

# **PREDICTIVE ANALYSIS OF MACHINERY FAILURE**

## **Table of contents:**

- ▶ Problem Setting
- ▶ Problem Objective and Problem Definition
- ▶ Data Source and Data Description
- ▶ Data Preprocessing and Dimension Reduction
- ▶ Exploratory Analysis
- ▶ Model Building
- ▶ Performance Evaluation
- ▶ Reference

**Problem Setting:**

Today, all businesses depend on machines, and we want them to work at their best for a long period. In most cases, Predictive maintenance can be used to maximize the efficiency of machinery. It helps to minimize unanticipated breakdowns as we can fix the machines just in time as we monitor and predict their status. Here, the problem is defined within the context of a manufacturing process, where the machine used in the process can fail due to various reasons. The goal is to understand the factors that contribute to machine failure and to predict when it is likely to occur.

**Problem Definition:**

The specific problem being addressed is to identify the factors that contribute to machine failure and to predict when it is likely to occur. The following questions need to be answered through data analytics:

- What are the factors that contribute to machine failure?
- How can we predict when a machine is likely to fail?
- Which failure mode is the most likely to occur?

**Data Sources:**

Stephan Matzka, School of Engineering - Technology and Life, Hochschule für Technik und Wirtschaft Berlin, 12459 Berlin, Germany, [stephan.matzka '@' htw-berlin.de](mailto:stephan.matzka@htw-berlin.de)

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

**Data Description:**

The dataset consists of 10,000 data points stored as rows with 14 features in columns. The variables include:

- UID: unique identifier ranging from 1 to 10,000
- product ID: 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
- air temperature [K]: generated using a random walk process later normalized to a standard deviation of 2 K around 300 K
- process temperature [K]: generated using a random walk process normalized to a standard deviation of 1 K, added to the air temperature plus 10 K.
- rotational speed [rpm]: calculated from a power of 2860 W, overlaid with a normally distributed noise
- torque [Nm]: torque values are normally distributed around 40 Nm with a  $\sigma$  = 10 Nm and no negative values.
- tool wear [min]: The quality variants H/M/L add 5/3/2 minutes of tool wear to the used tool in the process.
- 'machine failure' label: indicates whether the machine has failed in this particular datapoint.

**The machine failure consists of five independent failure modes:**

- tool wear failure (TWF): the tool will be replaced or fail at a randomly selected tool wear time between 200 – 240 mins (120 times in our dataset).

At this point in time, the tool is replaced 69 times, and fails 51 times (randomly assigned).

- heat dissipation failure (HDF): heat dissipation causes a process failure, if the difference between air- and process temperature is below 8.6 K and the tool's rotational speed is below 1380 rpm. This is the case for 115 data points.
- power failure (PWF): the product of torque and rotational speed (in rad/s) equals the power required for the process. If this power is below 3500 W or above 9000 W, the process fails, which is the case 95 times in our dataset.
- overstrain failure (OSF): if the product of tool wear and torque exceeds 11,000 minNm for the L product variant (12,000 M, 13,000 H), the process fails due to overstrain. This is true for 98 datapoints.
- random failures (RNF): each process has a chance of 0,1 % to fail regardless of its process parameters. This is the case for only 5 datapoints, less than could be expected for 10,000 datapoints in our dataset.

If at least one of the above failure modes is true, the process fails and the 'machine failure' label is set to 1. It is therefore not transparent to the machine learning method, which of the failure modes has caused the process to fail.

Summary Statistics:

Summary statistics of dataset

	UDI	Air temperature [K]	Process temperature [K]	\	
count	10000.00000	10000.000000	10000.000000		
mean	5000.50000	300.004930	310.005560		
std	2886.89568	2.000259	1.483734		
min	1.00000	295.300000	305.700000		
25%	2500.75000	298.300000	308.800000		
50%	5000.50000	300.100000	310.100000		
75%	7500.25000	301.500000	311.100000		
max	10000.00000	304.500000	313.800000		

	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]	Machine failure	\
count	10000.000000	10000.000000	10000.000000	10000.000000	
mean	1538.776100	39.986910	107.951000	0.033900	
std	179.284096	9.968934	63.654147	0.180981	
min	1168.000000	3.800000	0.000000	0.000000	
25%	1423.000000	33.200000	53.000000	0.000000	
50%	1503.000000	40.100000	108.000000	0.000000	
75%	1612.000000	46.800000	162.000000	0.000000	
max	2886.000000	76.600000	253.000000	1.000000	

	TWF	HDF	PWF	OSF	RNF
count	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000
mean	0.004600	0.011500	0.009500	0.009800	0.00190
std	0.067671	0.106625	0.097009	0.098514	0.04355
min	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000	0.000000	0.000000
75%	0.000000	0.000000	0.000000	0.000000	0.000000
max	1.000000	1.000000	1.000000	1.000000	1.000000

- Type: This is a categorical variable with three possible values: 'M', 'L', and 'H'. 'M' is the most frequent value, followed by 'H' and then 'L'.
- Air temperature [K]: This variable is normally distributed, with a mean of 300K and a standard deviation of 2.571K.
- Process temperature [K]: This variable is normally distributed, with a mean of 310K and a standard deviation of 1.338K.
- Rotational speed [rpm]: This variable is bimodal, with two peaks at around 1200rpm and 1600rpm.
- Torque [Nm]: This variable is skewed to the right, with a mean of 40.003Nm and a standard deviation of 9.188Nm.
- Tool wear [min]: This variable is skewed to the right, with a mean of 107.951min and a standard deviation of 63.654min.

- Machine failure: This is a binary variable that indicates whether the machine failed or not. There were 98 instances of machine failure in the dataset, out of a total of 10000 instances.
- TWF: This is a binary variable that indicates whether there was a tool wear failure or not. There were 196 instances of tool wear failure in the dataset, out of a total of 10000 instances.
- HDF: This is a binary variable that indicates whether there was a heat dissipation failure or not. There were 221 instances of heat dissipation failure in the dataset, out of a total of 10000 instances.
- PWF: This is a binary variable that indicates whether there was a power failure or not. There were 140 instances of power failure in the dataset, out of a total of 10000 instances.
- OSF: This is a binary variable that indicates whether there was an overstrain failure or not. There were 128 instances of overstrain failure in the dataset, out of a total of 10000 instances.
- RNF: This is a binary variable that indicates whether there was a random failure or not. There were 38 instances of random failure in the dataset, out of a total of 10000 instances.

Histograms for each numerical variables in the dataset:

From the resulting histograms below, we can infer the following:

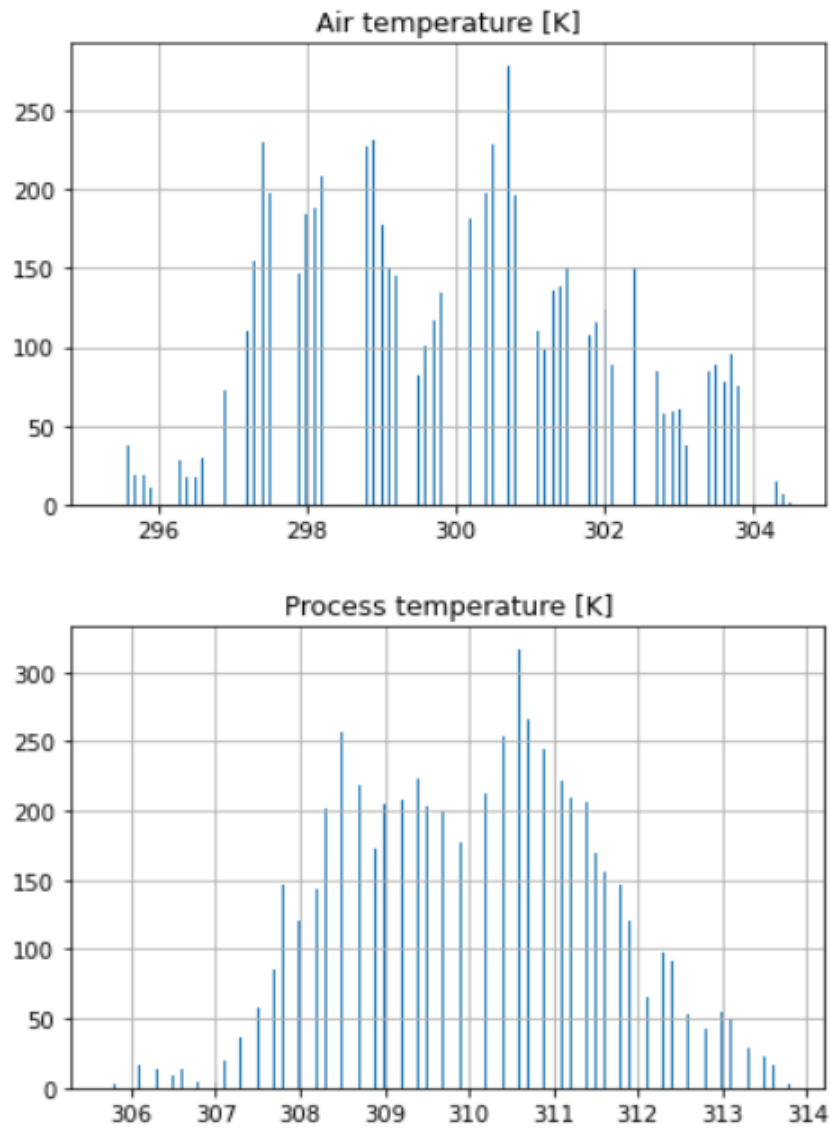
Air temperature [K]: The distribution is approximately normal with a slight left skew. The majority of the values fall between 295 K and 305 K.

Process temperature [K]: The distribution is approximately normal with a slight left skew. The majority of the values fall between 310 K and 320 K.

Rotational speed [rpm]: The distribution is roughly bimodal with peaks around 1375 and 1725 rpm. There is a dip in the middle around 1550 rpm.

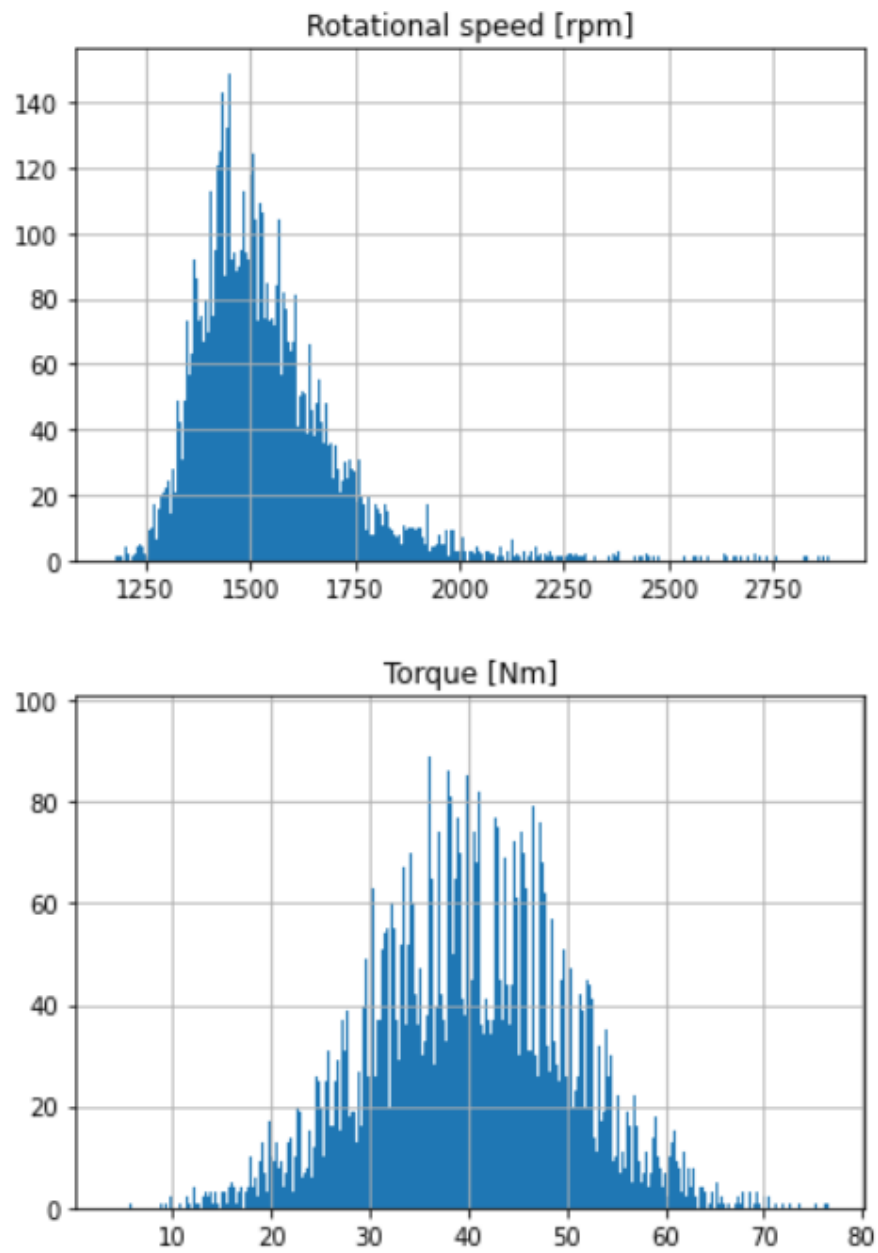
Torque [Nm]: The distribution is approximately normal with a slight left skew. The majority of the values fall between 25 Nm and 50 Nm.

Tool wear [min]: The distribution is right-skewed, with most values falling below 50 minutes. There is a long tail of values up to around 200 minutes.

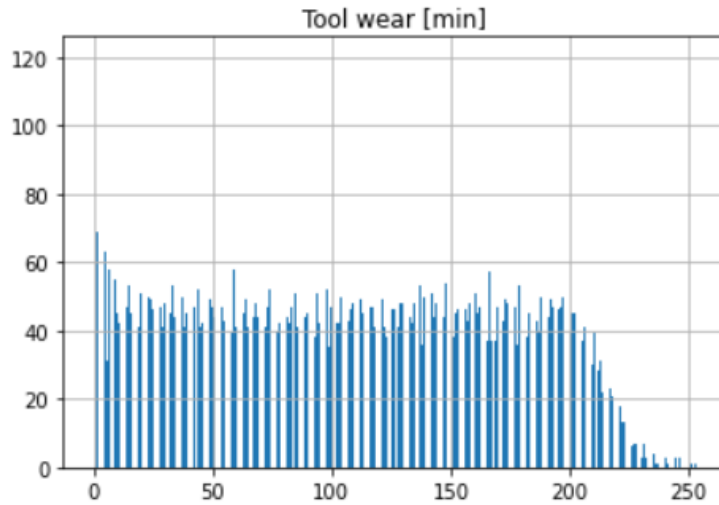


**Fig a – Histogram of numeric variables of Air temperature and Process Temperature**





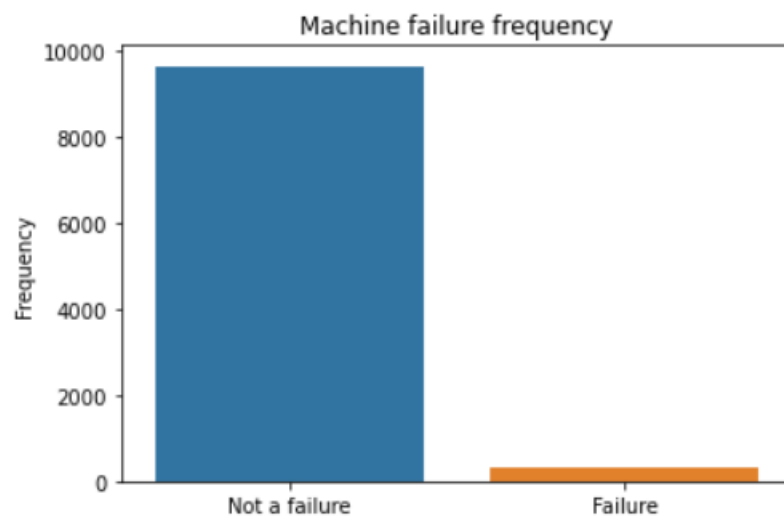
**Fig b – Histogram of Rotational and Torque**



**Fig c – Histogram of numeric variables of Tool Wear**

Exploratory Data Analysis:

Frequency of machine failure:



**Fig – Frequency of machine failure**

The resulting graph shows the frequency of machine failure in the dataset. It indicates that the majority of the machines did not experience failure, as there are significantly more instances of "Not a failure" (labeled as 0) compared to "Failure" (labeled as 1). However, it is important to note that there is still a significant number of machines that did experience failure, and this can be further analyzed to determine the factors contributing to these failures.

### Grouping the dataset by Type

	Machine failure	TWF	HDF	PWF	OSF	RNF
Type						
H	21	7	8	5	2	4
L	235	25	76	59	87	13
M	83	14	31	31	9	2

The above dataset is a resulting dataset of grouping Machine failure TWF HDF PWF OSF RNF with Type to plot the frequency of machine failure with respect to Type variable.

**Pie chart for each category of failure types:**

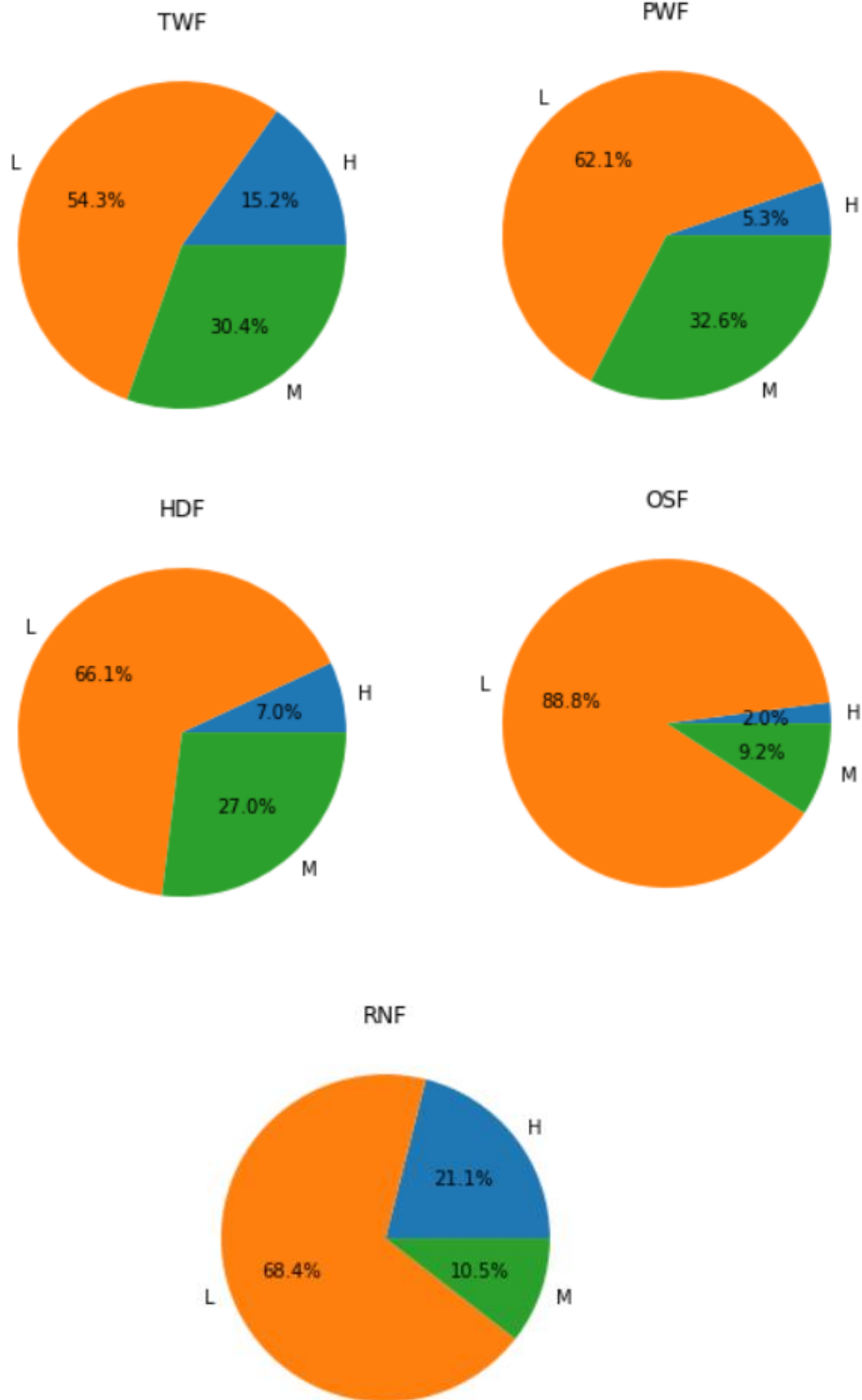


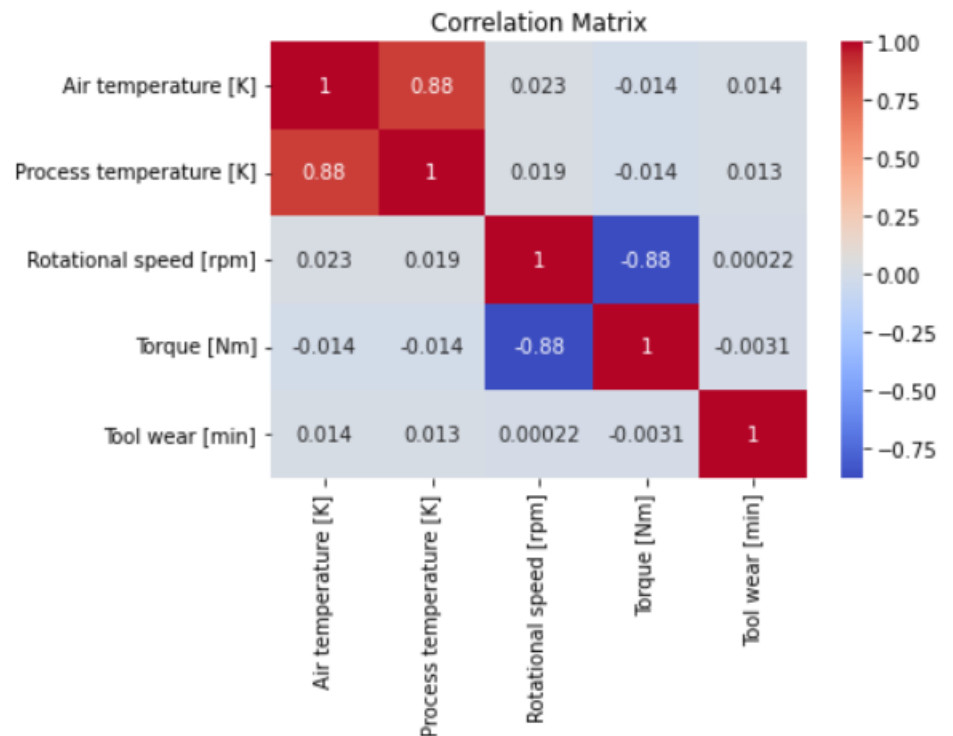
Fig- Pie charts of different failure modes.

The resulting graphs are pie charts that show the percentage distribution of each variable ('Machine failure', 'TWF', 'HDF', 'PWF', 'OSF', 'RNF') across the different types of products. From these pie charts, we can infer:

- Machine failure occurred frequently in type H compared to types M and L.
- TWF occurred more frequently in type M compared to types H and L.
- HDF occurred more frequently in type H compared to types M and L.
- PWF occurred more frequently in type H compared to types M and L.
- OSF occurred more frequently in type L compared to types H and M.
- RNF occurred more frequently in type L compared to types H and M.

Overall, it appears that there are some differences in the distribution of the variables across the different types. However, further analysis and statistical testing would be needed to confirm these findings.

Correlation plot:



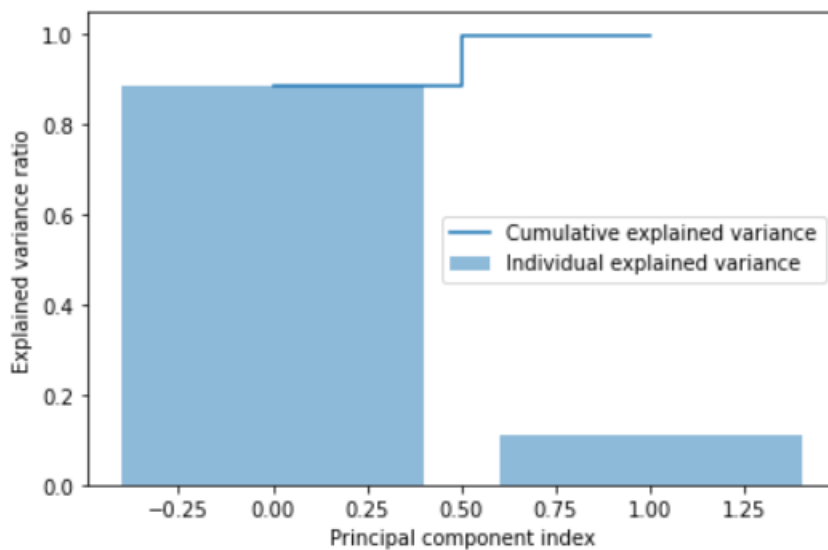
In the given correlation matrix, we can see that the variables with the highest positive correlations are "Machine failure", "PWF", "OSF", and "HDF", indicating that they are strongly associated with each other. We can also see that "Air temperature [K]" and "Process temperature [K]" are highly positively correlated with each other, which is not surprising since they both relate to temperature. On the other hand, "Rotational speed [rpm]" and "Torque [Nm]" have a high negative

correlation coefficient, indicating that they are inversely related, meaning that as one variable increases, the other tends to decrease.

## Dimension Reduction:

We tried to handle the dimension reduction part by applying Principal component analysis to the data and to handle the data by fitting a logistic regression model on the PCA components and we achieved an accuracy of 96.95%.

The variance captured by the two elements are 0.887 and 0.112.



## Handling Imbalanced Data:

We tried handling the imbalanced data by using 2 different ways, SMOTE and RUS and fitting the logistic regression model on the balanced data.

```
Accuracy: 59.45%
Classification Report:
              precision    recall  f1-score   support

     0           0.98       0.59       0.74       1939
     1           0.04       0.61       0.08         61

 accuracy          0.59       0.59       0.59       2000
 macro avg         0.51       0.60       0.41       2000
 weighted avg      0.95       0.59       0.72       2000
```

Confusion Matrix:

```
[[1152  787]
 [   24   37]]
```

This was by using SMOTE.

```
Accuracy: 59.45%
Classification Report:
              precision    recall  f1-score   support

     0           0.98       0.59       0.74       1939
     1           0.05       0.62       0.09         61

 accuracy          0.59       0.59       0.59       2000
 macro avg         0.51       0.61       0.41       2000
 weighted avg      0.95       0.59       0.72       2000
```

Confusion Matrix:

```
[[1151  788]
 [   23   38]]
```

This was by using RUS.

## Exploration of Candidate Data Mining Models and Selecting the Final Model.

Selecting only the relevant columns and applying on-hot encoding on the Type column.

Creating the train and test sets as X\_t, X\_te, y\_t, y\_te for model building.

Creating train and evaluation set as X\_train, X\_test, y\_train, y\_test.

Applying lazypredict library to see the baseline estimates of model performance:

Model	Accuracy	Balanced Accuracy	ROC AUC	F1 Score	\
LGBMClassifier	0.98	0.84	0.84	0.98	
BaggingClassifier	0.98	0.83	0.83	0.98	
DecisionTreeClassifier	0.98	0.83	0.83	0.98	
XGBClassifier	0.98	0.82	0.82	0.98	
NearestCentroid	0.73	0.78	0.78	0.81	
RandomForestClassifier	0.98	0.76	0.76	0.98	
AdaBoostClassifier	0.97	0.75	0.75	0.97	
ExtraTreeClassifier	0.96	0.70	0.70	0.96	
LinearDiscriminantAnalysis	0.97	0.69	0.69	0.96	
ExtraTreesClassifier	0.98	0.69	0.69	0.97	
LabelSpreading	0.97	0.69	0.69	0.96	
LabelPropagation	0.97	0.69	0.69	0.96	
Perceptron	0.96	0.67	0.67	0.96	
QuadraticDiscriminantAnalysis	0.96	0.66	0.66	0.96	
CalibratedClassifierCV	0.97	0.65	0.65	0.96	
LogisticRegression	0.97	0.65	0.65	0.96	
KNeighborsClassifier	0.97	0.64	0.64	0.96	
SGDClassifier	0.97	0.64	0.64	0.96	
SVC	0.97	0.63	0.63	0.96	
GaussianNB	0.96	0.60	0.60	0.95	
LinearSVC	0.97	0.59	0.59	0.96	
PassiveAggressiveClassifier	0.97	0.56	0.56	0.95	
RidgeClassifierCV	0.96	0.50	0.50	0.94	
RidgeClassifier	0.96	0.50	0.50	0.94	
DummyClassifier	0.96	0.50	0.50	0.94	
BernoulliNB	0.96	0.50	0.50	0.94	

From the above results we can see XGBoost, Random Forest, Decision Trees Classification, AdaBoost classifier and KNeighborsclassifier algorithms are performing well

Going ahead with the above algorithmic experimentation,

### 1)Applying XGBoost algorithm

- Performing feature scaling using Normalization technique.
- Transforming the X\_train.
- Applying modelling.



- Transforming the X\_test
- Accuracy score for y\_te is 0.98
- Classification report for the predictions and y\_test


	precision	recall	f1-score	support
0	1.00	0.99	0.99	1748
1	0.65	0.85	0.73	52
accuracy			0.98	1800
macro avg	0.82	0.92	0.86	1800
weighted avg	0.99	0.98	0.98	1800

## 2) Applying Random Forest Classifier

- Fitting the random forest classifier : RandomForestClassifier(max\_depth=5, random\_state=0)
- Accuracy score for y\_test is 0.97111
- Classification report for the random forest classifier

	precision	recall	f1-score	support
0	1.00	0.97	0.99	1780
1	0.26	0.90	0.41	20
accuracy			0.97	1800
macro avg	0.63	0.94	0.70	1800
weighted avg	0.99	0.97	0.98	1800

Setting max\_depth to 3 and random state to 0

	precision score: 0.9921146922972421				
	Accuracy score: 0.9683333333333334				
	precision	recall	f1-score	support	
0	1.00	0.97	0.98	1785	
1	0.19	0.87	0.31	15	
accuracy			0.97	1800	
macro avg	0.60	0.92	0.65	1800	
weighted avg	0.99	0.97	0.98	1800	

Setting max\_depth to 7 and random state to 0

```

Accuracy 0.975
      precision    recall  f1-score   support

     0       1.00      0.98      0.99       992
     1       0.23      0.88      0.36         8

 accuracy
macro avg      0.61      0.93      0.67      1000
weighted avg    0.99      0.97      0.98      1000

```

Setting max\_depth to 1 and random state to 0

```

Accuracy 0.969
      precision    recall  f1-score   support

     0       1.00      0.97      0.98      1000
     1       0.00      0.00      0.00         0

 accuracy
macro avg      0.50      0.48      0.49      1000
weighted avg    1.00      0.97      0.98      1000

```

it can be observed that by changing the max\_depth parameter from 5 to 3 to 7 and 1 in a Random Forest Classifier model would decrease the maximum depth of each decision tree in the random forest from 5 to 3 to 7 to 1 which would eventually reduce the complexity of the model and prevent overfitting, as decision trees with deeper levels of splitting may capture too much noise and may not generalize well to unseen data. So, choosing 7 as max\_depth would prevent overfitting and help acquire better model performance which means max\_depth=7 might also be optimal.

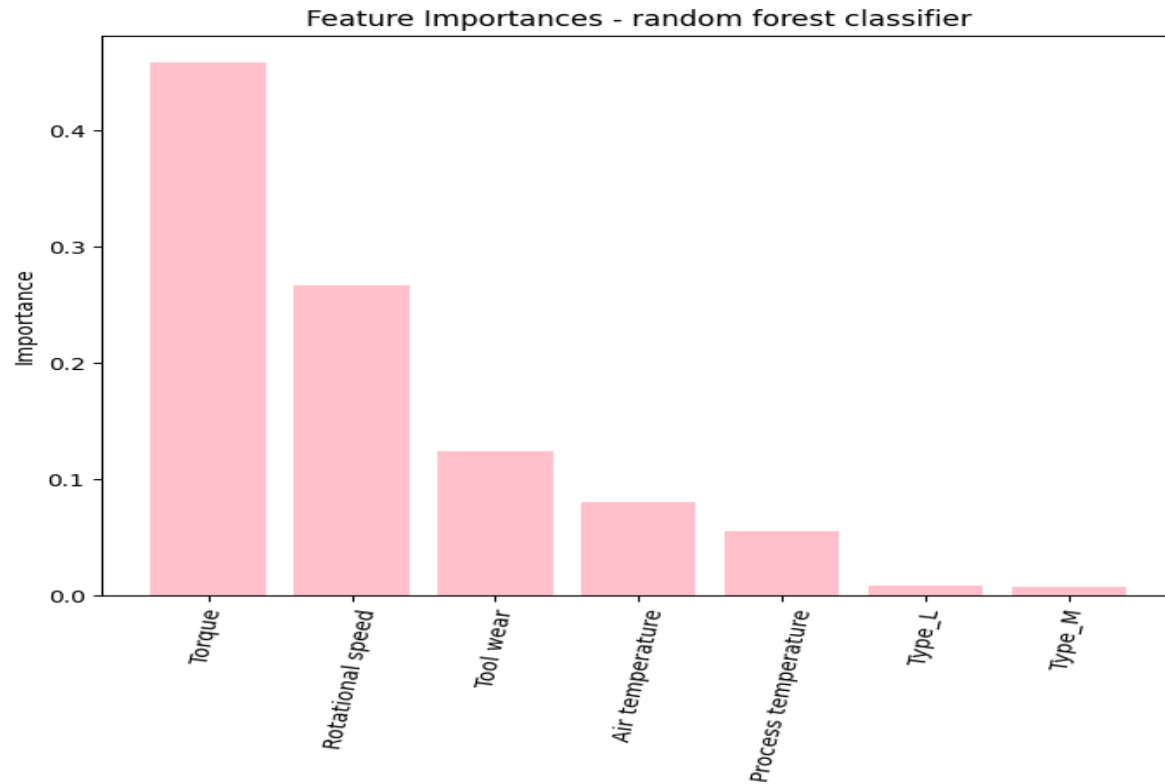


Fig – Feature importance of random forest classifier

### 3) Applying decision tree classifier

- Fitting the Decision tree classifier: `DecisionTreeClassifier(random_state=0)`
- Accuracy score for `y_test` is 0.978
- Classification report for decision tree classifier

Accuracy 0.978					
	precision	recall	f1-score	support	
0	0.99	0.99	0.99	973	
1	0.58	0.67	0.62	27	
accuracy			0.98	1000	
macro avg	0.79	0.83	0.80	1000	
weighted avg	0.98	0.98	0.98	1000	

#### 4) Applying Adaboost classifier

- Fitting the adaboost classifier
- Accuracy score for y\_test is 0.974
- Classification report for Adaboost classifier

```
Accuracy 0.974
              precision    recall  f1-score   support

         0       0.99      0.98      0.99       981
         1       0.39      0.63      0.48        19

 accuracy
macro avg      0.69      0.81      0.73      1000
weighted avg   0.98      0.97      0.98      1000
```

#### 5) Applying KNeighborsClassifier

- Fitting the KNeighborsclassifier
- Accuracy score for y\_test is 0.969
- Classification report for KNeighborsclassifier

```
Accuracy 0.969
              precision    recall  f1-score   support

         0       1.00      0.97      0.98       992
         1       0.13      0.50      0.21         8

 accuracy
macro avg      0.56      0.74      0.59      1000
weighted avg   0.99      0.97      0.98      1000
```

Here the XGBoost algorithm gives us the highest accuracy over all the algorithms experimented and we choose this algorithm that perfectly fits our model and we continue to use this algorithm for Model Performance Evaluation.

### Comparing the accuracy and Precision Scores for different models used :

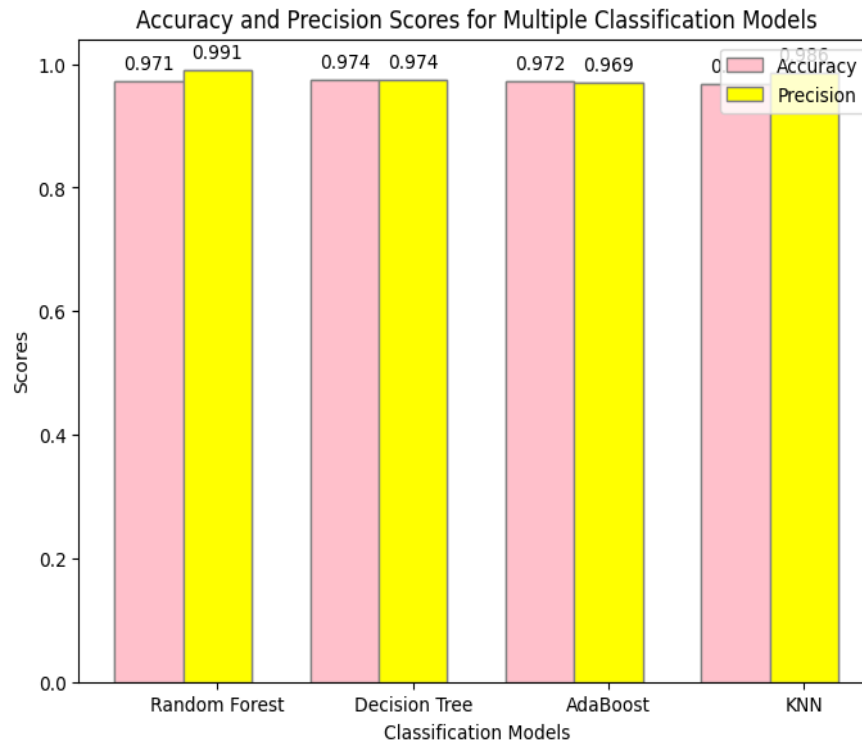


Fig – Accuracy and precision scores of different models

It can be observed that there is a very minute difference between the classifiers used and every model used gives a high performance and accuracy rate.

## Performance Evaluation: Comparison of model performance

Model Name	Accuracy	Precision
XGBoost	0.98	0.99
Random Forest	0.975	0.99
Decision Trees	0.978	0.98
AdaBoost classifier	0.974	0.99
K-Neighbors classifier	0.969	0.99

### Roc curve:

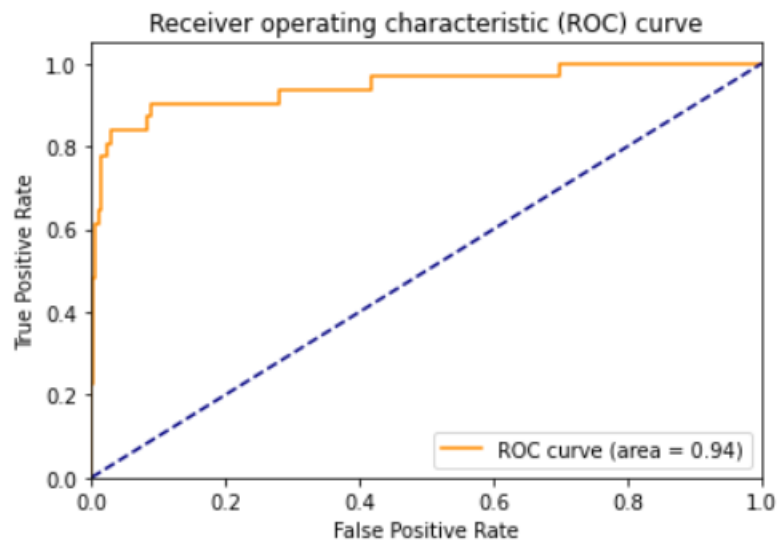


Fig – Area under ROC-AUC curve

### Confusion Matrix:

Confusion Matrix:

```
[[1725   7]
 [  50  18]]
```

Accuracy score for the  $y_{te}$  is **0.98**

## Classification Report :

	precision	recall	f1-score	support
0	0.99	0.98	0.99	979
1	0.52	0.76	0.62	21
accuracy			0.98	1000
macro avg	0.76	0.87	0.80	1000
weighted avg	0.98	0.98	0.98	1000

The area under the ROC curve is 0.94 which suggests that the classifier and the algorithm used has an excellent performance in distinguishing between positive and negative cases. An AUC of 0.94 indicates that the algorithm used has correctly identified positive cases with a high degree of accuracy while keeping the false positive rate relatively low. Typically, an AUC value of 0.5 suggests that the classifier is no better than random guessing, and an AUC value of 1.0 indicates that the classifier is making perfect predictions. Therefore, an AUC value of 0.94 is considered very good and suggests that the model has strong predictive power.

## Conclusion and Key take aways:

- We have implemented five models XGBoost, Random Forest, Decision Trees, AdaBoost classifier, and K-Neighbors classifier.
- After analyzing various algorithms, XGBoost demonstrated the maximum accuracy, making it the best option for our model with an accuracy of 0.98 and a precision of 0.99.
- The area under the ROC curve is 0.94 which suggests that the classifier and the algorithm used has an excellent performance in distinguishing between positive and negative cases.
- An AUC of 0.94 indicates that the algorithm used has correctly identified positive cases with a high degree of accuracy while keeping the false positive rate relatively low
- Ultimately, we can draw a conclusion that our model predicts the failure of the machine with 98% accuracy which shows that the model works well.

