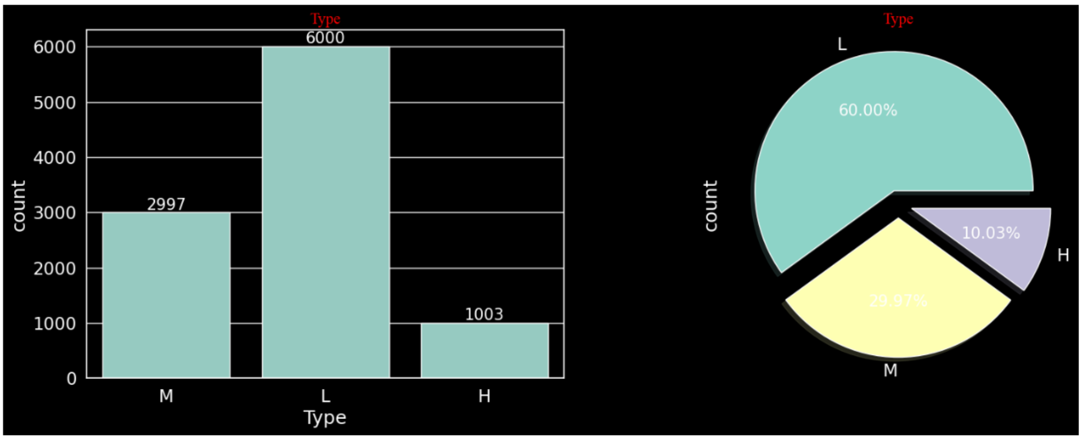
Value Counts of Categorical Variables:

```
[16]: for col in df[['Type','Target','Failure Type']]:
          print(df[col].value_counts())
          print("****"*8)
```

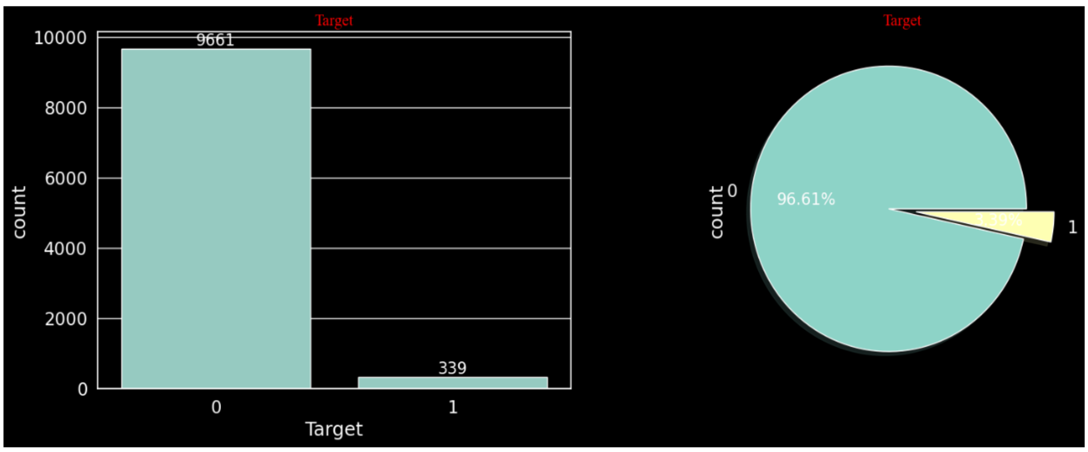
```
Type
L    6000
M    2997
H    1003
Name: count, dtype: int64
********************************
Target
0    9661
1     339
Name: count, dtype: int64
********************************
Failure Type
No Failure                  9652
Heat Dissipation Failure     112
Power Failure                 95
Overstrain Failure            78
Tool Wear Failure             45
Random Failures               18
Name: count, dtype: int64
********************************
```

Analyzing Categorical Columns:

```
[17]: ax = plt.figure(figsize=(18,6))
      ax = plt.subplot(1,2,1)
      ax = sns.countplot(x='Type', data=df)
      ax.bar_label(ax.containers[0])
      plt.title("Type", fontsize=20,color='Red',font='Times New Roman')
      ax =plt.subplot(1,2,2)
      ax=df['Type'].value_counts().plot.pie(explode=[0.1, 0.1,0.1],autopct='%1.2f%%',shadow=True)
      ax.set_title(label = "Type", fontsize = 20,color='Red',font='Times New Roman')
      plt.show()
```

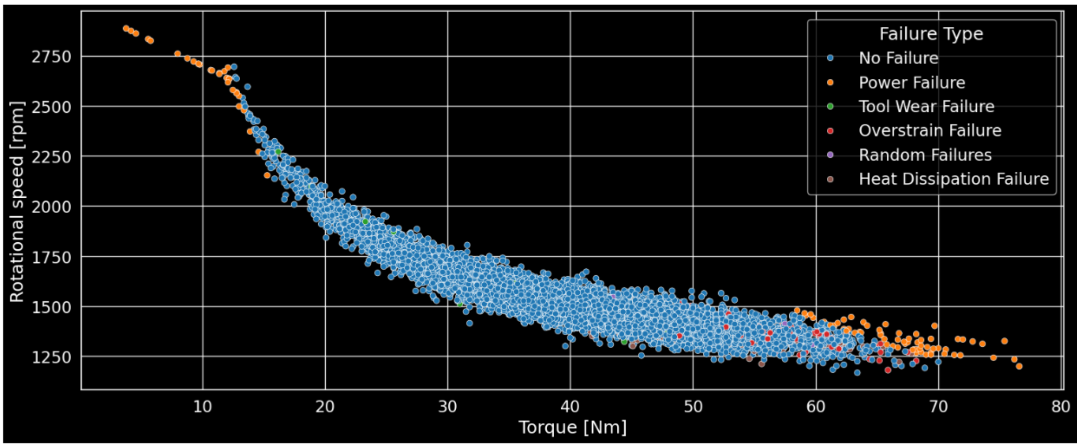


Count plot and pie chart depicting the number and percentage of ***Types*** of machines

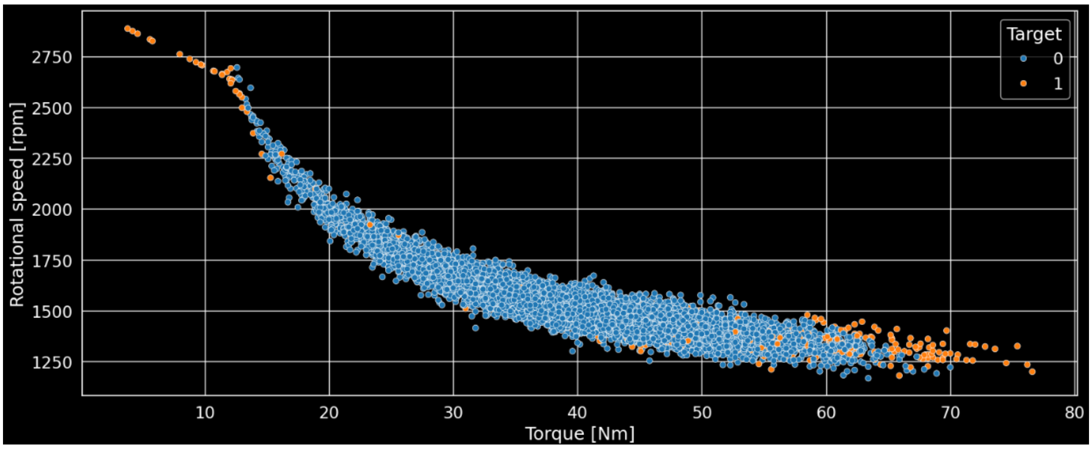


Count plot and pie chart depicting the number and percentage of ***Failures or Not*** of machines

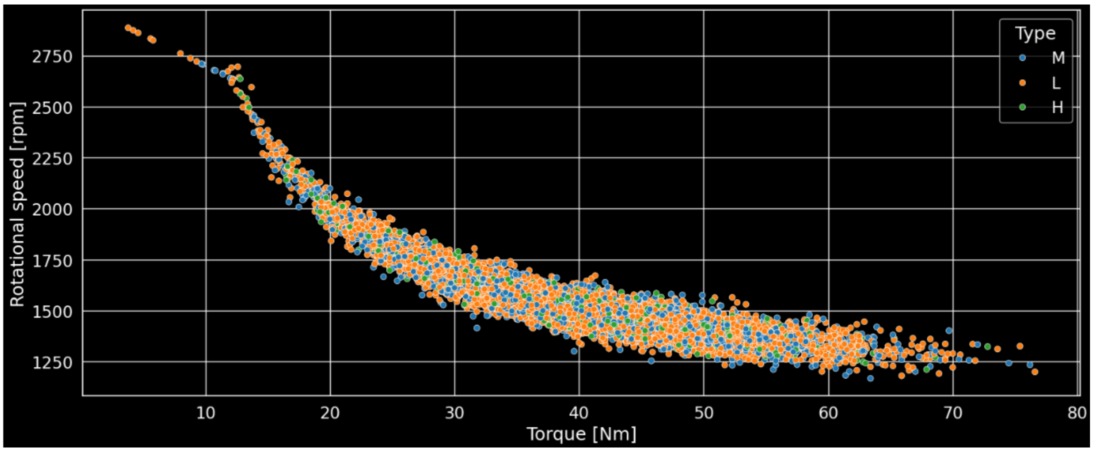
```
plt.figure(figsize=(18,7))
sns.scatterplot(data=df, x="Torque [Nm]", y="Rotational speed [rpm]", hue="Failure Type",palette="tab10")
plt.show()
```



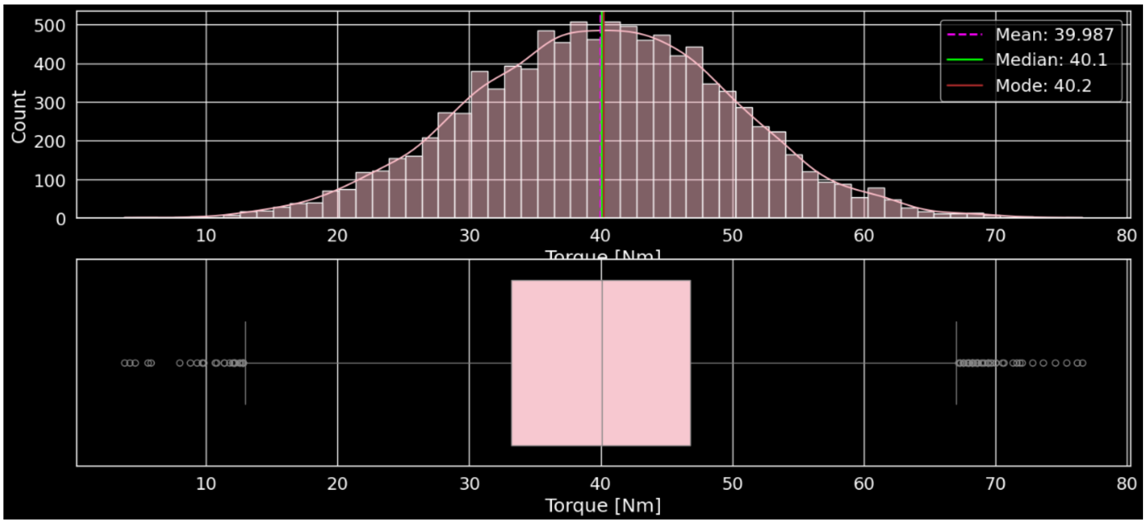
Scatter plot depicting the relation between ***Rotational Speed[rpm] and Torque [Nm]*** machines with hue of ***Failure Type***



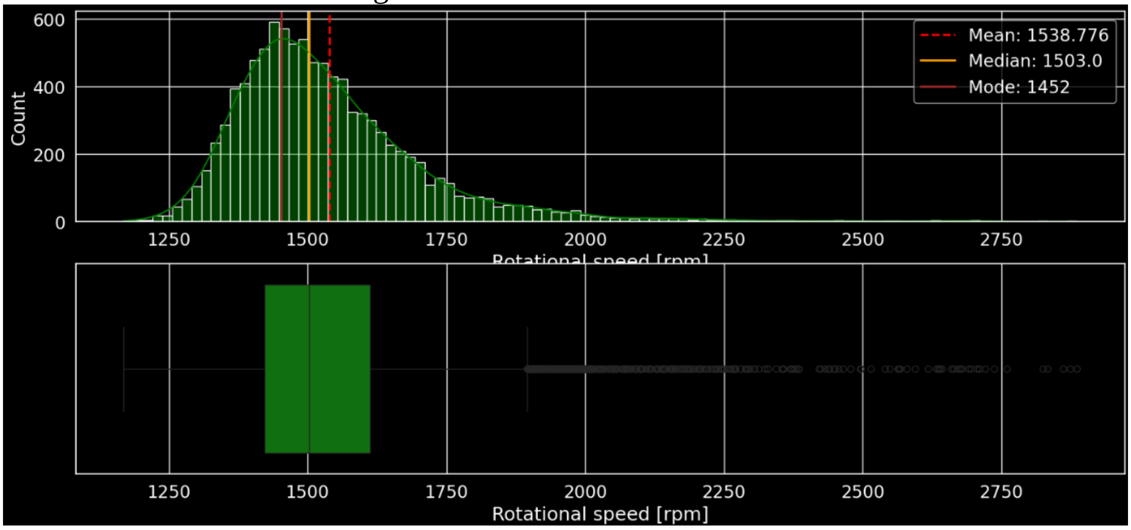
Scatter plot depicting the relation between ***Rotational Speed[rpm] and Torque [Nm]*** machines with hue of ***Failure or Not*** (0: Safe, 1: Failure)



Scatter plot depicting the relation between ***Rotational Speed[rpm] and Torque [Nm]*** machines with hue of ***Type of Machines***



Distribution plot and Box plot of ***Torque [Nm]*** which also ***depicts mean, median and mode*** for better understanding of the variable



Distribution plot and Box plot of ***Rotational Speed [rpm]*** which also ***depicts mean, median and mode*** for better understanding of the variable

Model Training:

After initial EDA, we will encode the remaining categorical variable and train the X,y for further development

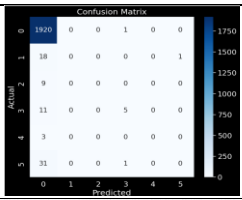
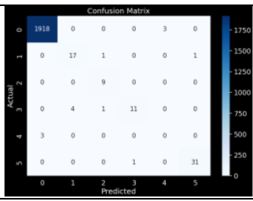
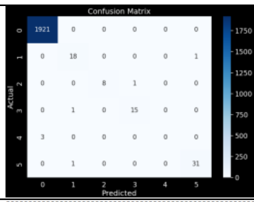
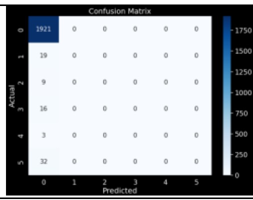
```
encoder = ce.OrdinalEncoder(cols=['Type','Failure Type'])
df = encoder.fit_transform(df)
df.head()
```

	Type	Air temperature [°C]	Process temperature [°C]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]	Target	Failure Type	Temperature difference [°C]
0	1	25.95	36.45	1551	42.8	0	0	1	10.5
1	2	26.05	36.55	1408	46.3	3	0	1	10.5
2	2	25.95	36.35	1498	49.4	5	0	1	10.4
3	2	26.05	36.45	1433	39.5	7	0	1	10.4
4	2	26.05	36.55	1408	40.0	9	0	1	10.5

```
scaler = LabelEncoder()
df['Failure Type'] = scaler.fit_transform(df['Failure Type'])
```

```
X = df.drop(columns="Failure Type" , axis=1)
y = df["Failure Type"]
```

```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=21)
```

Models	**Logistic Regression**	**Decision Tree**	**Random Forest Classifier**	**SVM**
Accuracy	Training Accuracy: **96.73 %** Model Accuracy Score: **96.25 %**	Training Accuracy: **100%** Model Accuracy Score: **99.3 %**	Training Accuracy: **100%** Model Accuracy Score: **99.65 %**	Training Accuracy: **96.45%** Model Accuracy Score: **96.05 %**
Confusion Matrix				

Classification Report

Logistic Regression:

```
Classification_Report:
              precision    recall  f1-score   support

           0       0.96      1.00      0.98      1921
           1       0.00      0.00      0.00        19
           2       0.00      0.00      0.00         9
           3       0.71      0.31      0.43        16
           4       0.00      0.00      0.00         3
           5       0.00      0.00      0.00        32

    accuracy                           0.96      2000
   macro avg       0.28      0.22      0.24      2000
weighted avg       0.93      0.96      0.95      2000
```

Decision Tree:

```
Classification_Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      1921
           1       0.81      0.89      0.85        19
           2       0.82      1.00      0.90         9
           3       0.92      0.69      0.79        16
           4       0.00      0.00      0.00         3
           5       0.97      0.97      0.97        32

    accuracy                           0.99      2000
   macro avg       0.75      0.76      0.75      2000
weighted avg       0.99      0.99      0.99      2000
```

Random Forest Classifier:

```
Classification_Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      1921
           1       0.90      0.95      0.92        19
           2       1.00      0.89      0.94         9
           3       0.94      0.94      0.94        16
           4       0.00      0.00      0.00         3
           5       0.97      0.97      0.97        32

    accuracy                           1.00      2000
   macro avg       0.80      0.79      0.79      2000
weighted avg       1.00      1.00      1.00      2000
```

SVM:

```
Classification_Report:
              precision    recall  f1-score   support

           0       0.96      1.00      0.98      1921
           1       0.00      0.00      0.00        19
           2       0.00      0.00      0.00         9
           3       0.00      0.00      0.00        16
           4       0.00      0.00      0.00         3
           5       0.00      0.00      0.00        32

    accuracy                           0.96      2000
   macro avg       0.16      0.17      0.16      2000
weighted avg       0.92      0.96      0.94      2000
```

Sample Predictions

Logistic Regression	Actual	Predicted	Decision Tree	Actual	Predicted	Random Forest Classifier	Actual	Predicted	SVM	Actual	Predicted
2047	0	0	**684**	0	0	**6710**	0	0	**6842**	0	0
1322	0	0	**3083**	0	0	**4077**	0	0	**2819**	0	0
6856	0	0	**5588**	0	0	**5125**	0	0	**4080**	5	0
6784	0	0	**3154**	0	0	**4580**	0	0	**3651**	0	0
1104	0	0	**2681**	0	0	**3897**	0	0	**9264**	0	0
1283	0	0	**7836**	0	0	**3078**	0	0	**8552**	0	0
7643	0	0	**606**	0	0	**2749**	0	0	**6048**	0	0
6472	0	0	**354**	0	0	**396**	0	0	**2115**	0	0
5272	0	0	**3583**	0	0	**8083**	0	0	**6843**	0	0
2672	0	0	**23**	0	0	**8028**	0	0	**8562**	0	0

```
# Set precision for display
pd.set_option('display.precision', 2)

# Set the index to Model to hide the default index
models.set_index('Model', inplace=True)

# Sort and style the DataFrame
styled_models = models.sort_values(by='Model Accuracy Score', ascending=False).style.background_gradient(
    cmap='coolwarm').set_properties(**{
    'font-family': 'Arial',
    'color': 'LightGreen',
    'font-size': '15px'
})

# Display styled DataFrame
styled_models
```

Model	**Training Accuracy**	**Model Accuracy Score**
Decision Tree	100.000000	99.650000
Random Forest	100.000000	99.300000
Support Vector Machines	96.730000	96.250000
Logistic Regression	96.640000	96.050000

RESULTS:

Model Performance Overview
Four machine learning models were trained and evaluated to predict machine failures. The performance of each model, in terms of training and test accuracy, is as follows:

Model	Training Accuracy	Test Accuracy
Logistic Regression	96.73%	96.25%
Decision Tree	100%	99.30%
Random Forest Classifier	100%	99.65%
Support Vector Machine (SVM)	96.45%	96.05%

1. **Logistic Regression**: Performed well with 96.25% accuracy, balancing interpretability and robustness.
2. **Decision Tree**: Achieved high accuracy but may risk overfitting due to its 100% training accuracy.
3. **Random Forest**: Outperformed other models with 99.65% accuracy, showing strong predictive capabilities on the test set due to ensemble benefits.
4. **SVM**: Delivered a similar performance to Logistic Regression, with a slight trade-off in interpretability.

The **outcomes** of a Predictive Maintenance Classification project, are highly valuable for industrial operations. Here are the key outcomes:

1. Machine Failure Prediction

- **Outcome**: A machine learning model capable of predicting whether a machine will fail at a given point in time based on operational data and sensor readings.
- **Impact**: Early detection of potential failures allows for maintenance to be scheduled before breakdowns occur, minimizing downtime and costly repairs.

2. Failure Type Classification

- **Outcome**: The model not only predicts whether a failure will occur but also identifies the **type of failure**, enabling precise diagnosis.
- **Impact**: This helps in planning the necessary repairs, allowing the maintenance team to address the specific issue more efficiently.

3. Reduced Downtime

- **Outcome**: By predicting failures in advance, businesses can schedule maintenance at optimal times, avoiding unexpected machine breakdowns.
- **Impact**: Operational disruptions are minimized, improving overall productivity and efficiency.

4. Optimized Maintenance Schedules

- **Outcome**: Maintenance can be performed based on actual equipment conditions rather than fixed schedules (preventive maintenance).
- **Impact**: This reduces the frequency of unnecessary maintenance activities and lowers associated costs, while still preventing failures.

5. Cost Savings

- **Outcome**: The reduction in emergency repairs, unexpected downtime, and unnecessary maintenance leads to significant cost savings.
- **Impact**: Companies save money on repairs, spare parts, and the labor costs associated with unplanned maintenance.

6. Improved Equipment Lifespan

- **Outcome**: Predictive maintenance helps ensure that machines are maintained in optimal condition, preventing excessive wear and tear.
- **Impact**: This extends the lifespan of equipment, delaying the need for costly replacements.

7. Data-Driven Insights

- **Outcome**: Through the analysis of features such as temperature, rotational speed, torque, and tool wear, the project provides valuable insights into how these parameters impact machine performance and failure rates.
- **Impact**: These insights can be used to further optimize machine operations and improve future designs or operational guidelines.

8. Increased Safety

- **Outcome**: By predicting failures in advance, the risk of catastrophic machine breakdowns that could pose safety hazards is reduced.
- **Impact**: This leads to a safer working environment for employees and less risk of equipment-related accidents.

9. Actionable Insights for Stakeholders

- **Outcome**: Reports or dashboards showing predictive maintenance metrics, failure probabilities, and key operational insights can be created for decision-makers.
- **Impact**: Stakeholders can make informed decisions on maintenance planning, resource allocation, and operational adjustments based on real-time data.

10. Improved Operational Efficiency

- **Outcome**: By preventing failures and reducing unnecessary maintenance, overall operational efficiency improves.
- **Impact**: Companies experience smoother operations, higher machine availability, and better resource management.

These outcomes collectively will help to significant improvements in both the operational and financial performance of a company, making predictive maintenance a critical component of modern industrial practices.

SUMMARY:

This project focused on the development of machine learning models aimed at predictive maintenance, designed to forecast machine failures before they occur. By analyzing historical operational data, we implemented four models: Logistic Regression, Decision Tree, Random Forest, and Support Vector Machine (SVM).

The key outcomes of this initiative include the ability to predict potential machine failures and classify specific failure types, allowing for proactive maintenance strategies. This capability minimizes unexpected downtimes and costly repairs while optimizing maintenance schedules based on real-time data, ultimately leading to significant cost savings. Furthermore, regular condition-based maintenance enhances equipment lifespan and fosters a safer work environment by mitigating the risk of catastrophic failures.

Through data-driven insights and real-time dashboards, stakeholders can make informed decisions that improve resource allocation and operational planning. Overall, this project not only enhances operational efficiency but also ensures smoother operations and better management of resources, driving long-term benefits for the organization.

CONCLUSION:

The predictive maintenance project successfully demonstrated the efficacy of machine learning models in anticipating machine failures and optimizing maintenance processes. By leveraging historical operational data, we achieved significant advancements in both predictive accuracy and operational efficiency. The ability to predict potential failures and classify their types not only reduced downtime but also minimized maintenance costs and extended the lifespan of equipment.
Moreover, the implementation of data-driven insights and real-time monitoring tools empowered stakeholders to make informed decisions, enhancing resource allocation and operational planning. This proactive approach contributes to a safer work environment by reducing the risk of catastrophic failures.
In summary, the project highlights the transformative potential of machine learning in predictive maintenance, paving the way for improved productivity, cost savings, and overall organizational resilience. Future work could further refine these models and explore additional data sources to enhance predictive capabilities and drive even greater value.

BIBLIOGRAPHY:

1. **Books:**
   - **Piotrowski, J.** (2024). *Predictive Maintenance: Optimizing Equipment Life-Cycle Decisions*
   - **Romero, V., & Martínez, J.** (2023). *Predictive Maintenance with Machine Learning.*
   - **Starner, T. K.** (2020). *Industrial Machine Learning: Predictive Maintenance*.

2. **Research Papers:**
   - "Artificial Intelligence for Predictive Maintenance Applications: Key Components, Trustworthiness, and Future Trends" by Aysegul Ucar, Mehmet Karakose, and Necim Kırımcı (2024).

   - "Explainable Predictive Maintenance: A Survey of Current Methods, Challenges and Opportunities" by Logan Cummins et al. (2024)2. This survey discusses current methods of explainable AI (XAI) as applied to predictive maintenance, along with challenges and future research directions.

3. **Online Resources:**
   - UC Irvine Machine Learning Repository https://archive.ics.uci.edu/dataset/601/ai4i+2020+predictive+maintenance+dataset

4. **Web Articles:**
   - "Artificial Intelligence for Predictive Maintenance Applications: Key Components, Trustworthiness, and Future Trends"1. This article reviews recent developments in AI
     based predictive maintenance, focusing on key components, trustworthiness, and future trends
   - "Predictive Maintenance in IoT-
     Monitored Systems for Fault Prevention". This paper focuses on predictive maintenance for simple machinery systems monitored by the Internet of Things (IoT), proposing a novel clustering-based process to identify potential machinery faults

5. **Thesis:**
   - S. Matzka (2020). Explainable Artificial Intelligence for Predictive Maintenance Applications. DOI:10.1109/AI4I49448.2020.00023

6. **Datasets:**
   - AI4I 2020 Predictive Maintenance Dataset
     Donated on 8/29/2020
     https://archive.ics.uci.edu/dataset/601/ai4i+2020+predictive+maintenance+dataset.

7. **Python Notebook:**
   https://drive.google.com/drive/folders/1DU0DDQLojkH1XDAWVB0w80L2PMcG2KZI?usp=sharing