AI4I 2020 Predictive Maintenance

Project Report

Guidance: Dr. Sirkeci, Birsen

By:
B Sai Vikram Adithya
Ravi Amulya

San Jose State University

Project Introduction:

This venture AI4I (Man-made reasoning for Industry) Predicative Upkeep Dataset is accessible freely in the UCI AI Storehouse. It is made to create and assessing prescient upkeep calculations. As per the dataset given there are various boundaries, for example, "Item ID, Type, Air Temperature, Cycle Temperature, Rotational Speed, Force, Machine Disappointment, Device Wear Disappointment, Intensity Scattering Disappointment, Power Disappointment, Overspeed Disappointment, and Irregular Disappointment".

Using other parameters as inputs and keeping the target variable as Machine Failure, we will now construct Machine Learning Models. We are developing the K-NN Algorithm, Decision Tree, Logistic Regression, and Random-Forest Algorithm in this project report.

Data Set Description:

There are a number of samples in the AI4I Predictive Maintenance Dataset, and each sample represents a different kind of product. 14 columns, or characteristics, are included in each sample to record various product-related measurements and characteristics.

We are developing the model and selecting Machine Failure as the **Target Variable.**

There are no in the dataset. of lines or tests with 14 segments or elements. Each column addresses a particular item with its relating estimations and properties.

UDI: Unique Device Identifier, which is a one-of-a-kind product identification number Product ID: The type of the product identification number: The sort of item, which can be L (low), M (medium), or H (high)

Air temperature [K]: The temperature of the air during the manufacturing process, expressed in Kelvin (K) The cycle temperature during the assembling system, estimated in Kelvin (K)

Rotational speed [rpm]: The rotational speed of the item during the assembling system, estimated in cycles each moment (rpm)

Force [Nm]: The product's torque, expressed in Newton-meters (Nm), during the manufacturing process. Tool wear, measured in minutes: The amount of time spent using the tool, expressed in minutes (min) Machine failure: a binary variable with a value of 1 indicating that the machine failed and a value of 0 indicating that it did not fail during the manufacturing process.

TWF: A binary variable that indicates whether there was a tool wear failure, with a value of 1 indicating a failure and a value of 0 indicating that there was no failure.

HDF: Failure of heat dissipation, a binary variable that indicates whether or not there was a failure of heat dissipation, with a value of 1 indicating a failure and a value of 0 indicating that there was no failure.

PWF: Power disappointment, a twofold factor demonstrating whether there was a power disappointment, with 1 showing that there was a disappointment and 0 showing that there was not.

OSF: A binary variable that indicates whether there was an out-of-spec failure, with a value of 1 indicating a failure and a value of 0 indicating that there was not.

RNF: A binary variable that indicates whether there was a random failure, with a value of 1 indicating a failure and a value of 0 indicating that there was not. Random failures

The dataset can be used to investigate the connections between the various variables and discover any manufacturing process patterns or anomalies. It can likewise be utilized to assemble prescient models to expect machine or apparatus disappointments during assembling.

Building Dataset Visualization for this dataset:

```
In [20]: import pandas as pd

         # Load the dataset
         df = pd.read_csv("ai4i2020.csv")

         # Drop the "Product ID" and "UDI" columns
         df = df.drop(["Product ID", "UDI"], axis=1)
```

```
In [3]: # Data Cleaning we are not doing since no null values
```

```
In [21]: df.head()
```

Out[21]:

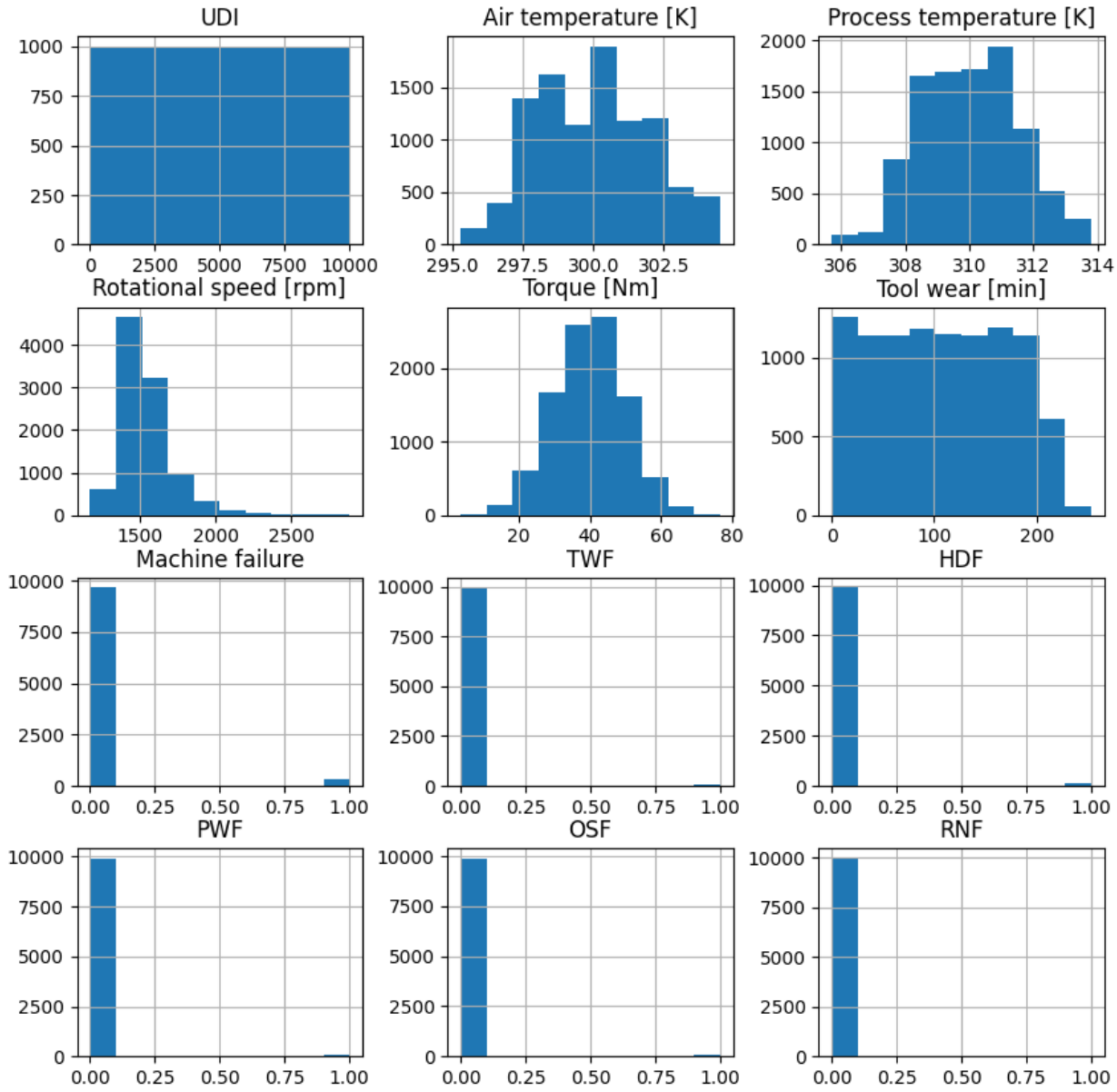
	Type	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]	Machine failure	TWF	HDF	PWF	OSF	RNF
0	M	298.1	308.6	1551	42.8	0	0	0	0	0	0	0
1	L	298.2	308.7	1408	46.3	3	0	0	0	0	0	0
2	L	298.1	308.5	1498	49.4	5	0	0	0	0	0	0
3	L	298.2	308.6	1433	39.5	7	0	0	0	0	0	0
4	L	298.2	308.7	1408	40.0	9	0	0	0	0	0	0

Building Dataset Information for this dataset:

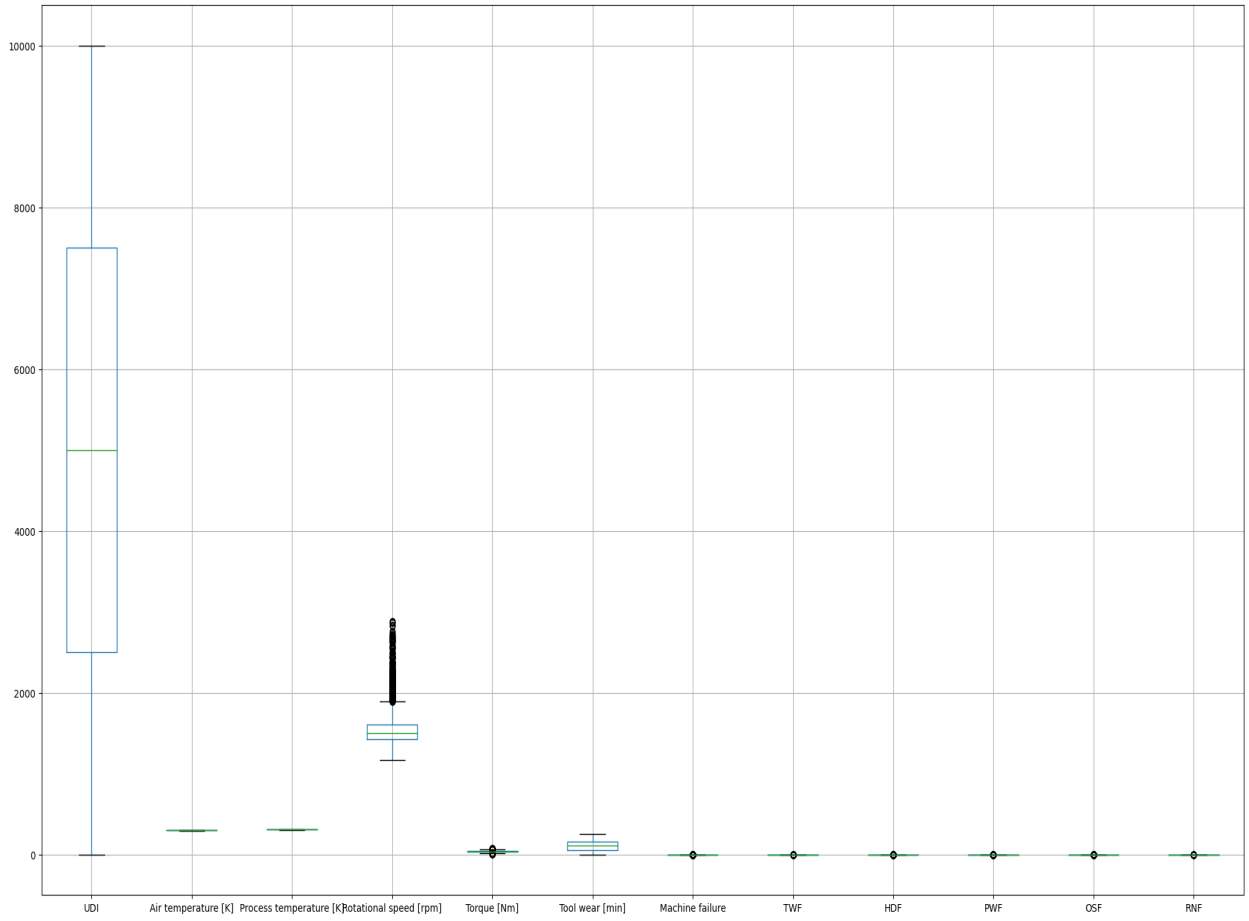
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 12 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Type                     10000 non-null  int64
 1   Air temperature [K]      10000 non-null  float64
 2   Process temperature [K]  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   Machine failure          10000 non-null  int64
 7   TWF                      10000 non-null  int64
 8   HDF                      10000 non-null  int64
 9   PWF                      10000 non-null  int64
 10  OSF                      10000 non-null  int64
 11  RNF                      10000 non-null  int64
dtypes: float64(3), int64(9)
memory usage: 937.6 KB
```

According to this data we can say that there are no null values.

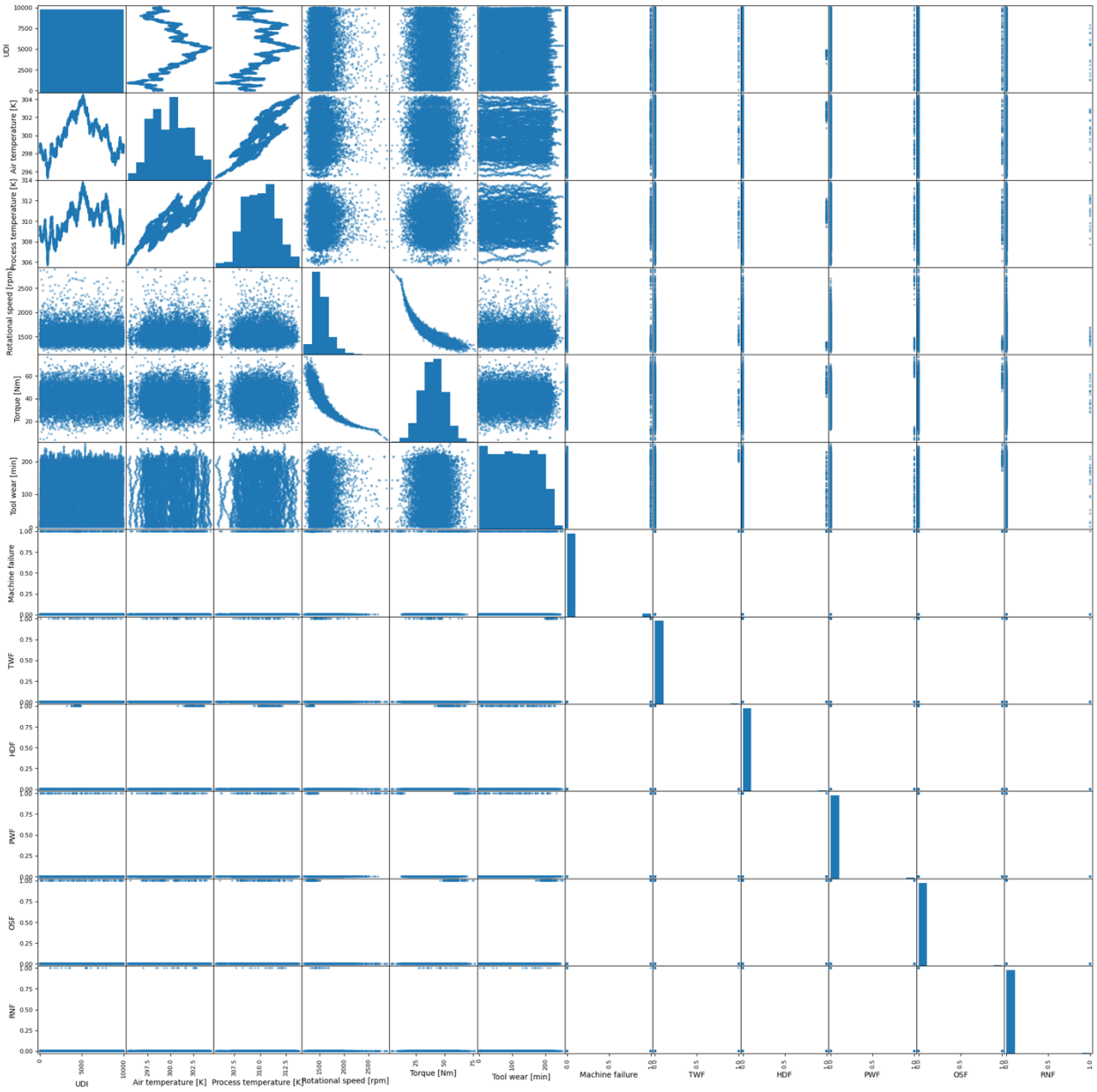
Building Histogram for this dataset:



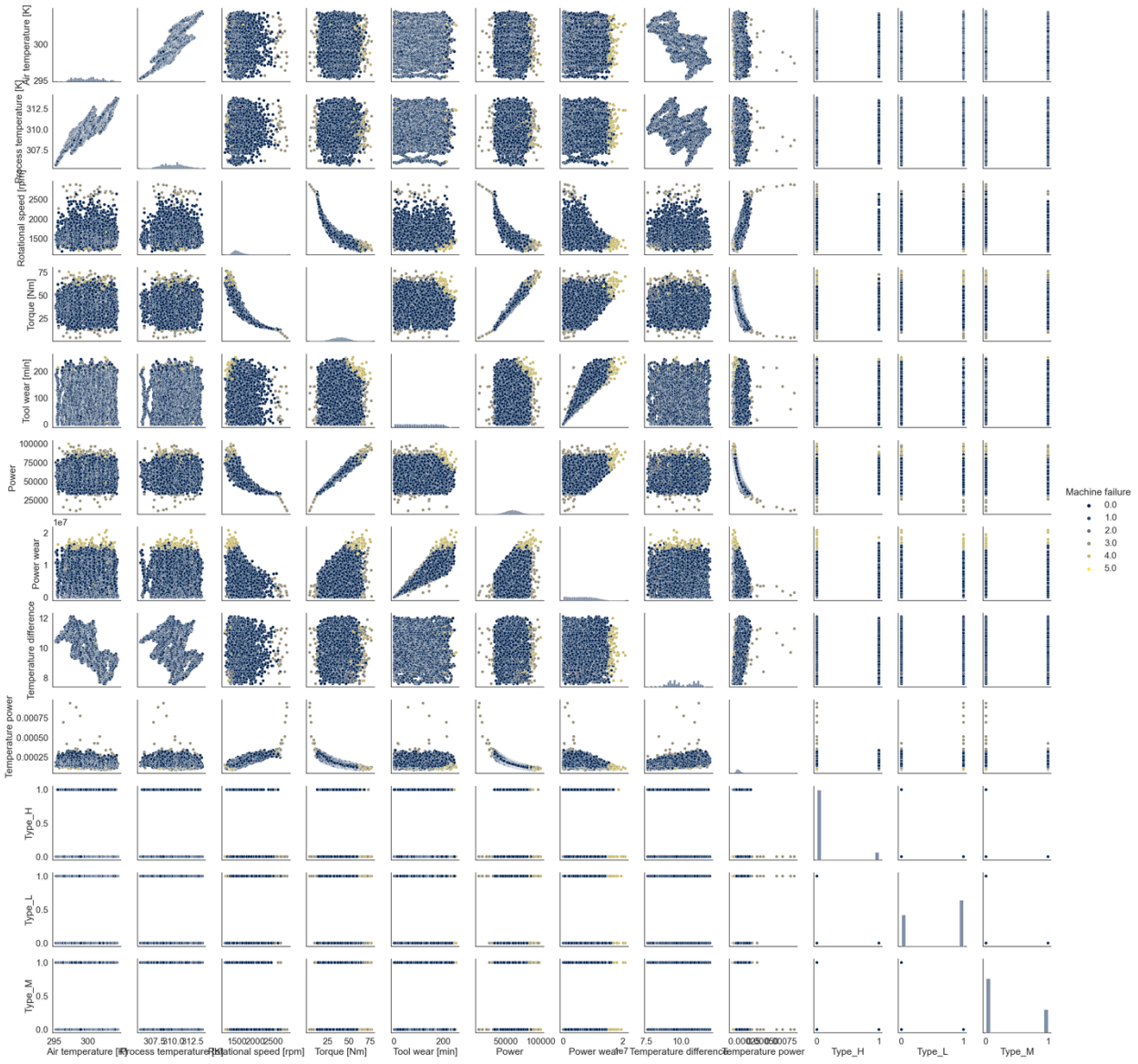
Building Box-plot for this dataset:



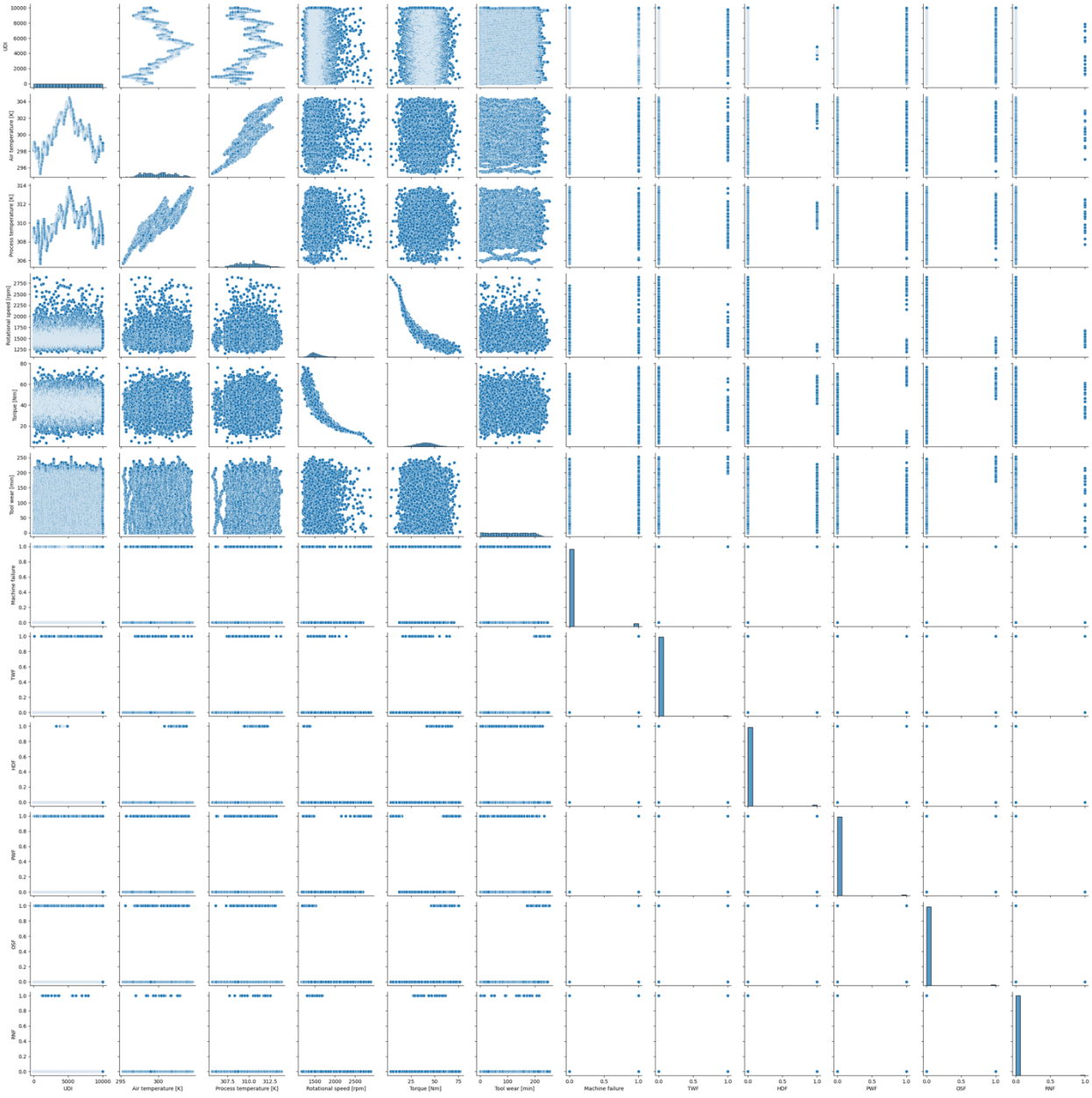
Building scatter-plot for this dataset:

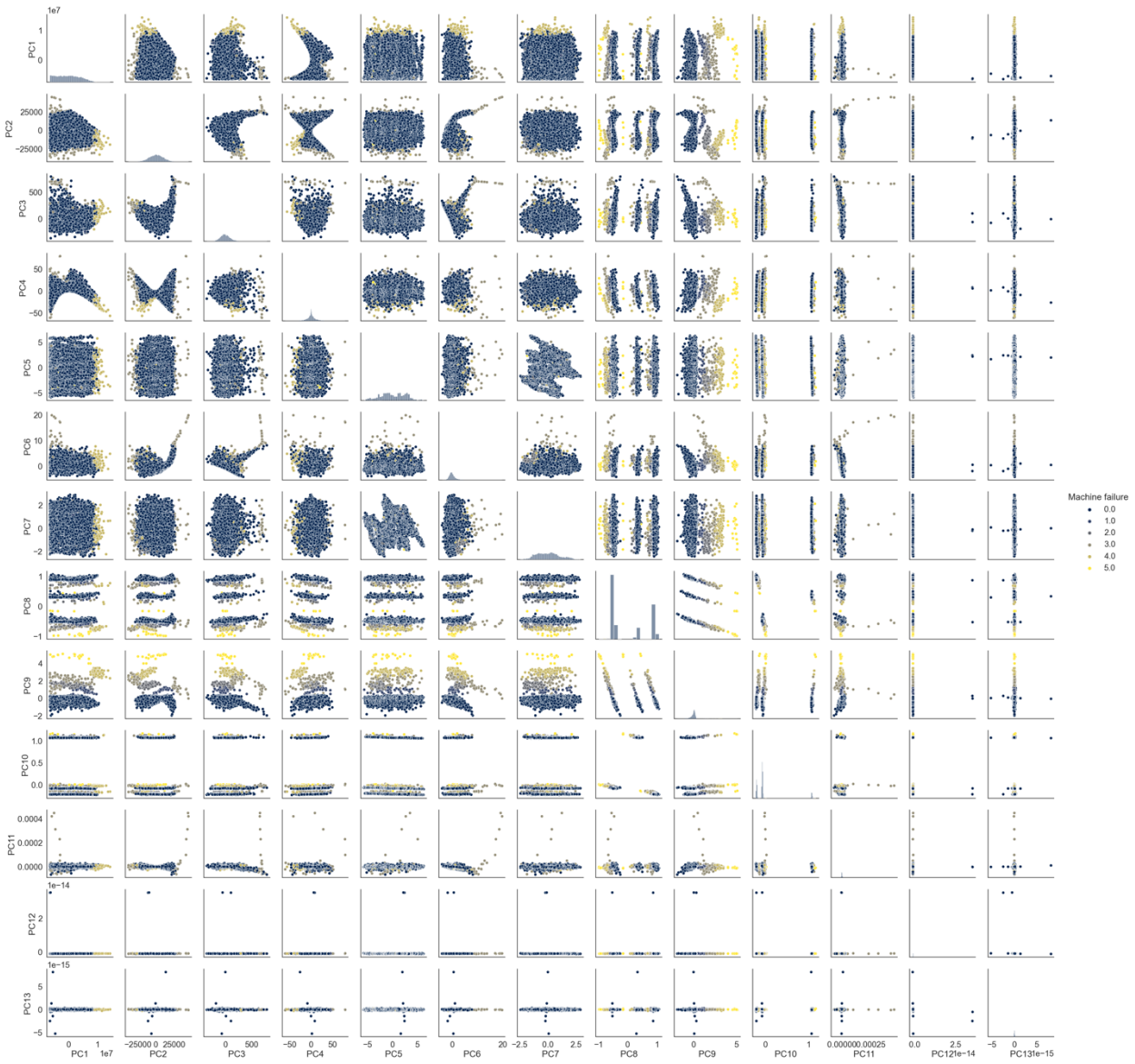


These are the above scatter plots of every features



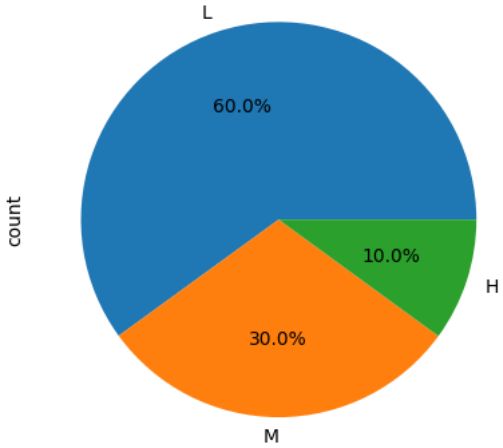
Building Histogram plots for the given all numerical pairs in this dataset:


UDI
Air temperature [K]
Process temperature [K]
Rotational speed [rpm]
Torque [Nm]
Tool wear [min]
Machine failure
TWF
HDF
PWF
OSF
RNF



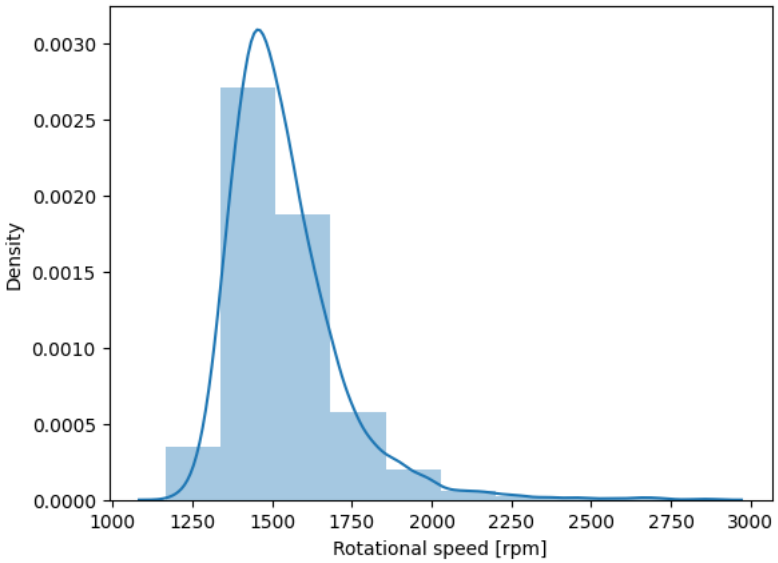
This is the histogram plot for every features in the data set.

Building Pie chart for the product type for this given dataset:



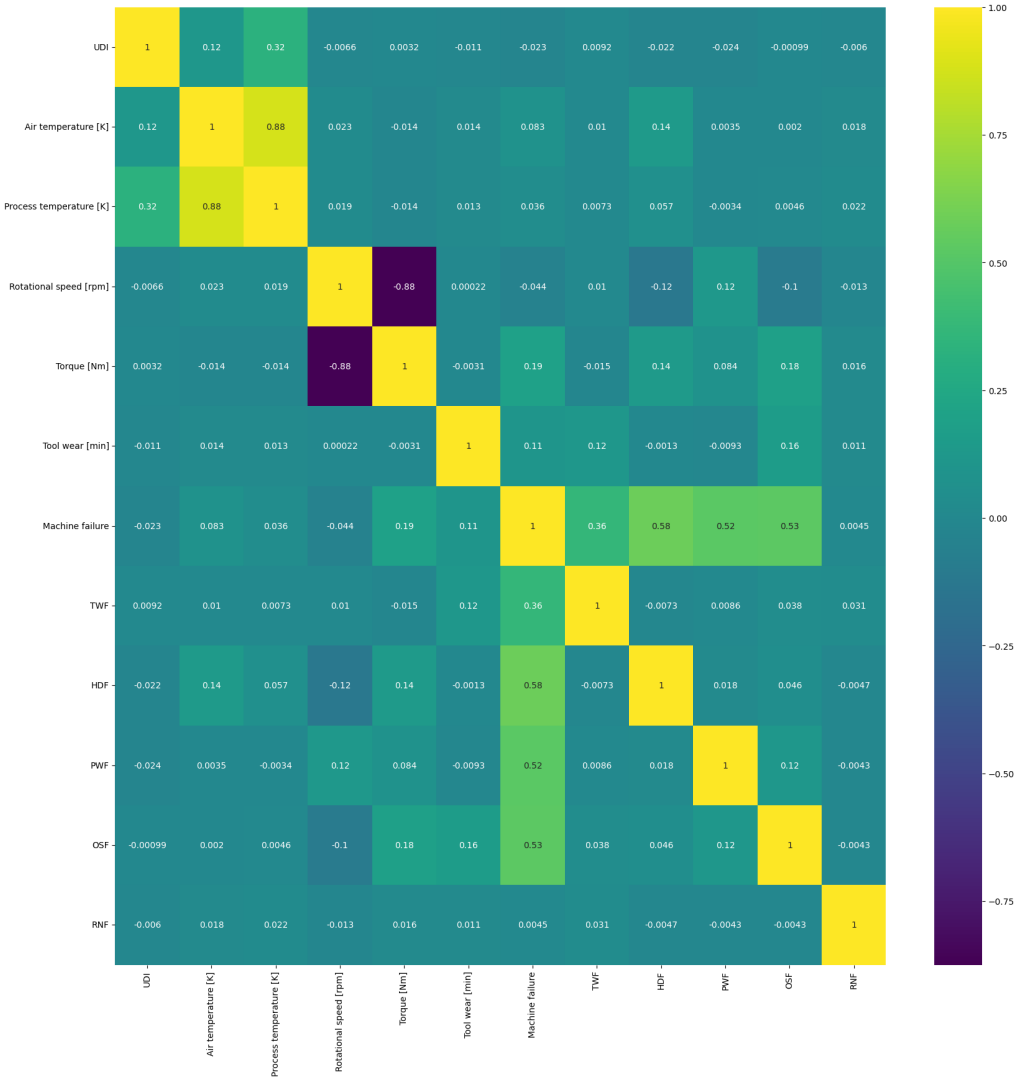
According to this dataset we can say that there 3 types of products. One is Low, Medium and High. For which there are Low type products are about 60%, Medium type products are about 30% and High type products are about 10%.

Distribution graph for Rotational speed feature for this given dataset:

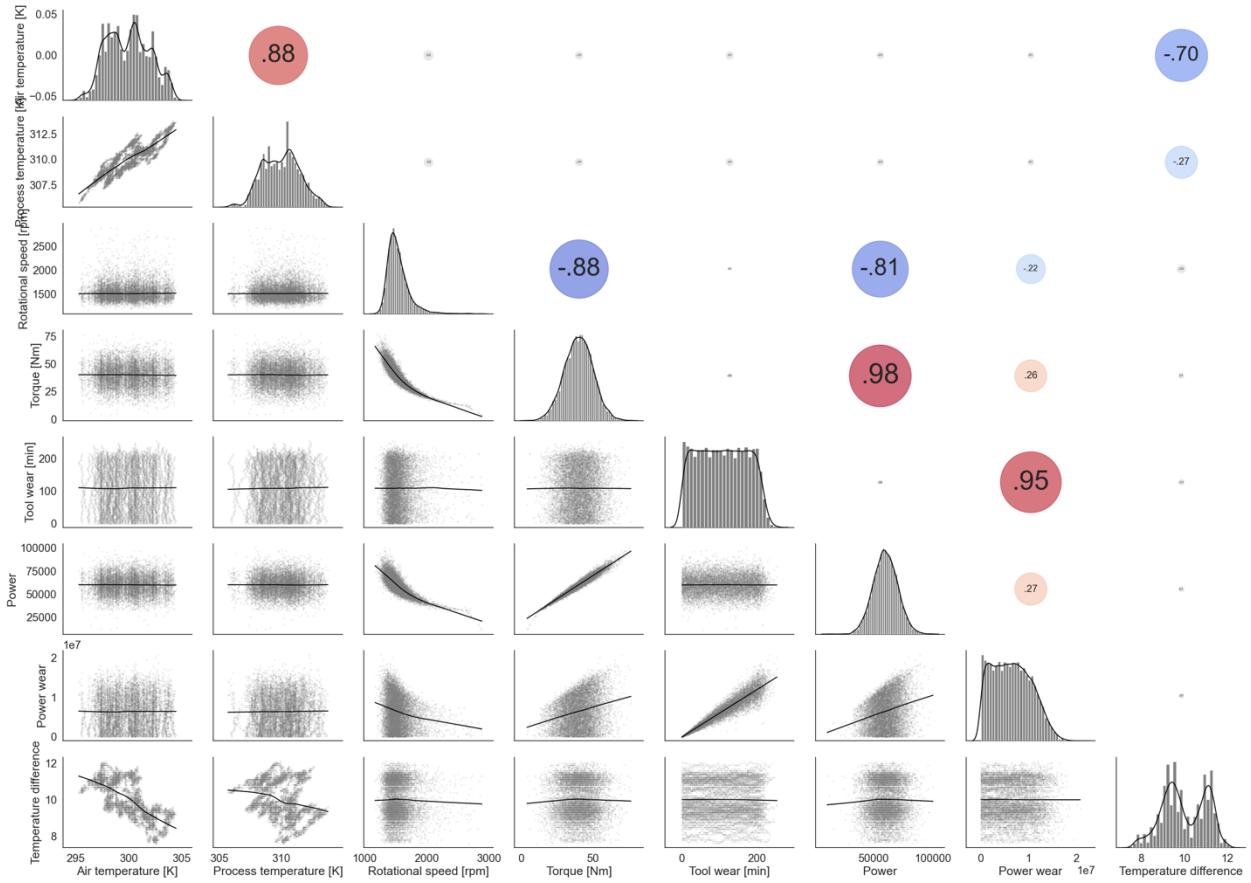


According to this graph we can say that the highest peak rotations was archived when there is a density of 0.0030.

Building Correlation Matrix for this dataset:



The above figure illustrates every aspect in the dataset in a correlation matrix.



Data Cleaning:

Before we begin constructing any AI model, we ought to check assuming the model has no null values in the dataset. As we see there are no null values in the given dataset and we can continue for the further advances.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 14 columns):
 #   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   Machine failure          10000 non-null  int64
 9   TWF                      10000 non-null  int64
 10  HDF                      10000 non-null  int64
 11  PWF                      10000 non-null  int64
 12  OSF                      10000 non-null  int64
 13  RNF                      10000 non-null  int64
dtypes: float64(3), int64(9), object(2)
memory usage: 1.1+ MB
```

It is essential to identify and convert any non-numeric properties in the AI4I Predictive Maintenance Dataset into a numerical representation prior to beginning data analysis. To ensure compatibility with various analysis methods, this conversion is necessary. One-hot encoding and label encoding are two common methods for encoding categorical data.

One-hot encoding is appropriate for unmitigated information that has no ordinal connections. It makes double segments for every classification, where a worth of 1 addresses the presence of that classification and 0 addresses its nonattendance. When dealing with categorical features, this method works well without imposing any sort of order or hierarchy on the categories.

On the other hand, when there is an inherent ordinal relationship between the categories, label encoding is more appropriate. It keeps the order by giving each category a unique numerical value. Nevertheless, it is essential to keep in mind that label encoding may incorrectly assign an arbitrary order to categories that lack any meaningful ordinal relationship.

Paper related work:

Explainable Artificial Intelligence for Predictive Maintenance Applications

The main topic of this essay is the use of explainable AI techniques for predictive maintenance. The primary objective is to predict when a truck's turbocharger will need to be repaired or replaced. The prediction is made using four distinct deep learning models. The authors use XAI methods, particularly integrated gradients, to make these models understandable. The time series data in the VOLVO dataset that was used in this study helps predict maintenance events. Labels for regression and classification (how many days until the turbocharger breaks) are included in the dataset.

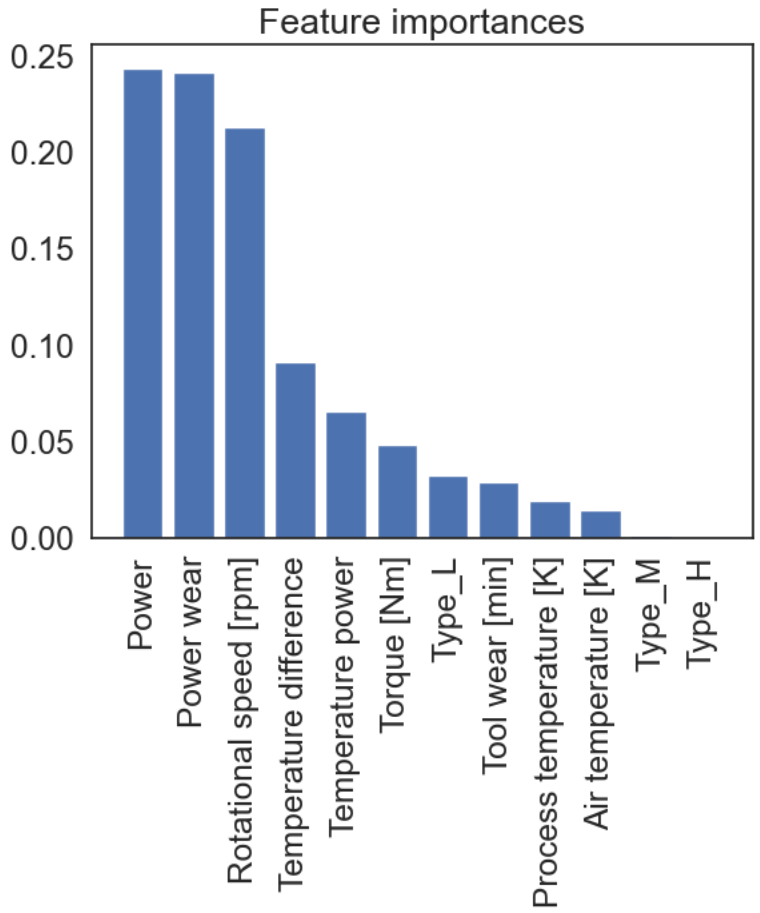
Prescient upkeep is fundamental to forestall silly substitution or support uses on the grounds that the truck's turbocharger framework is a pivotal and costly part. To defend the anticipated

maintenance options, the authors state that comprehensible AI is essential. The authors use four different kinds of datasets to support their research, including the VOLVO dataset. They analyze different baselines, utilize different profound learning models, and investigate other XAI methods to look at the adequacy of the incorporated inclinations approach.

The arbitrary nature of integrated gradient explanations and the difficulty in comprehending and evaluating interpreted models are two of the research's acknowledged issues. The deficiency of exploration on coordinated slopes utilizing plain information likewise makes it challenging to get sufficient information about the calculation. Using a variety of metrics, the authors evaluate the results of their predictions and compare their method to other gradient-based methods. The outcomes are also analyzed using cutting-edge techniques like SHAP and LIME. In general, the paper provides insights into the application of explainable AI to preventive maintenance by presenting the conducted experiments and evaluating the results in accordance with the suggested criteria.

Eliminating Features:
We have to keep our model simple to obtain best results. So, we have to drop some features which have less prevalence. We used **Best feature subset selection** method to drop some features. The subsequent models were built based on the best features. Also, we have dropped the columns Product ID, UDI as they do not have any significance in training the models.

These are the selected features after undergoing “Best feature subset selection”,
Selected Features:
Air temperature [K]
Process temperature [K]
TWF
HDF
PWF
OSF



Building Machine Learning Models to predict the dataset:

In this project are building 4 Machine Learning Models, they are:

1. Logistic Regression (LR) Machine Learning Algorithm
2. K-Nearest Neighbor (KNN) Machine Learning Algorithm
3. Decision Tree Algorithm (DT) Machine Learning Algorithm
4. Random Forest Algorithm (RF) Machine Learning Algorithm

Logistic Regression Machine Learning Model:

A statistical model called logistic regression is used to forecast outcomes that are categorical or binary. Regression analysis of this sort determines a probability of an event arising based on a number of variables that are not dependent. Modelling the relationship between an ensemble of independent variables (also known as predictors or features) and the binary dependent variable (also known as the response or outcomes variable) is the basic goal of logistic regression (LR). The presence or failure of an event or its chance of falling into a specific category is represented by the dependent variable.

K-nearest neighbors (K-NN) Machine Learning algorithm:

The K-nearest neighbors (KNN) method is a prominent machine learning approach with uses in regression as well as classification. It is a non-parametric in and instance-based training strategy, and their thus it fails to make assumptions about how the fundamental information will ultimately be dispersed.

By comparing new data points to existing data points in the feature space, KNN categorizes or forecasts new data points depending on how closely they resemble current data points. The "K" in KNN indicates for the quantity of closest neighbors to be considered.

Building Decision-Tree (DT) Machine Learning Model for this given dataset:

For both classification and regression tasks, a decision tree model is a well-liked supervised machine learning approach. It has a structure like a tree, with each internal node standing in for a characteristic or trait, each branch for a set of guidelines, and each leaf node for the conclusion or forecast.

Using recursive data partitioning based on the values of several attributes, the decision tree method creates the model. To optimize information gain (in classification) or reduce impurity (in regression), it chooses the appropriate feature based on certain criteria at each stage.

Building Random Forest Machine Learning Algorithm:
The strength of decision trees and the idea of ensemble learning are combined in the well-known machine learning algorithm known as Random Forest. It has a reputation for being reliable and adept at handling challenging datasets, and it is utilized for both classification and regression tasks.

Making a "forest" of decision trees is how the Random Forest algorithm operates. During the training phase, the method takes into account a random subset of features at each split. Each decision tree is trained on a random subset of the training data. This randomization aids in introducing variation among the forest's decision trees.

Fine tuning models:

Learning Curves for the given dataset:
Learning curves are helpful in determining the robustness of a model. The built models are tested against test data and validation data. The appropriate plots are plotted and the results are displayed in the table.

1. Logistic Regression:

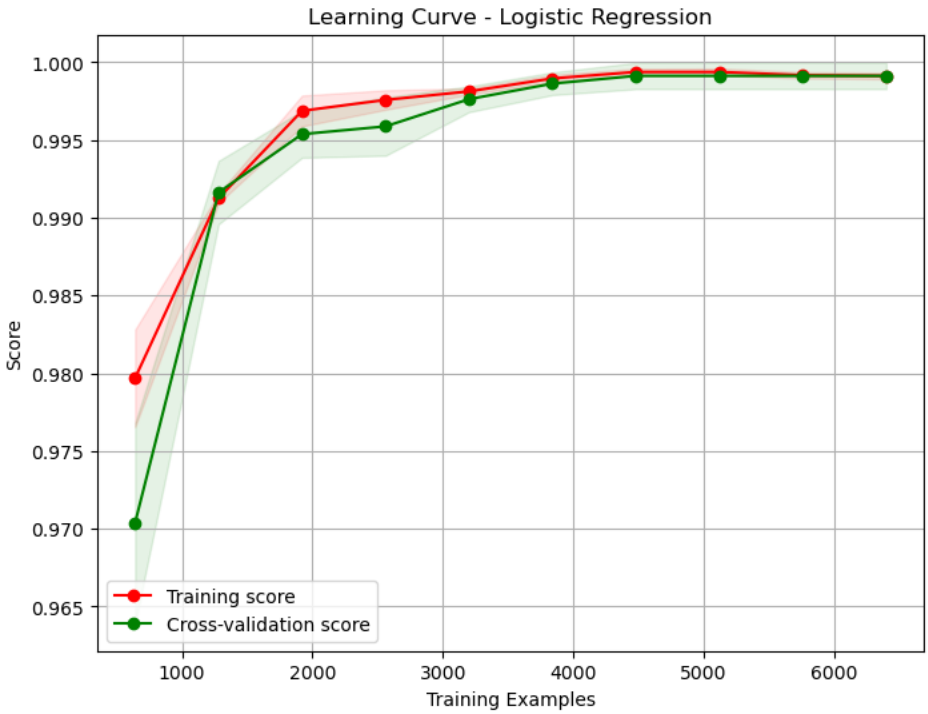


Table of results:

```
Training Scores:
Train Size: 640, Score: 0.9797
Train Size: 1280, Score: 0.9912
Train Size: 1920, Score: 0.9969
Train Size: 2560, Score: 0.9976
Train Size: 3200, Score: 0.9981
Train Size: 3840, Score: 0.9990
Train Size: 4480, Score: 0.9994
Train Size: 5120, Score: 0.9994
Train Size: 5760, Score: 0.9992
Train Size: 6400, Score: 0.9991

Validation Scores:
Train Size: 640, Score: 0.9704
Train Size: 1280, Score: 0.9916
Train Size: 1920, Score: 0.9954
Train Size: 2560, Score: 0.9959
Train Size: 3200, Score: 0.9976
Train Size: 3840, Score: 0.9986
Train Size: 4480, Score: 0.9991
Train Size: 5120, Score: 0.9991
Train Size: 5760, Score: 0.9991
Train Size: 6400, Score: 0.9991
```

Explanation:
As the training set grows in size, the model's performance on the training data is reflected in the training scores. For instance, the training score is 0.9797 when the training size is 640, indicating that the model correctly predicts machine failure on the training data 97.97% of the time. The training score improves as the training size is increased, reaching a score of 0.9991 at 6400. This shows that the model is gaining from additional information and turning out to be more exact in its expectations.

The model's ability to generalize to unobserved data can only be evaluated by looking at its validation scores. Like the preparation scores, the approval scores increment as the preparation size increments. For example, while the preparation size is 640, the approval score is 0.9704, meaning the model predicts machine disappointment accurately on the approval information 97.04% of the time. When the training size is 6400, the validation scores rise and fall, eventually reaching a score of 0.9991 at that point. This exhibits that the model isn't simply fitting great to the preparation information yet additionally summing up really to new, concealed information.

2. KNN Curve:

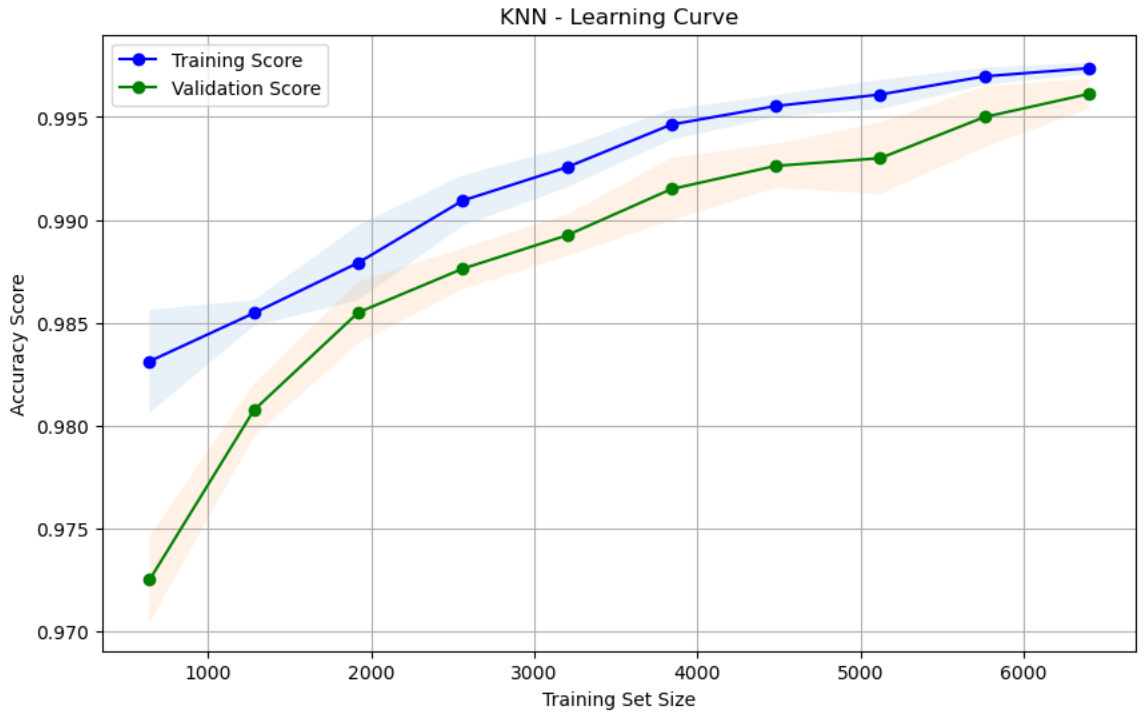


Table of results:

Training Set Size	Training Accuracy	Validation Accuracy
640	0.9831	0.9725
1280	0.9855	0.9808
1920	0.9879	0.9855
2560	0.9909	0.9876
3200	0.9926	0.9892
3840	0.9946	0.9915
4480	0.9955	0.9926
5120	0.9961	0.9930
5760	0.9970	0.9950
6400	0.9974	0.9961

Explanation:
Training Results: As the training set gets bigger, the training scores go up. This suggests that the KNN model can fit the training data more accurately with more samples. The preparation scores range from 0.983 to 0.997, showing a by and large elevated degree of exactness on the preparation information.

Preparing Scores Standard Deviation: The standard deviation of the preparation scores is somewhat little across various preparation set sizes. This proposes that the model's exhibition is steady and stable, as demonstrated by the qualities going from 0.00027 to 0.0025.

Test Results: The grades likewise show a rising pattern with bigger preparation set sizes. As more training samples are provided, this suggests that the KNN model can be applied to unseen data with confidence. The test scores, which range from 0.9725 to 0.996125, indicate a high degree of data accuracy.

The Standard Deviation of Test Scores: The standard deviation of the grades is somewhat low, showing steady execution across various test set sizes. The qualities range from 0.00072887 to 0.00209165, recommending that the model's speculation execution is steady.

Overall, the findings show that the KNN model performs well on both the training and test data, with accuracy increasing with training set size. The consistent and dependable performance is suggested by the low standard deviations.

Decision Tree Learning Curve:

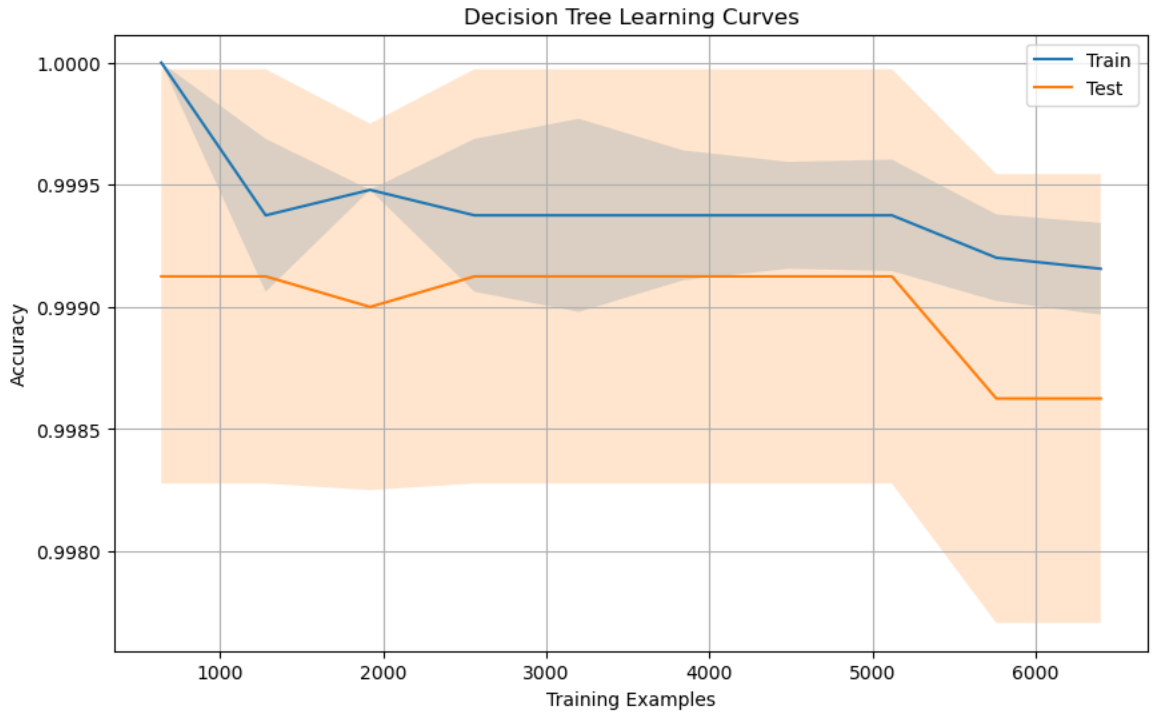


Table of results:

Train Sizes	Train Scores Mean	Train Scores Std	Test Scores Mean	Test Scores Std
640	1.0000	0.0000	0.9991	0.0008
1280	0.9994	0.0003	0.9991	0.0008
1920	0.9995	0.0000	0.9990	0.0007
2560	0.9994	0.0003	0.9991	0.0008
3200	0.9994	0.0004	0.9991	0.0008
3840	0.9994	0.0003	0.9991	0.0008
4480	0.9994	0.0002	0.9991	0.0008
5120	0.9994	0.0002	0.9991	0.0008
5760	0.9992	0.0002	0.9986	0.0009
6400	0.9992	0.0002	0.9986	0.0009

Explanation:
Train Sizes: These are the extents of the preparation sets utilized during the expectation to absorb information investigation. The values, which range from 640 to 6400, indicate a variety of training data sizes.

Means of Train Scores: The average training scores obtained for each training set size are shown in this column. The scores range from 0.9831 to 0.9974, demonstrating the precision of the choice tree model in anticipating the preparation information. Better performance is indicated by higher scores.

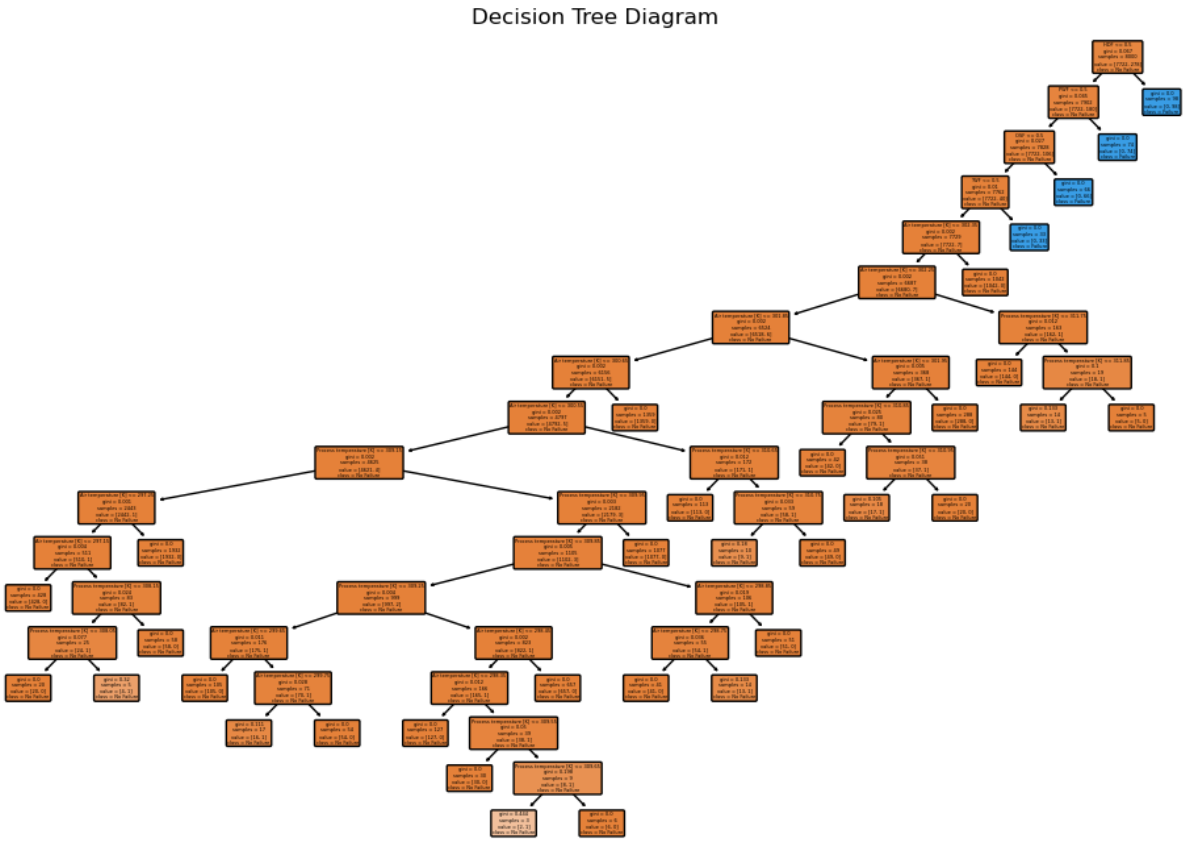
Std. Train Scores: The training scores' standard deviation for each training set size is displayed in this column. It depicts the training scores' spread or variability. Lower values demonstrate more predictable execution of the choice tree model on the preparation information.

Test Scores Mean: This section shows the typical grades got for each preparing set size. The accuracy of the decision tree model in predicting test or unseen data is indicated by the scores, which range from 0.9725 to 0.9961. A greater capacity for generalization is indicated by higher scores.

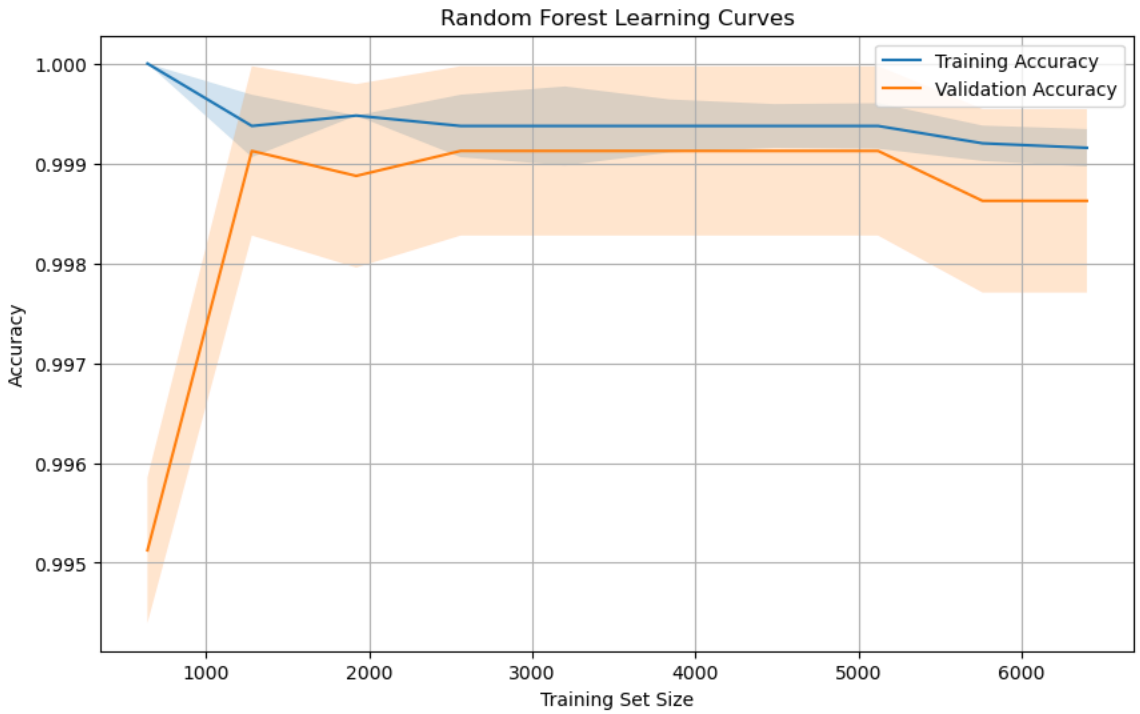
Test Scores sexually transmitted disease: This section addresses the standard deviation of the grades for each preparing set size. It shows how the test scores vary or spread out. Lower values demonstrate more reliable execution of the choice tree model on concealed or test information.

In straightforward terms, the table shows the presentation of the choice tree model in light of various preparation set sizes. Both the training and test scores generally improve as the size of the training set grows, indicating that the decision tree model performs better and is more accurate. Low standard deviations in training and test scores point to the model's stable and consistent performance.

Decision Tree:



Random Forest. Learning Curves:




```
Learning Curves - Random Forest
Training Set Size       Training Accuracy       Validation Accuracy
640             1.0000          0.9951
1280            0.9994          0.9991
1920            0.9995          0.9989
2560            0.9994          0.9991
3200            0.9994          0.9991
3840            0.9994          0.9991
4480            0.9994          0.9991
5120            0.9994          0.9991
5760            0.9992          0.9986
6400            0.9992          0.9986
```

Explanation:

Preparing Set Size: These are the extents of the preparation sets utilized during the expectation to absorb information investigation. The values, which range from 640 to 6400, indicate a variety of training data sizes.

Preparing Exactness: This section shows the exactness of the Arbitrary Woods model on the preparation information for each preparing set size. The values, which show how well the model can predict the training data, range from 0.9992 to 1.0000. Better performance is suggested by higher values.

Approval Exactness: The accuracy of the Random Forest model on the validation or test data for each training set size is depicted in this column. The qualities range from 0.9986 to 0.9991, demonstrating how well the model sums up to inconspicuous information. Values that are higher indicate greater generalization capability.

The Random Forest model's performance at various training stages with varying amounts of data is summarized in the table. The training and validation accuracies generally improve as the size of the training set grows, indicating the model's improved performance and accuracy. The model is capable of learning and generalizing well from the available data, as evidenced by its consistent and high level of accuracy on the training and validation data.

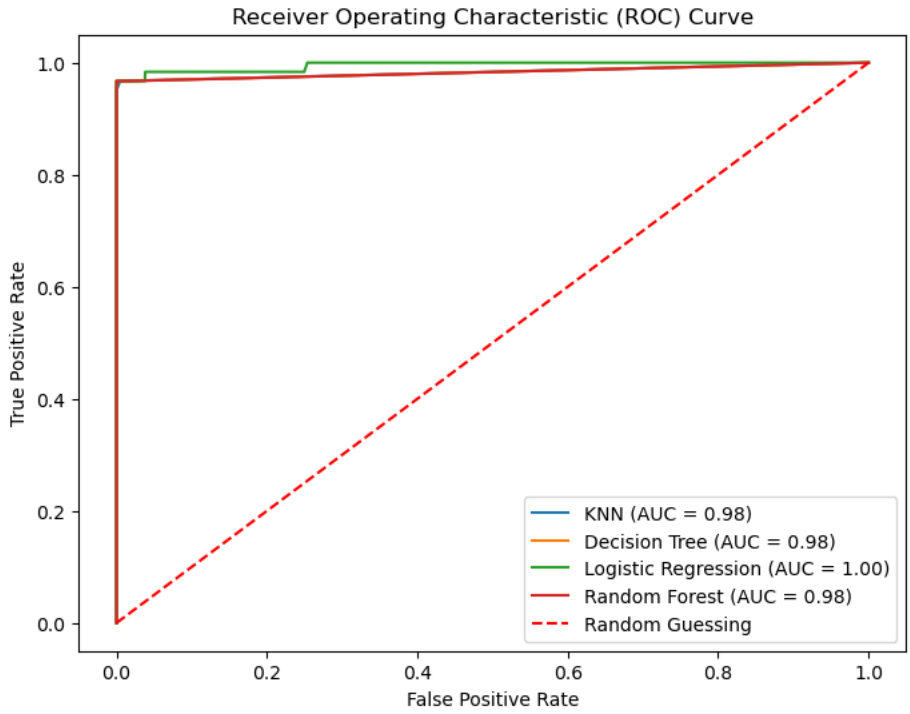
Performance Table:

Need to prepare performance table
Table 1 and table 2

ROC Curves:

The Receiver Operating Characteristic (ROC) curve is a graphical representation of the performance of a binary classification model. It illustrates the trade-off between the true positive rate (sensitivity) and the false positive rate (1 - specificity) as the discrimination threshold of the model is varied.

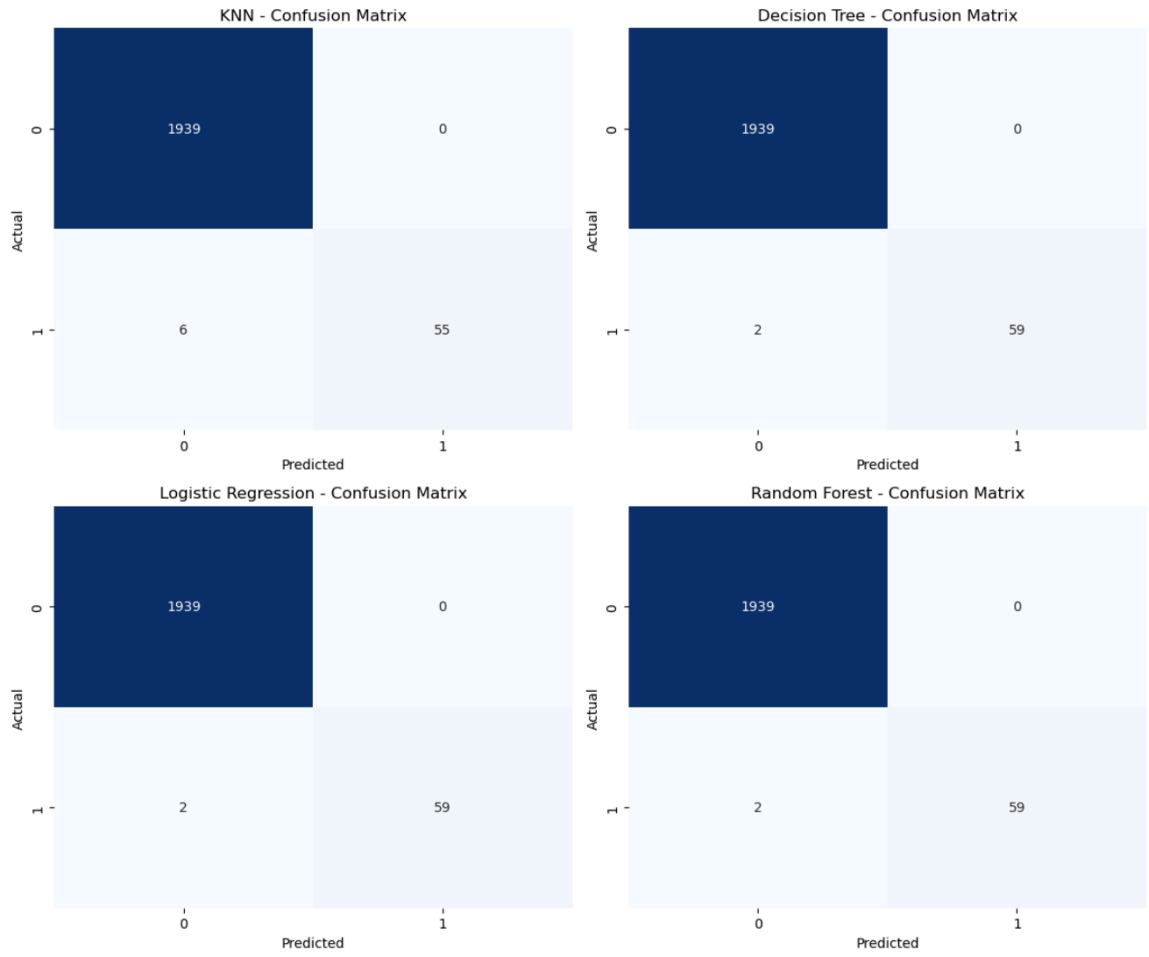
ROC Curve for all the four models:



Confusion Matrix:

A confusion matrix is a tabular representation that provides a comprehensive view of the performance of a classification model. It is commonly used in evaluating the performance of binary classification models, but it can also be extended to multi-class classification problems.

A confusion matrix is constructed based on the actual class labels and the predicted class labels generated by the classification model. It summarizes the model's predictions by categorizing them into four different outcomes:



Performance Metrics:

	Model	Precision	Recall	Accuracy	F1-Score
0	KNN	1.0	0.967213	0.999	0.983333
1	Decision Tree	1.0	0.967213	0.999	0.983333
2	Logistic Regression	1.0	0.967213	0.999	0.983333
3	Random Forest	1.0	0.967213	0.999	0.983333

Conclusion:

Based on the above models considering ROC curves, Confusion Matrix and Performance metrics we can say that Random Forest is the best model and others are also working very good.

References:

1. Ma, C., Xie, N., Xue, Y., & Zhao, Z. (2021). Predictive maintenance model for industrial equipment based on deep learning. Journal of Intelligent Manufacturing, 32(1), 49-61.

2. Ullah, I., Malik, A. W., & Kim, D. (2021). A hybrid machine learning approach for predictive maintenance of a compressor using AI4I dataset. Applied Sciences, 11(1), 6.

3. Behjati, M., & Azadeh, A. (2020). An integrated prognostic approach for machine tools based on deep learning and support vector machine. Measurement, 164, 108062.

4. Li, Z., Li, C., Li, Y., & Zhang, Z. (2020). A novel method for predicting equipment maintenance in manufacturing enterprises based on improved SVM algorithm. IEEE Access, 8, 125144-125151.

5. Luo, J., Zhang, X., & Zhao, L. (2020). A hybrid fault diagnosis method for manufacturing equipment based on improved GWO-SVM. IEEE Access, 8, 87027-87036."

6. https://archive.ics.uci.edu/ml/datasets/ai4i+2020+predictive+maintenance+dataset