Fuse Machines

AI Fellowship 2023

A Final Report

on

FAILURE CLASSIFICATION

Submitted by

Samundra Karki

Submitted to

Fuse Machine

Submission date

June 2 2023

Table of Content

CHAPTER 1: ABSTRACT **1**
CHAPTER 2: Introduction **1**
2.1 Problem Statement 1
2.2 About Dataset 1
2.3 Literature Review 2
CHAPTER 3: FEATURE ENGINEERING AND DATA ANALYSIS **4**
3.1 Data Statistics 4
3.2 Correlations between Features 4
3.3 Data Distribution and Box plot 7
3.4 Min Max Scaling 9
3.5 Boxcox transformation 12
CHAPTER 4: MODEL AND ANALYSIS **15**
4.1 Logistic Regression 15
4.2 RANDOM FOREST 17
4.3 XGB Boosting 19
CHAPTER 5: CONCLUSION **21**

List of Figures

Figure 2.1 information of dataset 1
Figure 2.2 Count of data at target=0 and target=1 2
Figure 3.1 Correlation heat map 5
Figure 3.2 Distribution of Torque and Failures 5
Figure 3.3 Distribution of Tool wear(min) and Failures 6
Figure 3.4 Air vs Process Temperature 6
Figure 3.5 Rotational speed vs Torque 7
Figure 3.6 Original Data Distribution 7
Figure 3.7 Box Plot of original data 8
Figure 3.8 Distribution of Feature in regard to failure 9
Figure 3.9 Distribution of minmax scaling 9
Figure 3.10 Boxplot of minmax scaling 10
Figure 3.11 Distribution of Log of Minmax 10

Figure 3.12 Box plot of log of minmax 11
Figure 3.13 Distribution of BoxCox transformation..... 12
Figure 3.14 Boxplot of BoxCox transformation 13
Figure 3.15 Pair Plot of every features 14
Figure 4.1 Comparison of Different Data Distribution of Logistic Regression 16
Figure 4.2 Comparison of Logistic Regression with SMOTE Sampling 16
Figure 4.3 Comparison Log-Min-max and Min-max Scaling with Random Forest..... 17
Figure 4.4 Comparison of three Machine Learning Algorithms..... 17
Figure 4.5 Performance with Best hyper-parameters 18
Figure 4.6 Classification Report and Confusion Matrix with XGB boosting..... 19
Figure 4.7 Performance for Hyper-parameter with XGB boosting 20

List of Table

Table 2.1 Sample Data..... 1
Table 3.1 Data Statistics 4
Table 4.1 SMOTE Sampling..... 16
Table 4.2 Hyper-parameter for Random Forest..... 18
Table 4.3 Hyper Parameters for XGB Boosting 19

CHAPTER 1: ABSTRACT

The project use data from Predictive Maintenance to predict Failure of Machines. Logistic Regression is first used to compare different data transformation methods. SMOTE method was used to handle the imbalanced data which increased the recall up to 0.75. Random Forest method was used to compare the recall and accuracy for two data transformation for Min-max scaling and log min-max scaling , which showed pretty similar performance in accuracy and recall. Comparison of SVC, Random Forest and KNN showed good accuracy with Random Forest. Random Forest was tuned with hyper-parameters which improved the recall to 0.84 and accuracy 0.96. XGB boosting doesn't much improve accuracy and recall.

Keywords: Predictive Maintenance, Random Forest, Machine Learning

CHAPTER 2: Introduction

2.1 Problem Statement

Machine Failure is a critical problem in manufacturing industries. A variety of maintenance mechanism exists like Routine Maintenance, Corrective Maintenance, Preventive Maintenance, etc. This maintenance carries a considerable burden of managing resources for maintenance without being sure if the maintenance is actually required or not. In contrast to these methods, Predictive Maintenance predicts a probable failure of a machine and performs maintenance of such a machine reducing overhead costs of unnecessary maintenance. The key requirement of Predictive Maintenance is the prediction of the failure of the Machine. In this project, we take a 10000 dataset with a variety of operating features and apply Machine Learning Algorithms to predict the failure.

The source of dataset is:

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

2.2 About Dataset

It contains 10000 data sets with the following features

```
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   UDI                     10000 non-null  int64
 1   Product ID              10000 non-null  object
 2   Type                    10000 non-null  object
 3   Air temperature [K]     10000 non-null  float64
 4   Process temperature [K] 10000 non-null  float64
 5   Rotational speed [rpm]  10000 non-null  int64
 6   Torque [Nm]             10000 non-null  float64
 7   Tool wear [min]         10000 non-null  int64
 8   Target                  10000 non-null  int64
 9   Failure Type            10000 non-null  object
```

Figure 2.1 information of dataset

A peek at the data looks like the following

Table 2.1 Sample Data

Index	UDI	Product ID	Type	Air temperature [K]	Process temperature [K]	Rotational speed	Torque [Nm]	Tool wear [min]	Target

						[rpm]			
0	1	M14860	M	298.1	308.6	1551	42.8	0	0
1	2	L47181	L	298.2	308.7	1408	46.3	3	0
2	3	L47182	L	298.1	308.5	1498	49.4	5	0
3	4	L47183	L	298.2	308.6	1433	39.5	7	0
4	5	L47184	L	298.2	308.7	1408	40	9	0

Target represents whether the machine fails or not and as it is the subject of our classification problem the data looks very imbalanced.

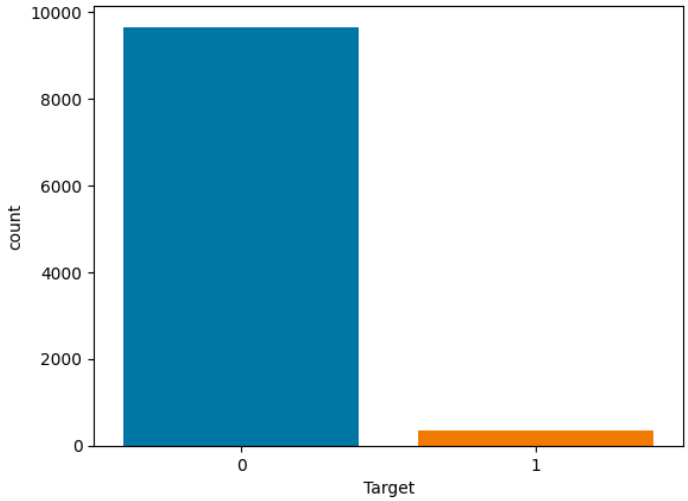


Figure 2.2 Count of data at target=0 and target=1

Only 3.39% of the data has Target = 1. As it is between 1-20% it is called Moderate imbalance (https://developers.google.com/machine-learning/data-prep/construct/sampling-splitting/imbalanced-data). So, the target needs to be stratified in order to preserve the distribution of data while training.

Keeping that in mind 80%-20% split of data is performed for training and testing.

2.3 Literature Review

Preventative Maintenance (PvM), also known as time-based or scheduled maintenance, is a widely adopted technique used to prevent equipment failures by performing maintenance actions periodically according to a planned schedule. While PvM is generally effective in avoiding failures, it often leads to unnecessary corrective actions, resulting in increased operating costs.

In contrast, Predictive Maintenance (PdM) is a proactive maintenance approach that utilizes predictive tools to determine the optimal timing for maintenance actions. PdM relies on continuous monitoring of machine or process integrity, enabling maintenance activities to be performed only when necessary. By leveraging historical data, integrity factors, statistical inference methods, and engineering approaches, PdM enables early detection of failures.
The proliferation of sensing technologies has significantly increased the amount of data extracted from production processes. This data, when processed and analyzed, can provide valuable insights and knowledge about manufacturing processes, production systems, and equipment. In industries, equipment maintenance plays a crucial role in operational efficiency and uptime. Therefore, it is essential to identify and address equipment faults promptly to avoid disruptions in production processes.

Machine Learning (ML) methods have emerged as a promising tool in Predictive Maintenance (PdM) applications, specifically for preventing failures in production lines on the factory floor. Researchers such as A. Kane, A. Kore, A. Khandale, S. Nigade, and P. P. Joshi have explored the use of ML techniques, including Random Forest and LSTM, for predictive maintenance tasks. Their study titled "Predictive Maintenance using Machine Learning" demonstrates the effectiveness of these algorithms in failure prediction.
In another study titled "Predict Failures in Production Lines: A Two-Stage Approach with Clustering and Supervised Learning" (author(s) not specified), a two-step method combining clustering and supervised learning is proposed for failure prediction. The research highlights the preference for Random Forest as a preferred machine learning method in their prediction framework.
These studies exemplify the utilization of machine learning algorithms, particularly Random Forest and LSTM, for predictive maintenance purposes. The research underscores the potential of machine learning techniques in accurately predicting failures and aiding in the optimization of maintenance strategies. The application of such approaches can lead to reduced downtime, increased operational efficiency, and cost savings for industries relying on production line equipment.

CHAPTER 3: FEATURE ENGINEERING AND DATA ANALYSIS

The Product ID and UID doesn't provide significant information in the failure of a machine. These data are dropped. Similarly, "Type" was changed from category to numerical value with the following mapping: 'L': 0, 'M': 1, 'H': 2

3.1 Data Statistics

Table 3.1 Data Statistics

index	Type	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]	Target
count	10000	10000	10000	10000	10000	10000	10000
mean	0.5003	300.00493	310.00556	1538.7761	39.98691	107.951	0.0339
std	0.6713009567	2.000258683	1.483734219	179.2840959	9.968933725	63.65414664	0.1809808427
min	0	295.3	305.7	1168	3.8	0	0
25%	0	298.3	308.8	1423	33.2	53	0
50%	0	300.1	310.1	1503	40.1	108	0
75%	1	301.5	311.1	1612	46.8	162	0
max	2	304.5	313.8	2886	76.6	253	1

3.2 Correlations between Features

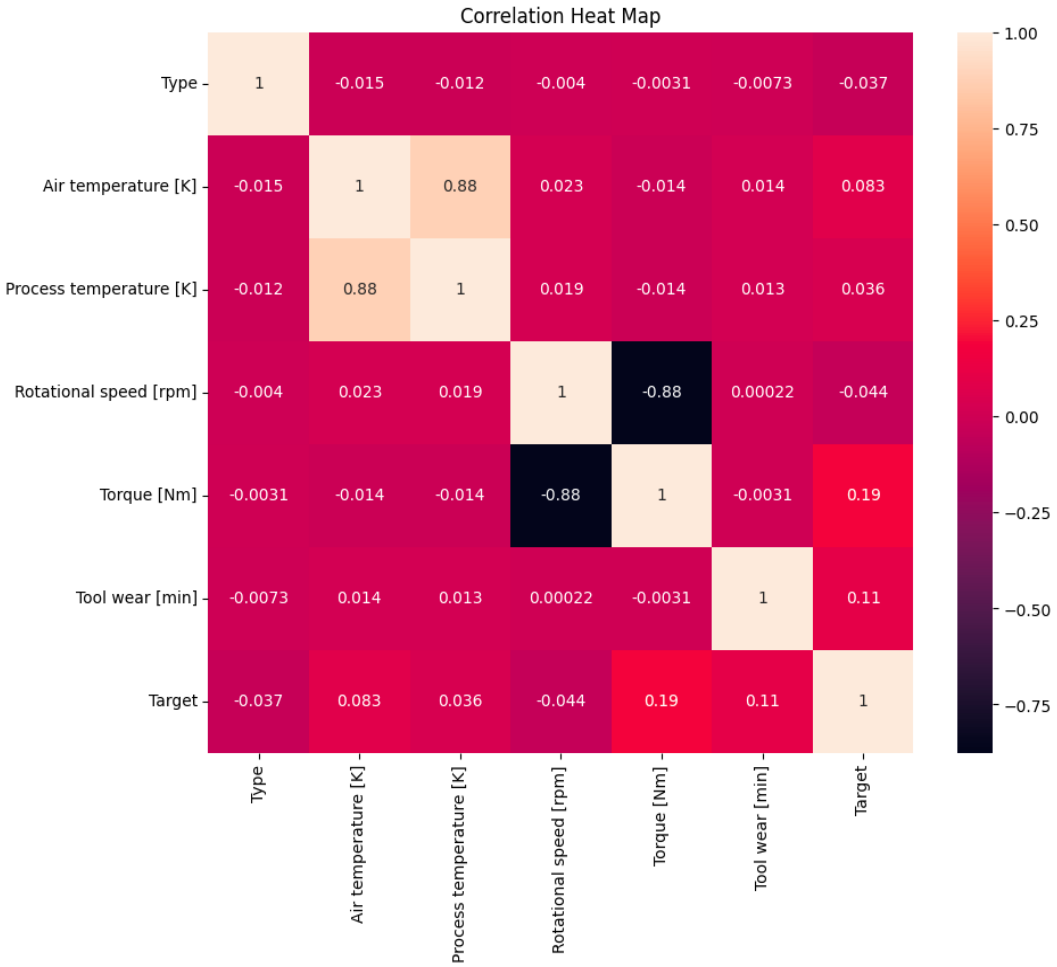


Figure 3.1 Correlation heat map

The heatmap shows that the Torque and Tool wear have more correlation with the Target. So, the distribution between Failures (Target=1) and Torque and Tool wear are observed:

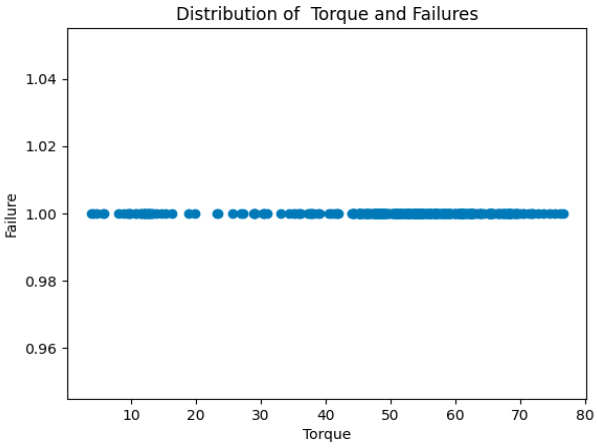


Figure 3.2 Distribution of Torque and Failures

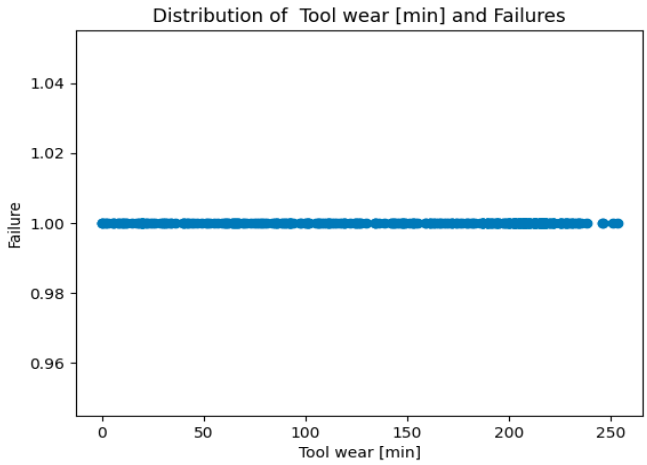


Figure 3.3 Distribution of Tool wear(min) and Failures

The distribution shows higher counts of Failure as the Torque progressively increases. Such relation is not much more prominent in the case of Tool Wear

Air Temperature shows a 88% correlation with the process temperature. An intuitive reason might be the rise in Ambient Temperature contributed to the rise in Process Temperature as well.

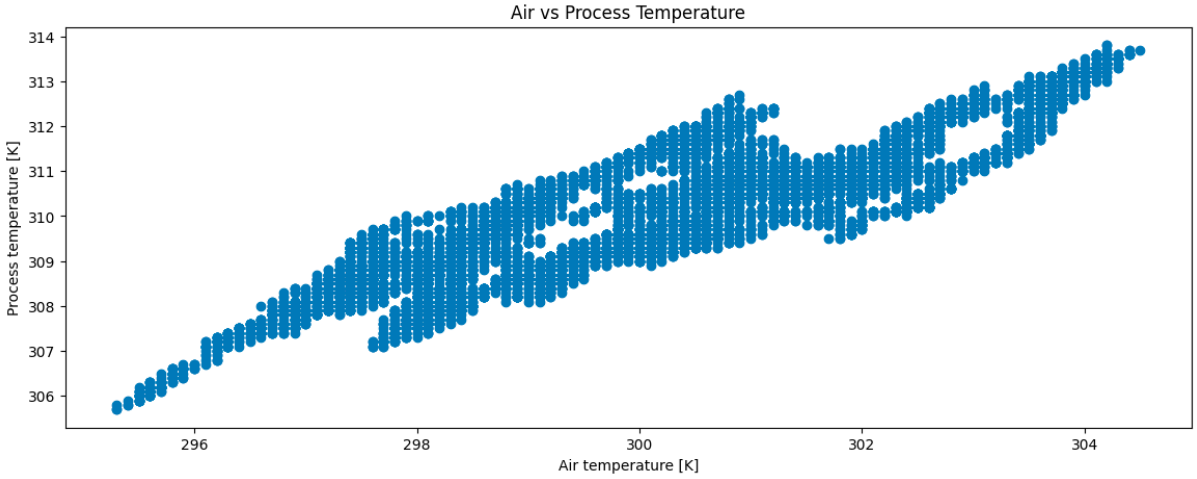


Figure 3.4 Air vs Process Temperature

Hight Torque requires a lower Rotational Speed which is very evident in the graph below as well as the correlation heatmap.

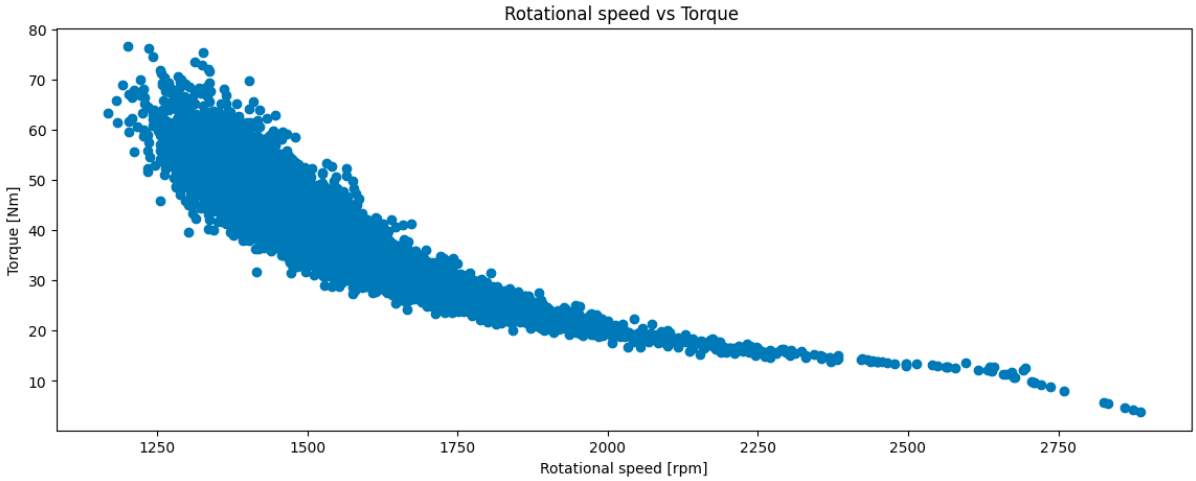


Figure 3.5 Rotational speed vs Torque

3.3 Data Distribution and Box plot

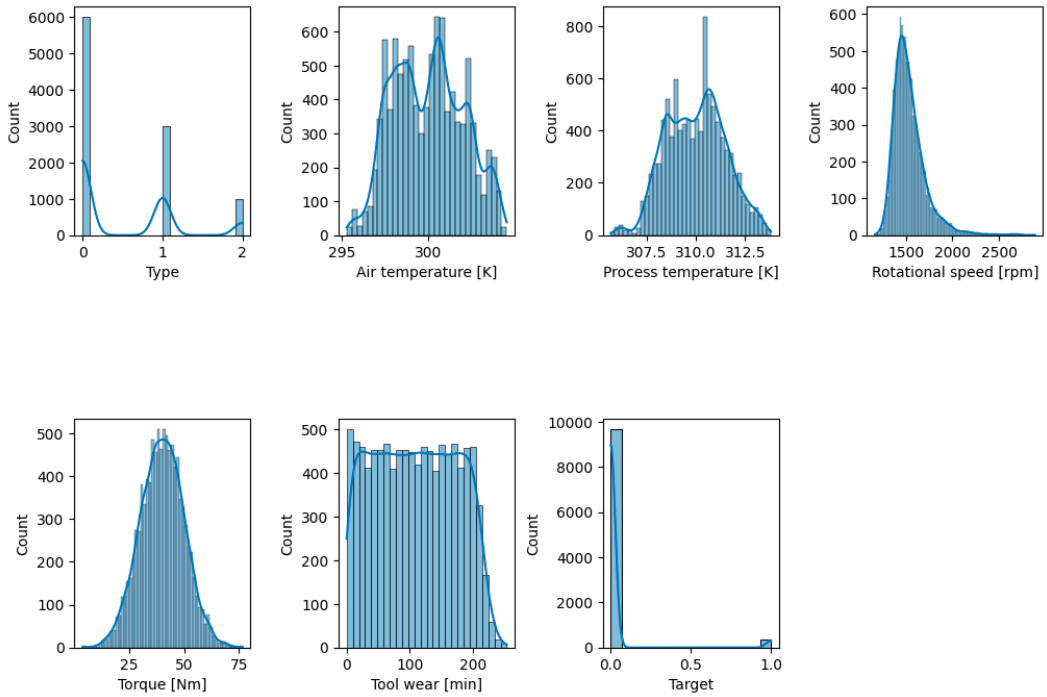


Figure 3.6 Original Data Distribution

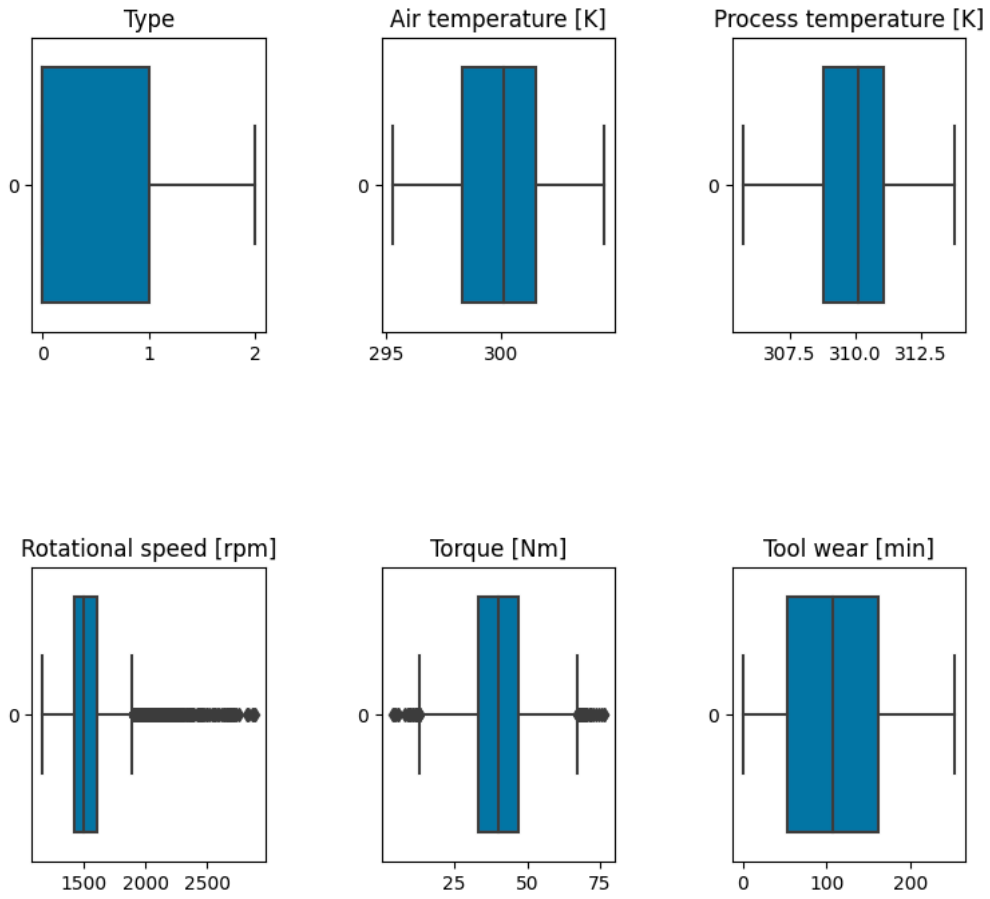


Figure 3.7 Box Plot of original data

As can be seen from distributions Torque looks normally distributed but other features aren't. The distribution of features in regards to failure is important to be inquired as well.

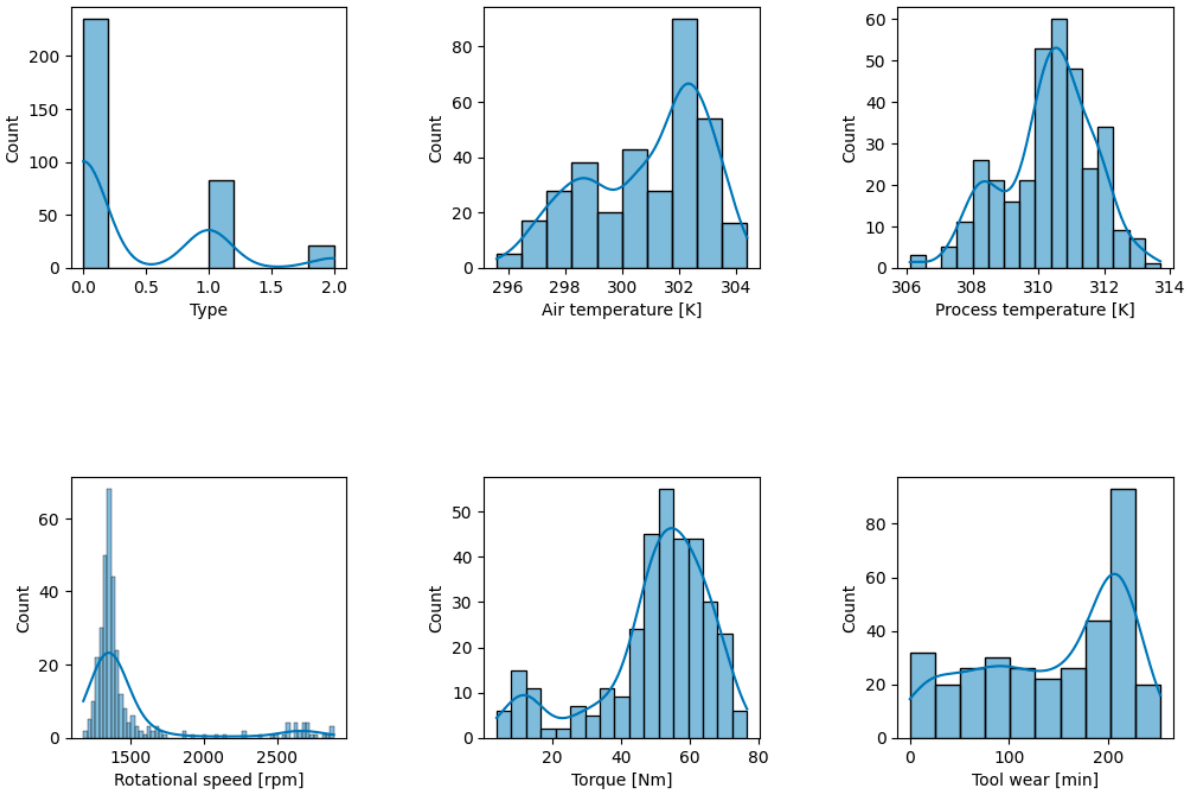


Figure 3.8 Distribution of Feature in regard to failure

3.4 Min Max Scaling

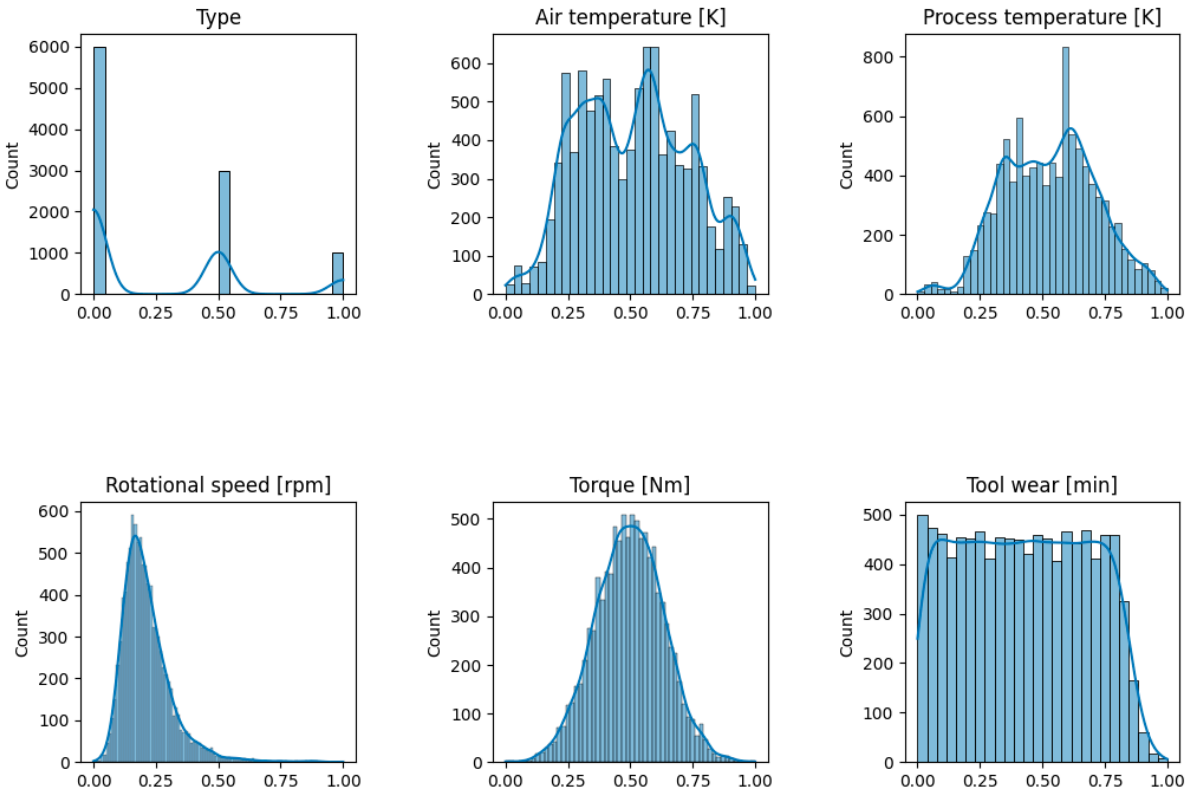


Figure 3.9 Distribution of minmax scaling

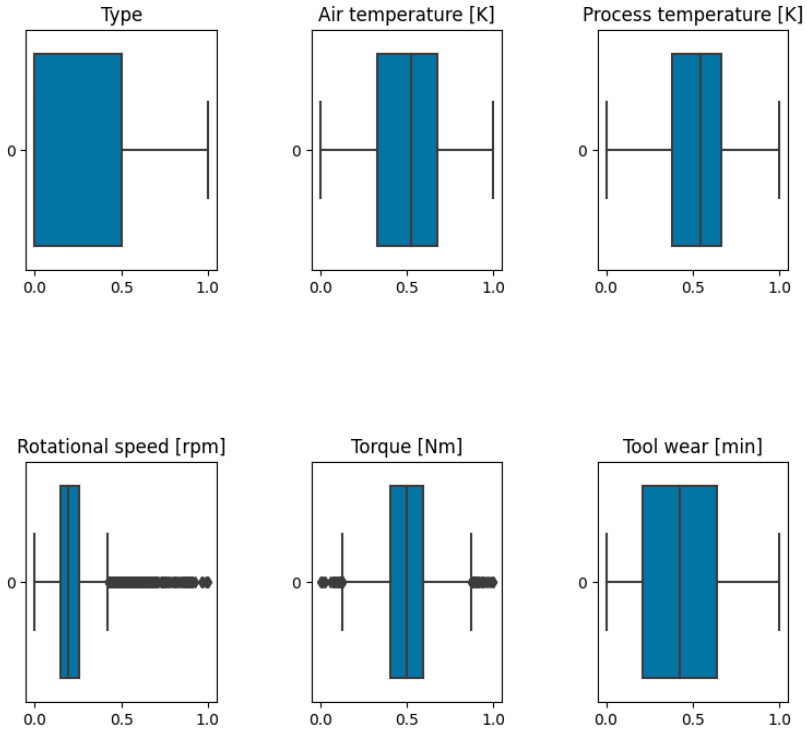


Figure 3.10 Boxplot of minmax scaling

LOGARITHM OF MIN MAX

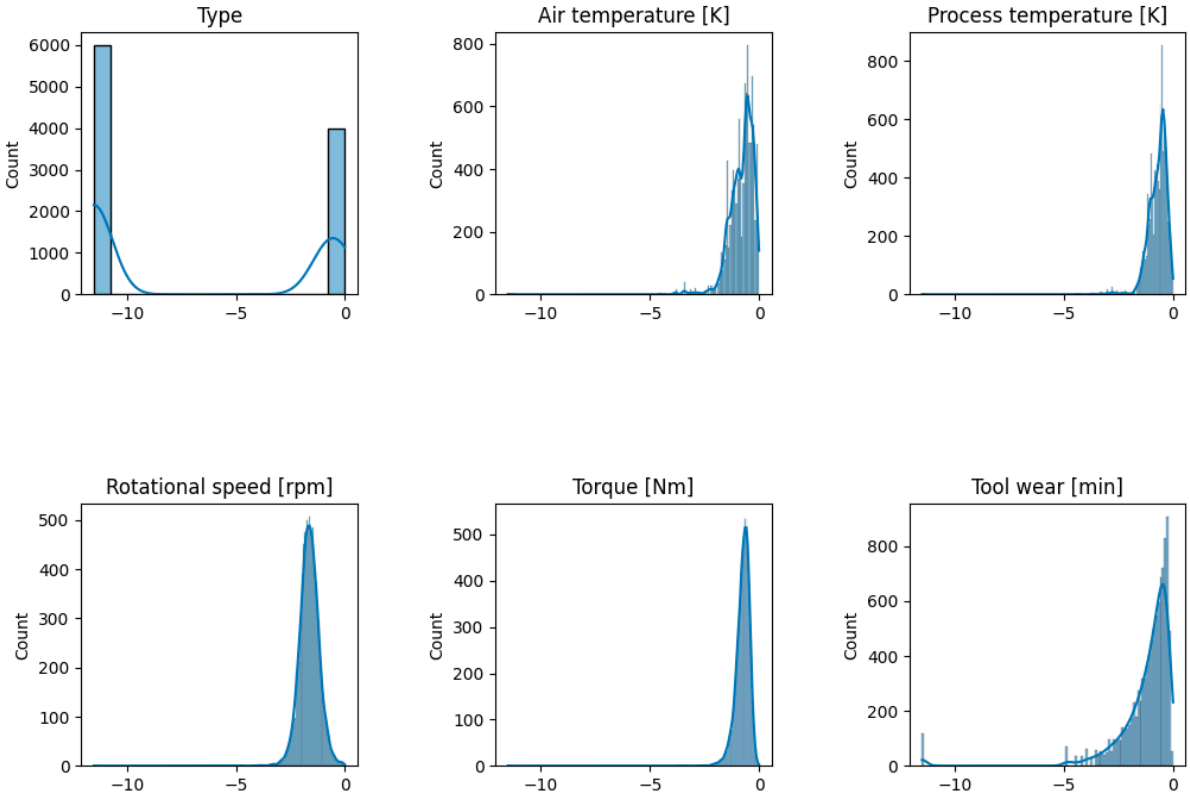


Figure 3.11 Distribution of Log of Minmax

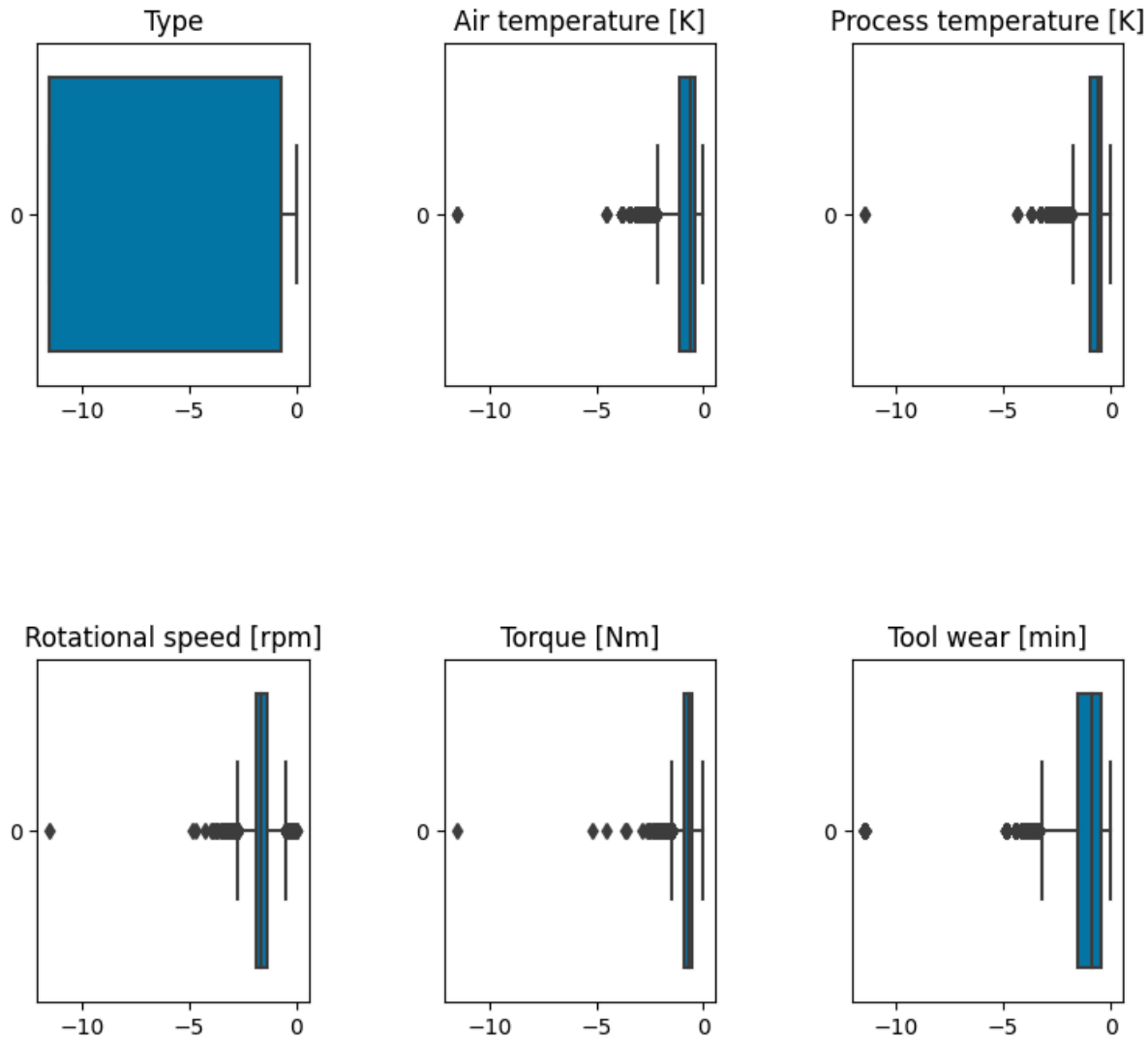


Figure 3.12 Box plot of log of minmax

3.5 Boxcox transformation

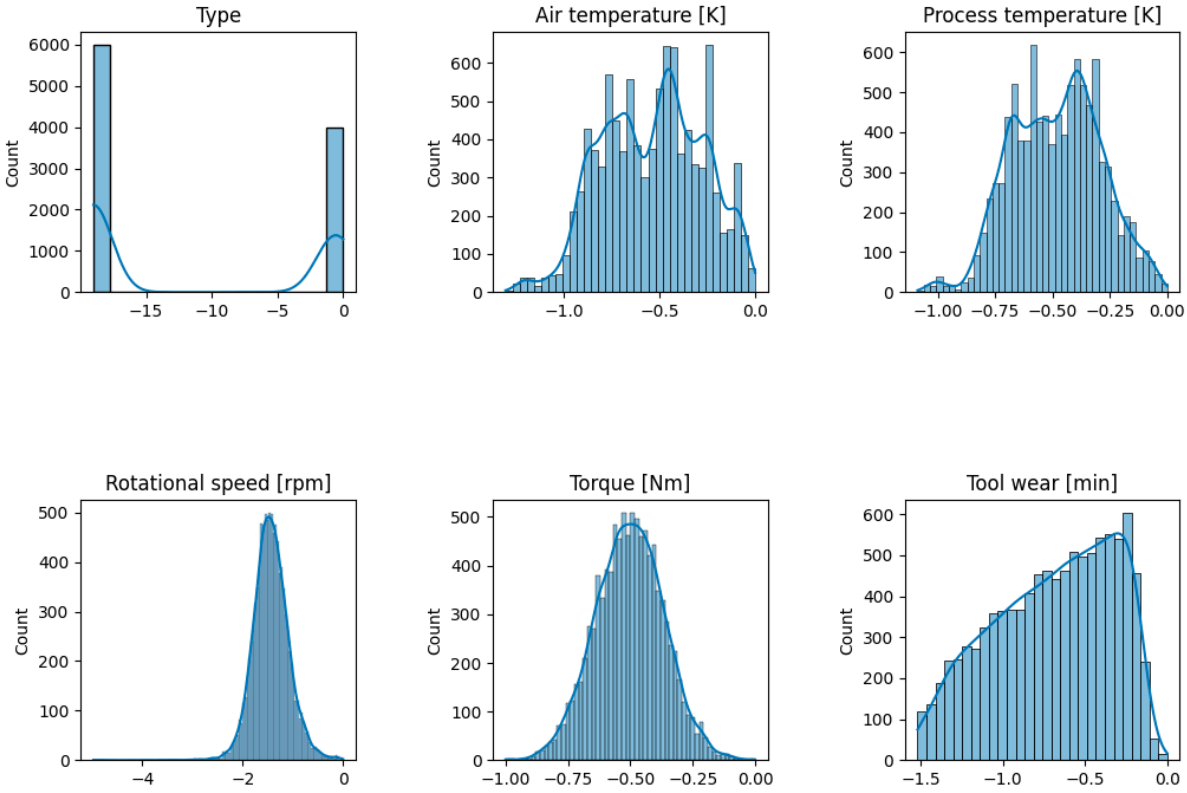


Figure 3.13 Distribution of BoxCox transformation

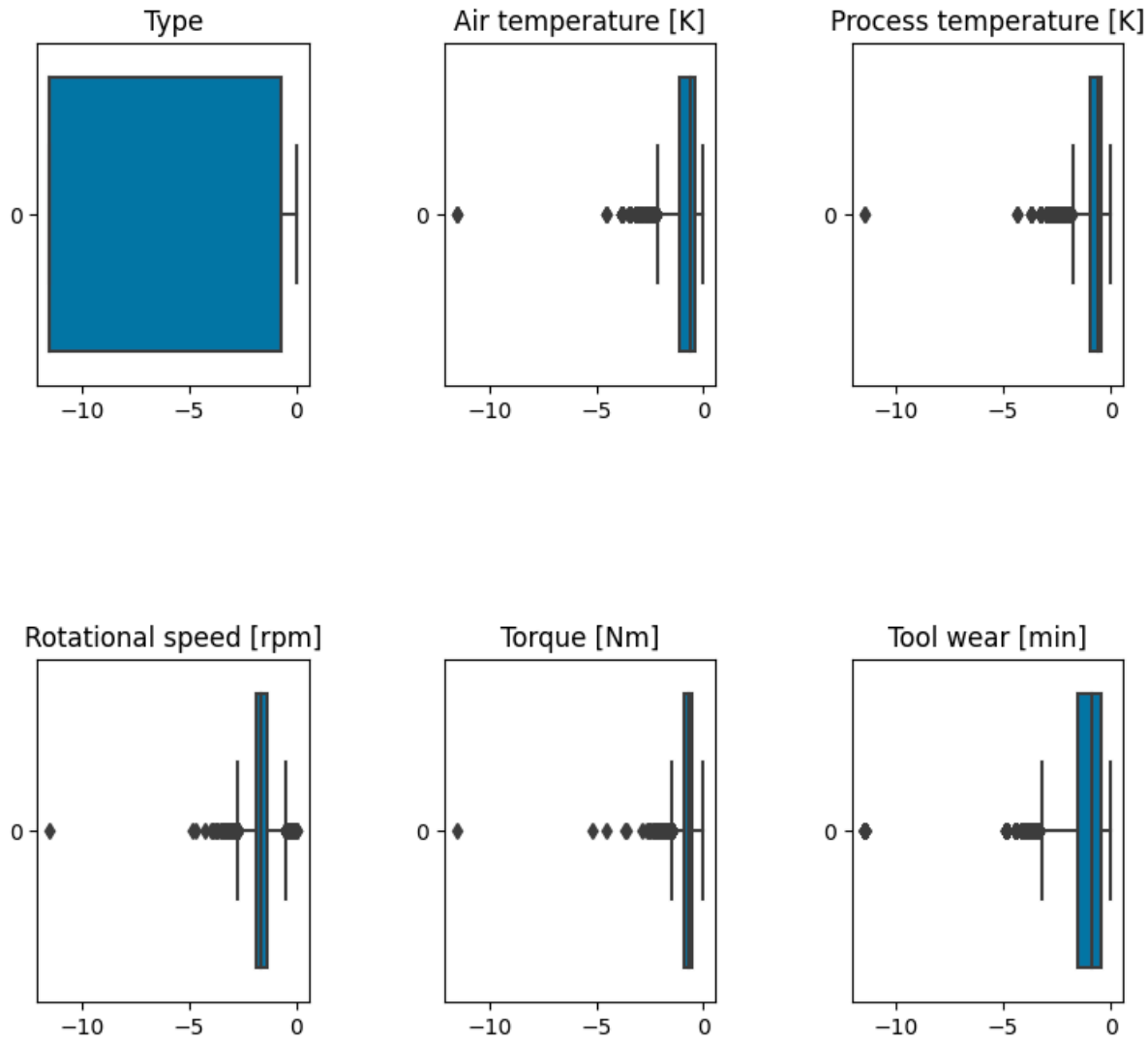


Figure 3.14 Boxplot of BoxCox transformation

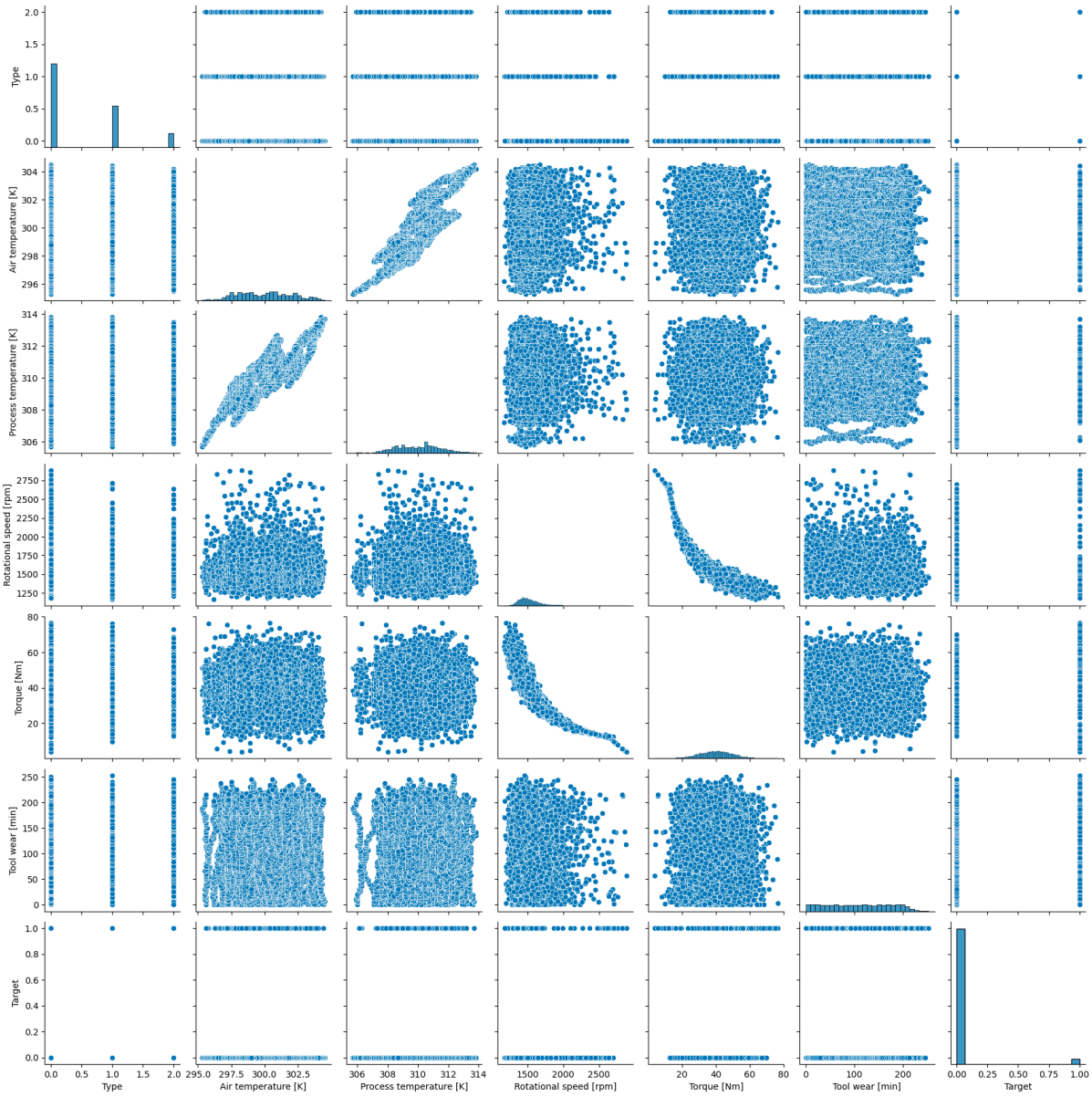


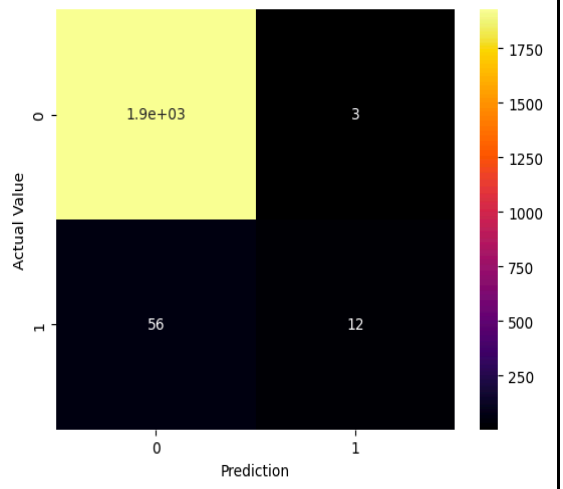
Figure 3.15 Pair Plot of every features

CHAPTER 4: MODEL AND ANALYSIS

Performance of different data distribution are first compared using Logistic Regression.

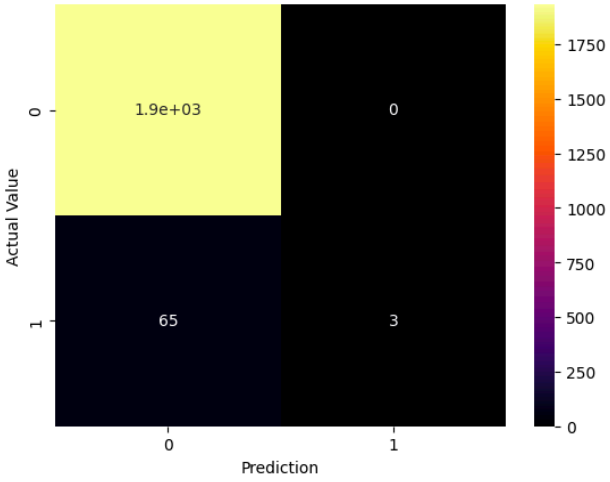
4.1 Logistic Regression

	precision	recall	f1-score	support
0	0.97	1.00	0.98	1932
1	0.80	0.18	0.29	68
accuracy			0.97	2000
macro avg	0.89	0.59	0.64	2000
weighted avg	0.97	0.97	0.96	2000



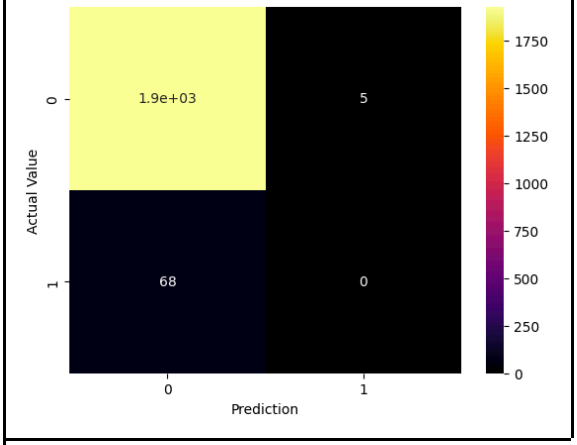
WITH DATA

	precision	recall	f1-score	support
0	0.97	1.00	0.98	1932
1	1.00	0.04	0.08	68
accuracy			0.97	2000
macro avg	0.98	0.52	0.53	2000
weighted avg	0.97	0.97	0.95	2000



WITH MIN MAX SCALING

	precision	recall	f1-score	support
0	0.97	1.00	0.98	1932
1	0.00	0.00	0.00	68
accuracy			0.96	2000
macro avg	0.48	0.50	0.49	2000
weighted avg	0.93	0.96	0.95	2000



With log min max

Figure 4.1 Comparison of Different Data Distribution of Logistic Regression

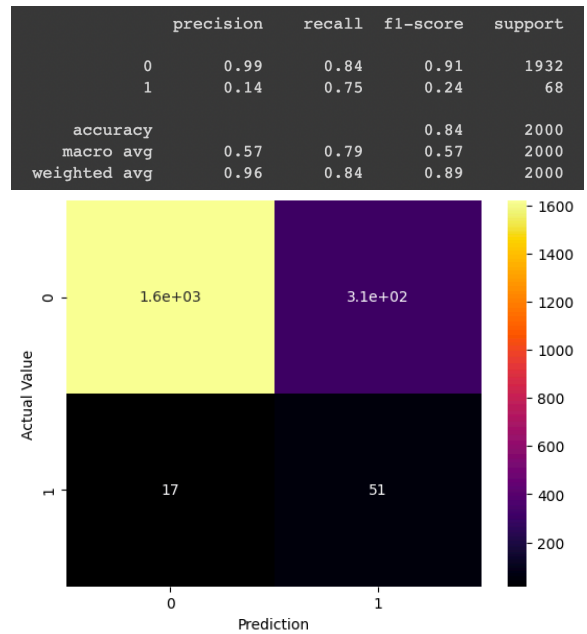
Recall of Target =1 is important metric in predictive maintenance and the use of data scaling hasn't improved much so data imbalance needs to be handled

4.1.1 SMOTE for handling class imbalance

SMOTE is an oversampling technique that generates synthetic samples from the minority class. It is used to obtain a synthetically class-balanced or nearly class-balanced training set, which is then used to train the classifier. The technique produces the following data set:

Table 4.1 SMOTE Sampling

0	7729
1	7729

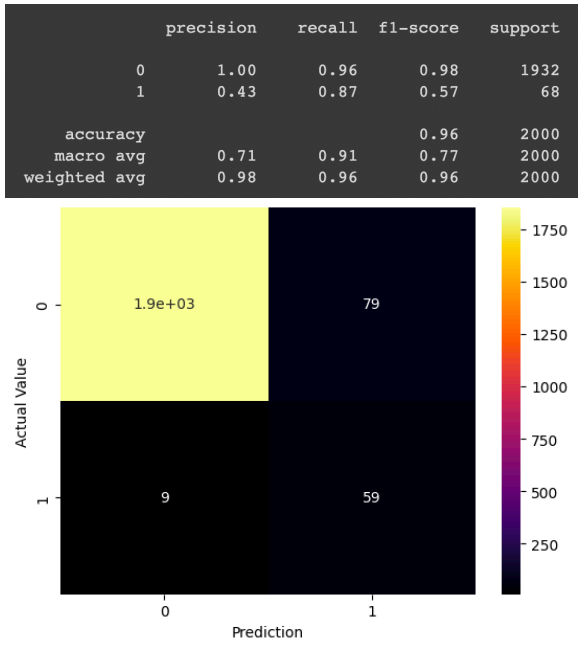


MIN MAX SCALER	With log min max

Figure 4.2 Comparison of Logistic Regression with SMOTE Sampling

The min-max scaler has higher accuracy and recall as well. As most of the literature suggests the use of Random Forest for such binary classification, both of these data scaling operations is tested using Random Forest.

4.2 RANDOM FOREST



With Log Min-Max scaling	With Min-Max scaling only

Figure 4.3 Comparison Log-Min-max and Min-max Scaling with Random Forest

As both have similar accuracy, we use simple model

Now, we compare three classifiers: Support Vector, Random Forest and KNN

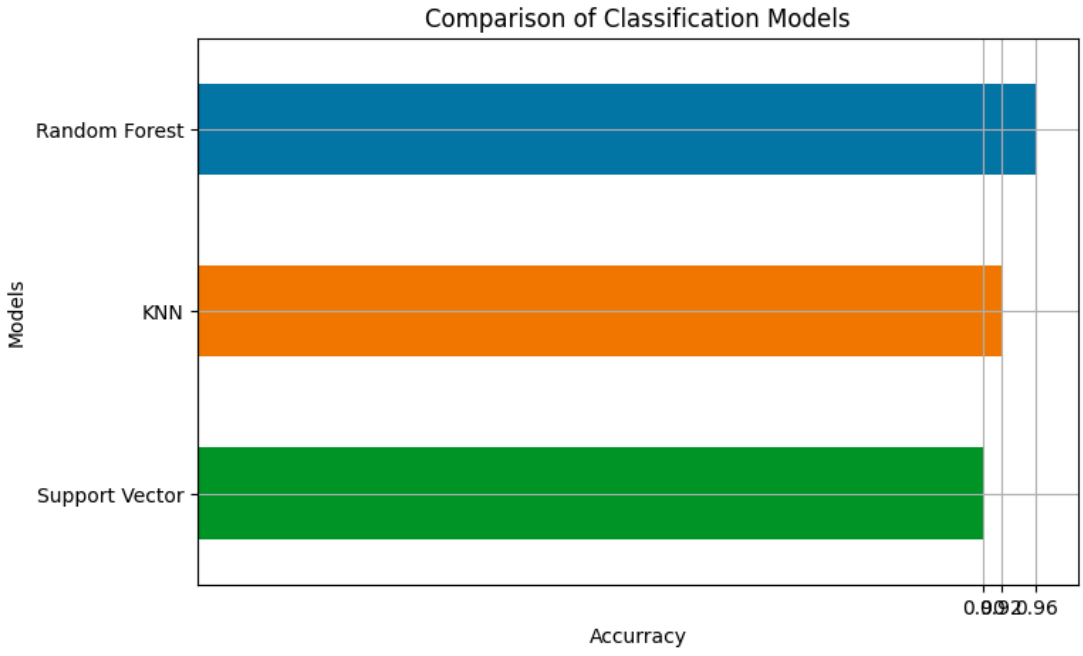


Figure 4.4 Comparison of three Machine Learning Algorithms

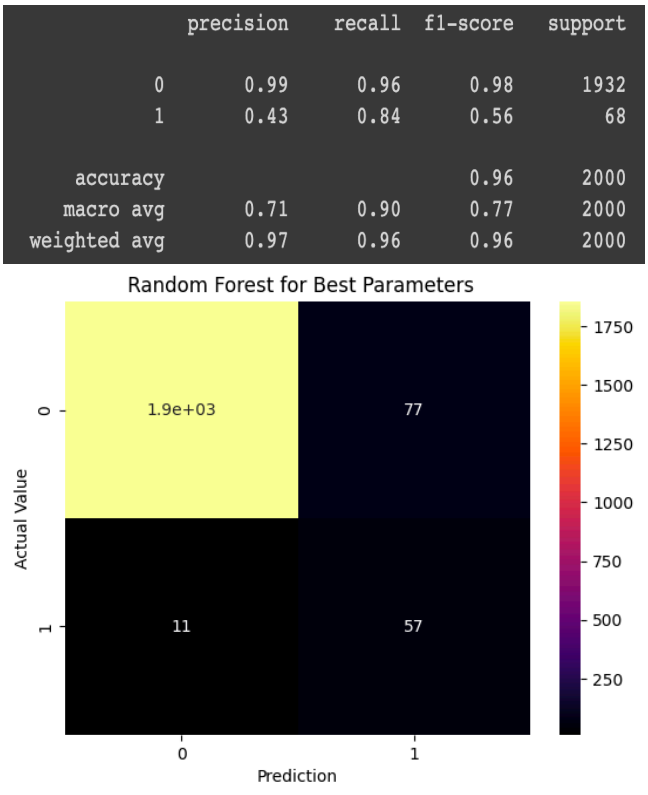
Following hyper-parameters were used with GridSearchCV for tuning:

Table 4.2 Hyper-parameter for Random Forest

Hyper-parameters	Value
n_estimators	[10, 50, 100, 200,250],
max_depth	[None, 10, 20,],
min_samples_split	[2, 5,],
min_samples_leaf	[1, 2,],

Best Random Forest model: RandomForestClassifier(n_estimators=200)

Best set of hyperparameters: {'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200}

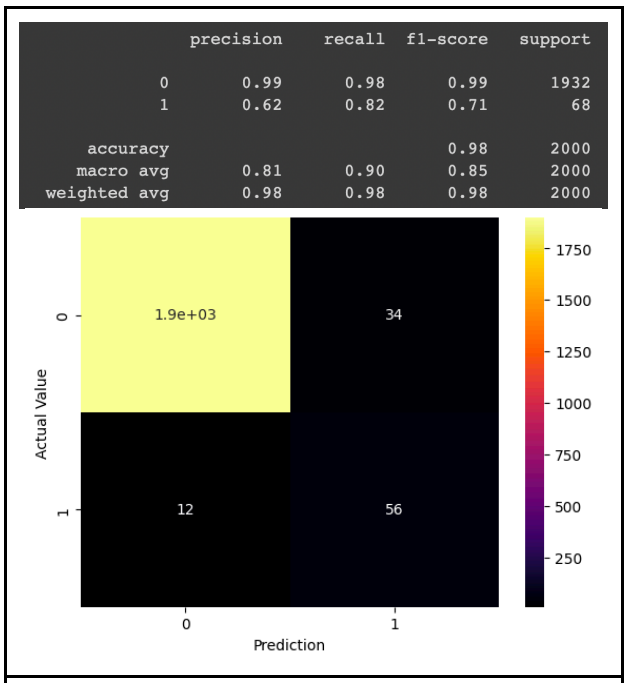


Estimation using best parameters from GridSearchCV

Figure 4.5 Performance with Best hyper-parameters

The accuracy of Random Forest is higher and the recall score of SVC and Random Forest are same, so Random Forest is selected for boosting to improve the model further.

4.3 XGB Boosting



Using XGB Boosting

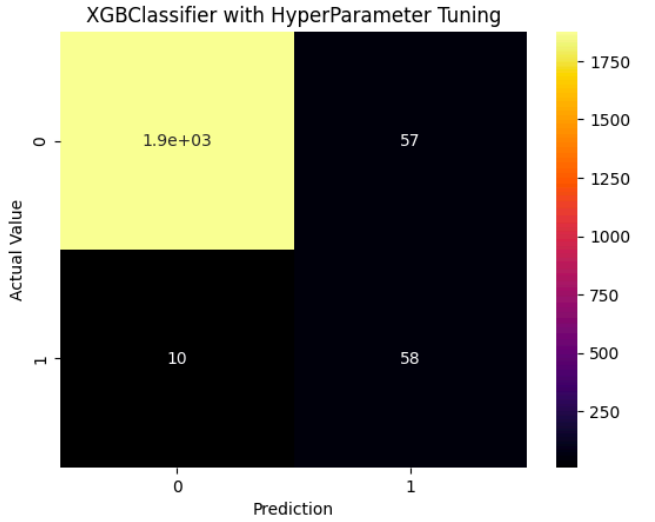
Figure 4.6 Classification Report and Confusion Matrix with XGB boosting

Table 4.3 Hyper Parameters for XGB Boosting

Hyper-parameters	Value
n_estimators	stats.randint(50, 200)
max_depth	stats.randint(0, 17)
subsample	stats.uniform(0.5, 0.5)
learning_rate	stats.randint(0, 17)

Best set of hyperparameters: {'learning_rate': None, 'max_depth': 7, 'n_estimators': 191, 'subsample': 0.9886610918271799}

	precision	recall	f1-score	support
0	0.99	0.97	0.98	1932
1	0.50	0.85	0.63	68
accuracy			0.97	2000
macro avg	0.75	0.91	0.81	2000
weighted avg	0.98	0.97	0.97	2000



XGB Boosting with tuned Hyper-parameters

Figure 4.7 Performance for Hyper-parameter with XGB boosting

CHAPTER 5: CONCLUSION

The project use data from Predictive Maintenance to predict Failure of Machines. Logistic Regression is first used to compare different data transformation methods. SMOTE method was used to handle the imbalanced data which increased the recall up to 0.75. Random Forest method was used to compare the recall and accuracy for two data transformation for Min-max scaling and log min-max scaling, which showed pretty similar performance in accuracy and recall. Comparison of SVC, Random Forest and KNN showed good accuracy with Random Forest. Random Forest was tuned with hyper-parameters which improved the recall to 0.84 and accuracy 0.96. XGB boosting doesn't much improve accuracy and recall.