A project report on

Predictive Maintenance

Submitted by

Y.V.Sai Yaswanth Reddy (12307855)

I.Maria Tejaswai (12309910)

P.Lavanya (12300764)

Manan Awasthi (12306419)

P.Balaji (12308401)

Students of

B.TECH

COMPUTER SCIENCE AND ENGINEERING

LOVELY PROFESSIONAL UNIVERSITY

2023-2027

Submitted to

Prof. Mahipal Singh Papola

1.ABSTRACT

This project focuses on predicting system failures in machines using machine learning. The dataset contains sensor data such as air temperature, process temperature, torque, toolwear, and rotational speed when it reaches their threshold they may occur the system failure, which in turn results in system failure. The following project aims to taking data from a Kaggle and used it to train classification models that will be able to predict future system failure. The project uses the concepts of Data Preprocessing, Classification and Ensemble Learning.

2.INTRODUCTION

To avoid unexpected breakdowns and costly downtimes, this project aims to accurately predict whether a machine is going to fail soon, based on real-time sensor and operational data. In industries such as manufacturing, aerospace, energy, and logistics, equipment failure not only leads to financial losses but also compromises safety and productivity. Therefore, predictive maintenance has become a key strategy in industrial operations, allowing companies to detect early signs of failure and take preventive action before a breakdown occurs.

As machinery becomes more advanced and integrated with digital monitoring systems, the volume and complexity of operational data increase rapidly. Managing and analyzing this data manually becomes difficult, especially in large organizations with hundreds of machines operating continuously. As a result, automated failure prediction using machine learning has emerged as a reliable solution to handle such challenges effectively.

System failure prediction falls under predictive analytics, where the goal is to assess the likelihood of failure using historical and real-time data. In earlier approaches, traditional statistical models were used for failure detection, but they often struggles with complex patterns imbalanced data, and scalability. To address these limitations, data mining and machine learning techniques have become increasingly popular, offering more accurate, scalable, and adaptable solutions.

This project leverages advanced machine learning models, including Decision Tree, Random Forest and XG BOOST, To identify patterns in machine behaviour and predict failures. Feature engineering is performed to extract meaningful insights from raw data. This system provides an efficient and practical tool for industries to minimize maintenance costs, improve safety, and enhance operational continuity.

3.DATA MINING TASK IDENTIFICATION

Data mining refers to the process of discovering hidden patterns, trends, and useful information from large datasets. It involves the application of various analytical methods to extract meaningful knowledge that can support informed decision-making. Based on the types of insights that can be obtained, data mining tasks can be categorized into regression, classification, clustering, data visualization and more.

The dataset in this project contains approximately 10% of the Failure detection. The failure detection labelled as ‘1’ and no failure detected labelled as ‘0’ in the dataset. So the task of the project is to predict whether a system is going to detect failure. Since, for the output there are only two classes, i.e, failure or no failure, therefore the problem falls under binary classification problem. To classify the data, various machine learning algorithms like Decision Tree, Random Forest and XG BOOST are applied in the project.

4. METHODS/TECHNIQUES APPLIED AND THEIR BRIEF DESCRIPTION

We used following classification models:

1.Decision tree

2.Random Forest

3.XG-Boost

4.1. Decision tree:

A Decision Tree is a supervised learning algorithm used mainly for classification tasks. It works by splitting the dataset into smaller subsets based on feature values, and builds a tree-like model of decisions. Each internal node of the tree represents a decision based on a feature, while each leaf node represents an outcome or class label.

Advantages of Using Decision Tree:

- Easy to Understand and Interpret: The tree structure makes it simple to visualize how decisions are made.
- No need for feature scaling: The model works well with both unscaled and scaled data.
- Handles both Numerical and Categorical data, in which suitable for read world sensor data.
- The tree automatically performs feature selection and focuses on the most important features during training.

4.2. Random Forest

Random Forest is an ensemble machine learning algorithm that builds multiple decision trees and combines their results to make more accurate and stable predictions. It is a supervised learning algorithm and is widely used for both classification and regression tasks.

The model works by creating several decision trees using different random subsets of the data and features. During prediction, each tree gives an output, and the final prediction is made by taking the majority vote from all trees

Advantages of Using Random Forest:

- High Accuracy: Combining multiple trees improves predictive performance compared to a single decision tree.
- Reduces Overfitting: The randomness in data sampling and feature selection makes the model more generalized.
- Handles Missing and Noisy Data Well: It's robust and performs well even with imperfect data.
- Works for Both Classification and Regression: Very flexible and reliable in different types of problems.
- No Need for Features Scaling: Can work well with raw or scaled features.

4.3. XG-BOOST(Extreme Gradient Boosting)

XGBoost is a powerful and efficient machine learning algorithm that belongs to the boosting family of ensemble techniques. Boosting is a method that builds multiple weak learners and combines them sequentially to form a strong learner. The idea is to focus more on the instances where previous models performed poorly, improving accuracy over time.

XGBoost addresses the common bias-variance trade-off in machine learning models. While a high bias leads to underfitting and high variance leads to overfitting, XGBoost uses various strategies to maintain a good balance between them, resulting in better generalization to unseen data.

Advantages of Using XGBoost:

- **Regularization:**
  Unlike traditional Gradient Boosting Machines, XGBoost includes L1 and L2 regularization, which helps control model complexity and reduces overfitting. This is why XGBoost is often referred to as a "regularized boosting" algorithm.
- **Parallel Preprocessing:**
  XGBoost is fast because it supports parallel computation during training, which significantly speeds up the learning process.
- **High Flexibility**
  It allows users to define custom objective functions and evaluation metrics, offering a high degree of flexibility for different types of problems.
- **Advanced Tree Pruning**
  In contrast to GBM which stops growing trees early if there is no improvement, XGBoost grows trees to a specified maximum depth and then prunes backward to improve performance. This results in more optimized and generalizable trees.

5. DATASET DESCRIPTION

The dataset contains 10000 observations and 10 features. Each observation represents a system maintenance and all the features together contributes about the system maintenance.

5.1 FEATURES DESCRIPTION:

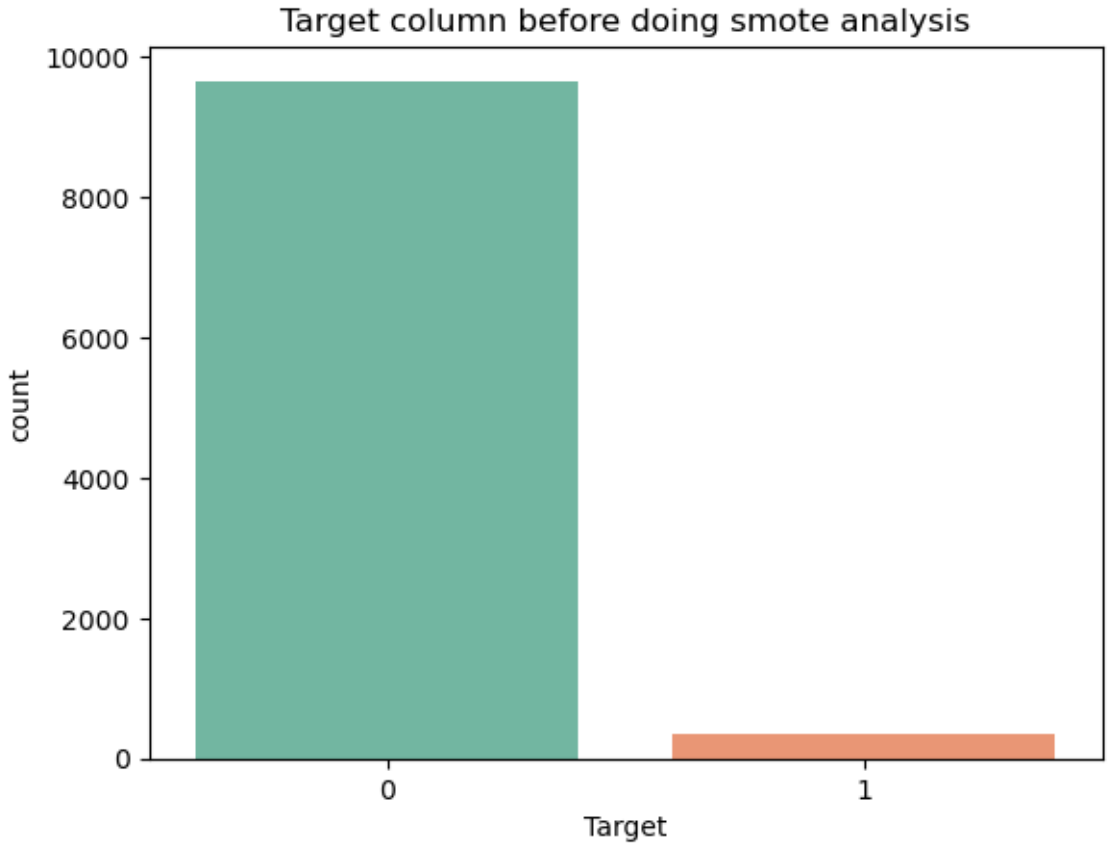
Type	It shows the type of machine whether it is low, medium or high
Air Temperature	It is sensor in which it shows how much air temperature is required
Process Temperature	It is sensor in which it shows the temperature for being processed
Rotational Speed	Speed of the machine
Torque	The Rotational force applied to the machine's components
Tool wear	It tracks the duration of usage of the tool under operational stress
Tor*Toolwear	The cumulative mechanical stress or load the tool has experienced over time
Temp_diff	How much hotter the machine is compared to the surrounding air
power	How much work the machine is doing at any moment
Target	It shows whether the machine is failure or running well

So actually, in raw dataset there is one categorical useful column, which is a type column and using label encoding I changed into numerical.

5.2. EXPLORATORY DATA ANALYSIS

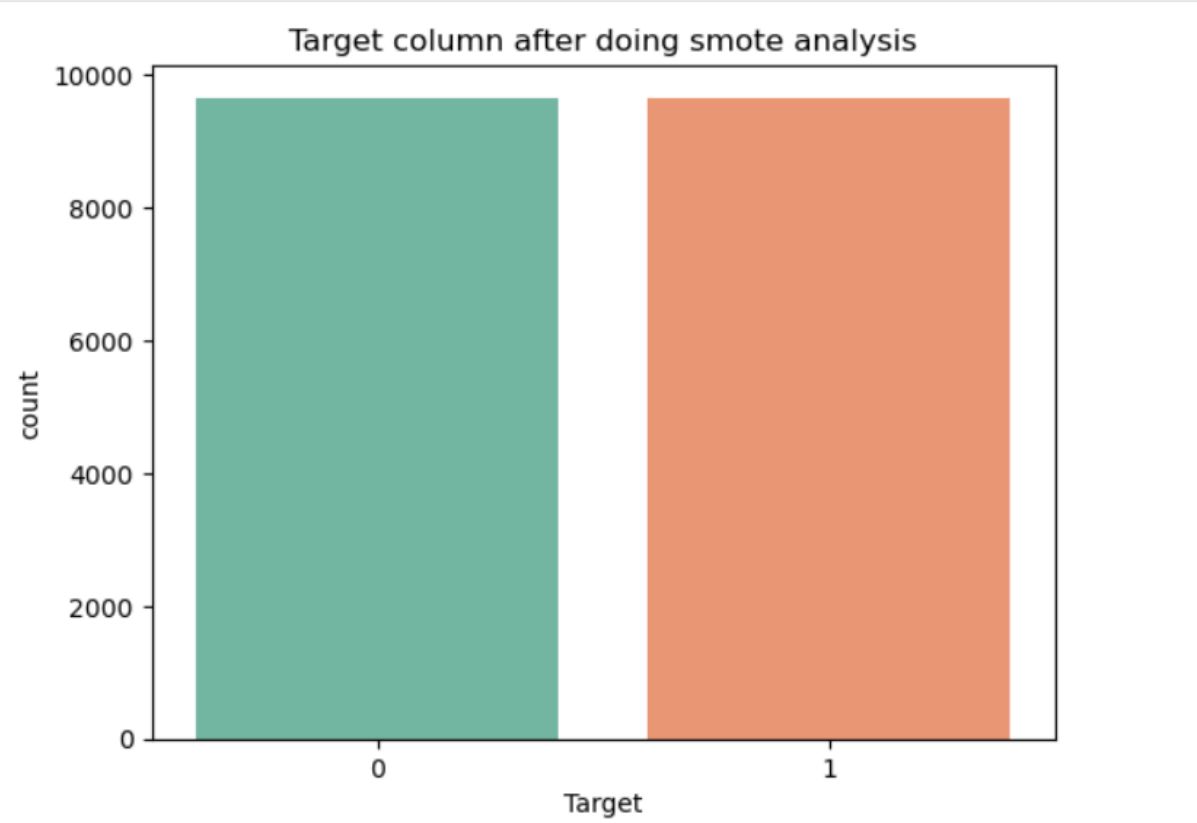
In statistics, exploratory data analysis (EDA) is an approach to analysing data set to summarize their main characteristics, often with visual methods.

- Frequency of target data

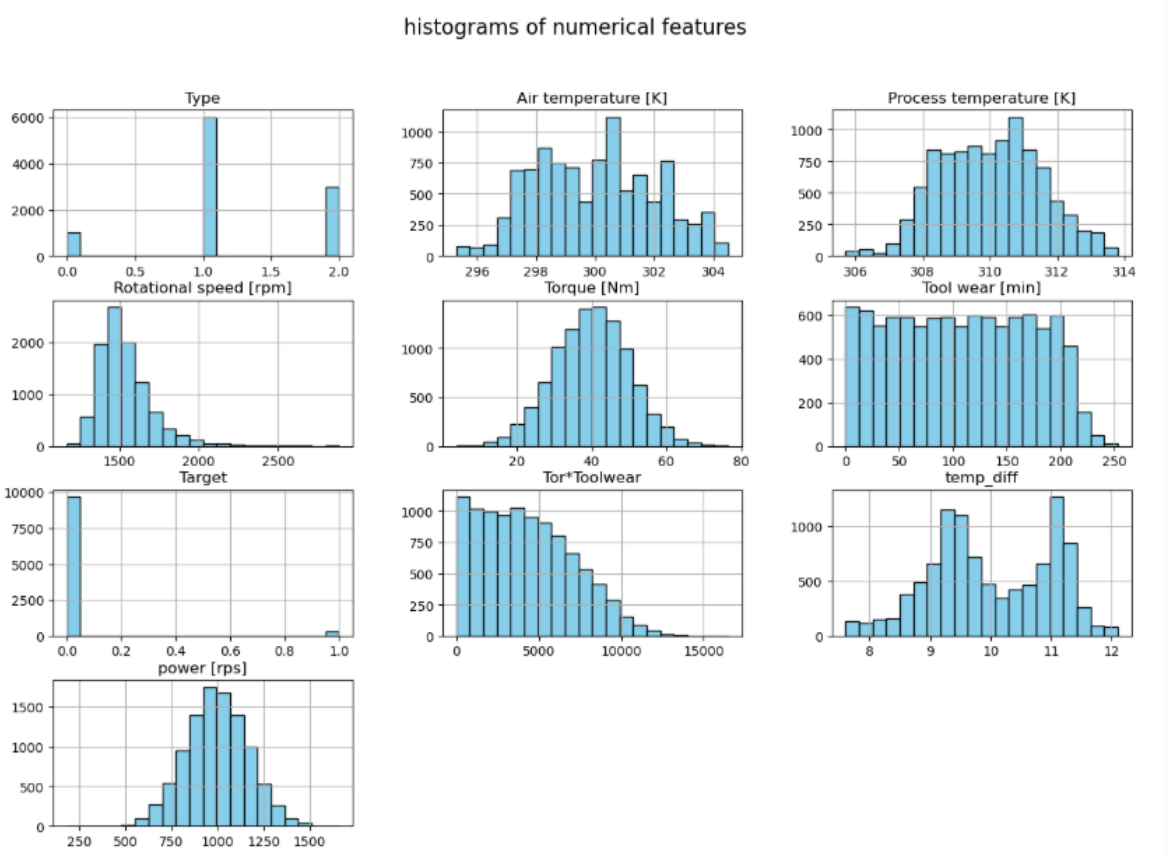


By the above count plot chart we say that the no failure is in 90% of the times and 10% of the times it is in failure and the target column is not in balances. So, to balance the target column, used SMOTE analysis.

- After doing smote analysis some synthetic data was added to ones as before the data is fully biased towards the zero.



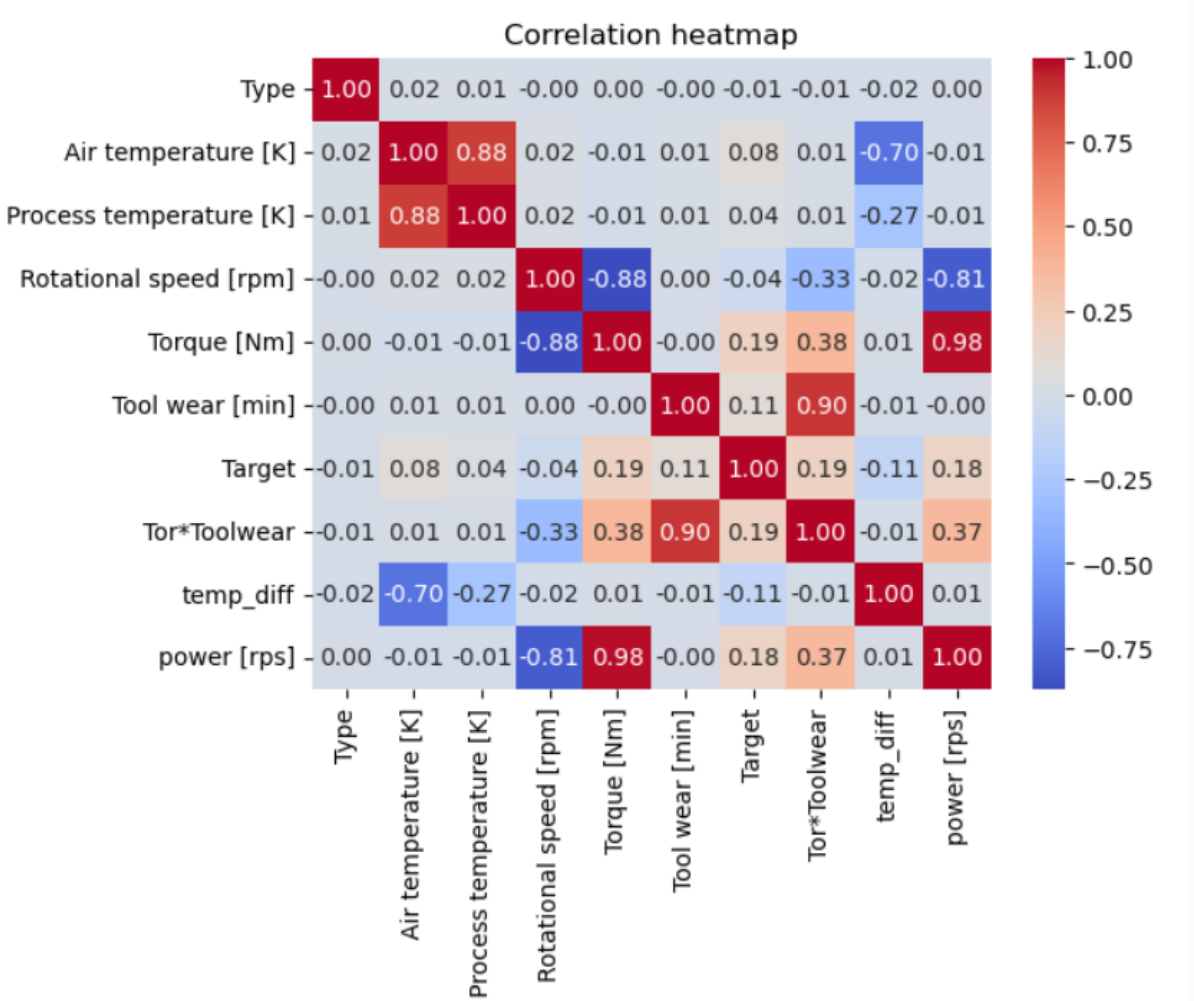
- Histogram for all numerical data after converting all categorical type data into numerical



Histograms are used to visualize the distribution of numerical data by dividing the values into intervals (bins). Each bar in the histogram represents the frequency of data points falling into that range. By analysing the histograms of the features in the dataset, we can gain insights into the nature of the data, detect skewness, and potentially identify outliers.

5.3. CORRELATION MATRIX

This correlation matrix measures the linear correlation between two variables. The resulting value is in between [-1,1] in which -1 means perfect negative correlation and +1 means perfect positive correlation and 0 represents no correlation in which the 0 correlation doesn't affect the output variable.



6. DATA PREPROCESSING

Data Preprocessing is a data mining technique that involves transforming raw data into an understandable format. These transformations are required because the real-world data is generally incomplete like lacking attribute values, lacking certain attributes of interest, or containing only aggregate data. Noisy like containing outliers. Therefore, Data preprocessing is a proven method of resolving such issues.

6.1. DATA MANIPULATION

1. A variable named UDI, Product ID and Failure Type was simply dropped because all the features inside it were unique and had no significance on the output.

6.2. FEATURE SCALING

Feature scaling is the method to limit the range of variables so that they can be compared on common grounds. It is performed on continuous variables.

It can vary the results a lot while using certain algorithms and have a minimal or no effect in others. Most of the times the dataset contains values with high magnitudes, units and range. If left alone, these algorithms only take in the magnitude and neglect the units. For example, there is a difference between units like 300 kelvin and 27 degrees Celsius so it takes only the value and neglect the unit which may cause the problem.

So, to handle this problem, we use feature scaling techniques, to bring all the features on to one scale.

There are Four Common methods to perform feature scaling

1. Standardization
2. Min-Max Scaling
3. Mean Normalization
4. Robust Scaling

- In my project I used Mean Normalization in which the values are in between [-1,1] with μ=0.

$$x' = \frac{x - \text{mean}(x)}{\max(x) - \min(x)}$$

6.3. LABEL ENCODING

Label encoding is a process by which categorical variables are converted into numerical form that could be provided to ML algorithms to do a better job in predicting the output. It is because some Machine Learning libraries won't accept the categorical data as input. Thus, we convert them into numerical variables. We use this label encoding when our selected categorical column contains 1 or 2 or 3 unique variables if more than these then we need to use one hot encoding.

In the project I had applied it on the Type column in which it has 3 unique variables as 'L', 'M' and 'H' represents low, medium and high are converted into 'L=1', 'M=2' and 'H=3'.

6.4. FEATURE SELECTION

It is the process of finding and selecting the most useful features in a dataset and it is an important step of the machine learning. Unnecessary features decrease the training speed and decrease the model performance on the test set.

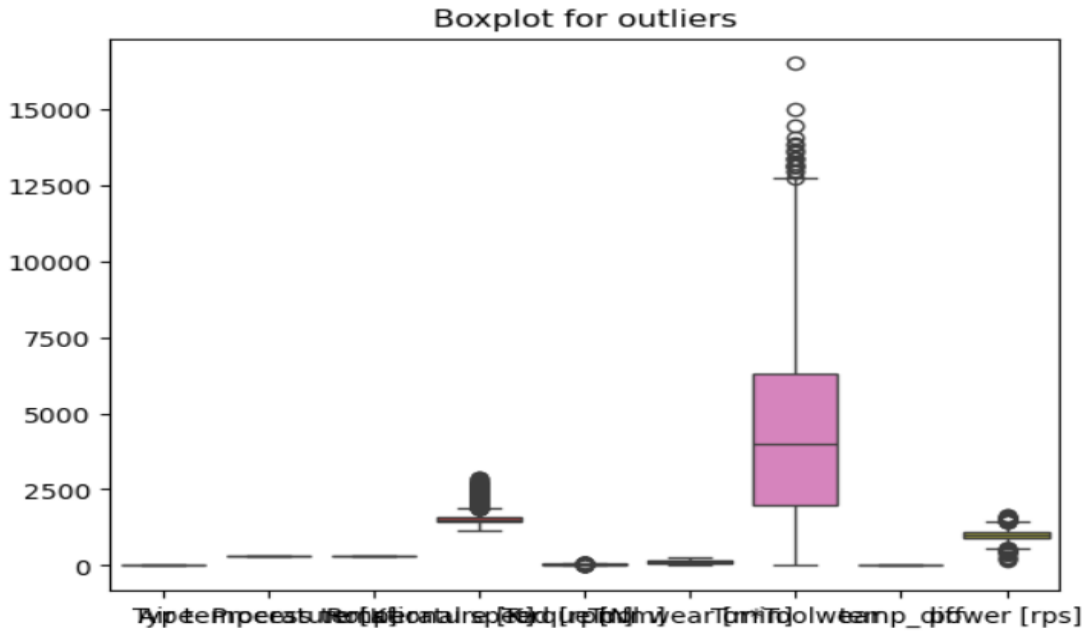
There are many benefits of performing feature selection before modelling, some of them are:

- Reduces Overfitting
- Improves Accuracy
- Reduces Training Time

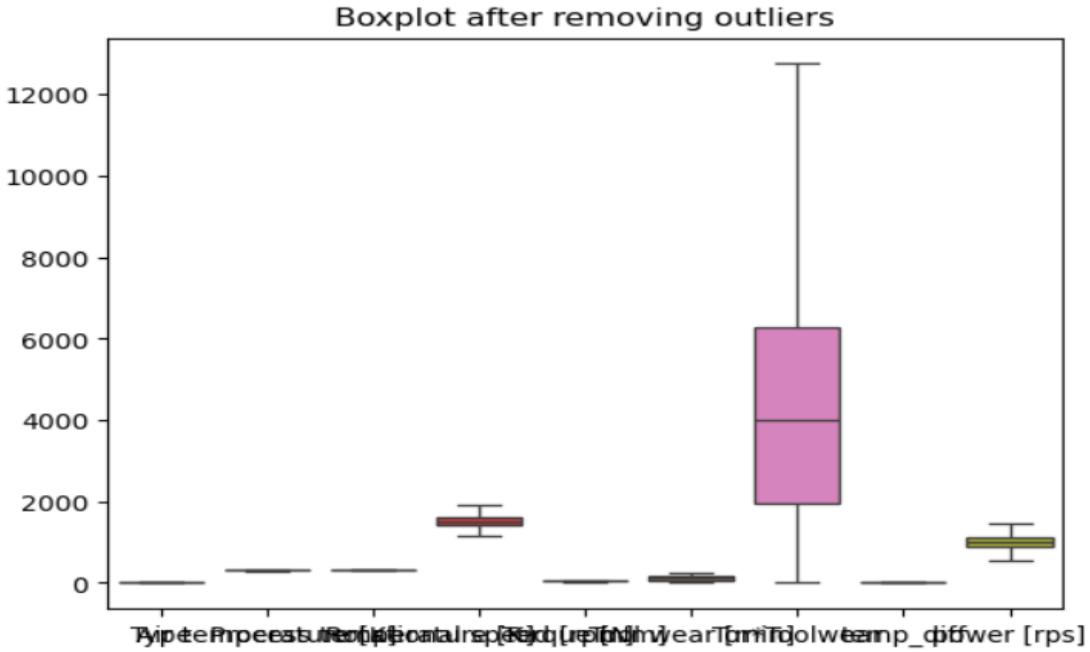
6.5. FEATURE ENGINEERING

- It is a process in which we add some additional data to the dataset so that additional data will shows more effect on test data and increase model performance.
- In project I added three additional columns those are power, tor*tool wear and temperature difference
- And there are some outliers in the dataset in which may cause the problem to the test set
- So, we Identified those outliers using Box plot and removed those outliers by using Inter Quantile Range (IQR)

- Box plot used to show the outliers in the dataset



- Box plot after applying IQR



7. MODEL EVALUATION

Predictive Modelling works on constructive feedback principle. Therefore, after building a model, it takes feedback from metrics, make improvements and continue until it achieves a desirable accuracy. Evaluation metrics explain the performance of a model. There are various different kinds of evaluation techniques to measure the performance of a model depending upon the type of model and implementation plan of model. The metric used in out project and their description are as follows.

7.1. Train/Test split

Splitting the dataset into two parts, so that the model can be trained and tested on different data. Better estimate of out-of-sample performance, but still a “high variance” estimate. Useful due to its speed, and flexibility. Data can be split into 80% of training and 20% of testing.

7.2. CONFUSION MATRIX

A confusion matrix is a n*n matrix, where n is the number of classes being predicted. For the problem in hand, we have n=2 because it is a binary classification and hence, we get a 2*2 matrix

	POSITIVE	NEGATIVE
POSITIVE	TRUE POSTIVE(TP)	FALSE POSITIVE(FP)
NEGATIVE	FALSE NEGATIVE(FN)	TRUE NEGATIVE(TN)

Some important evaluation measures for a confusion matrix are as follows:

ACCURACY:
The proportion of the total number of predictions that were correct.

$$\text{Accuracy} = (TP + FP)/ (TP + FP + TN + FN)$$

Precision:

The proportion of positive cases that were correctly identified.

$$\text{Precision} = (TP)/(TP+FP)$$

Recall:

The proportion of actual positive cases which are correctly identified.

Sensitivity=(TP)/(TP+FN)

Specificity

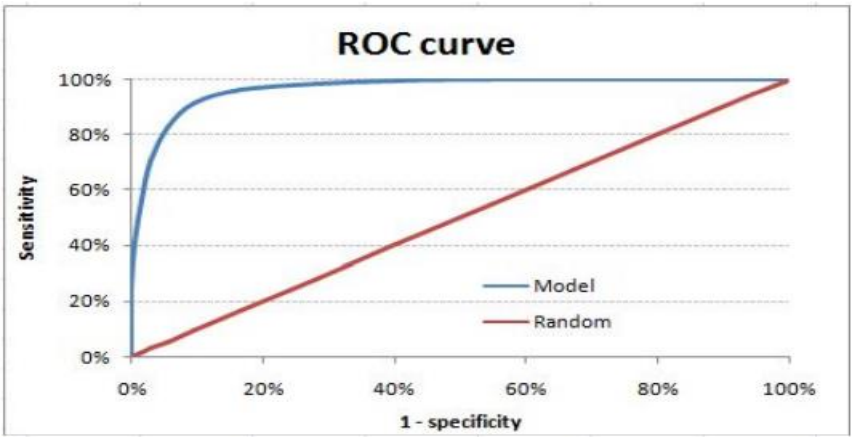
The proportion of actual negative cases which are correctly identified.

Specificity=(TN)/(TN+FP)

7.3. AREA UNDER THE ROC CURVE(AUC-ROC)

If we look at the confusion matrix above, we observe that for a probabilistic model, we get different value for each metric. Hence, for each sensitivity, we get a different specificity.

The ROC curve is the plot between sensitivity and (1- specificity). (1- specificity) is also known as false positive rate and sensitivity is also known as True Positive rate.



To bring this curve down to a single number, we find the area under this curve (AUC). Note that the area of entire square is always 1.

Following are the few thumb rules:

- 0.90-1 = excellent
- 0.80-0.90 = good
- 0.70-0.80 = fair
- 0.60-0.70 = poor
- 0.50-0.6 = fail

8. RESULTS AND DISCUSSION

8.1 Comparison of various models:

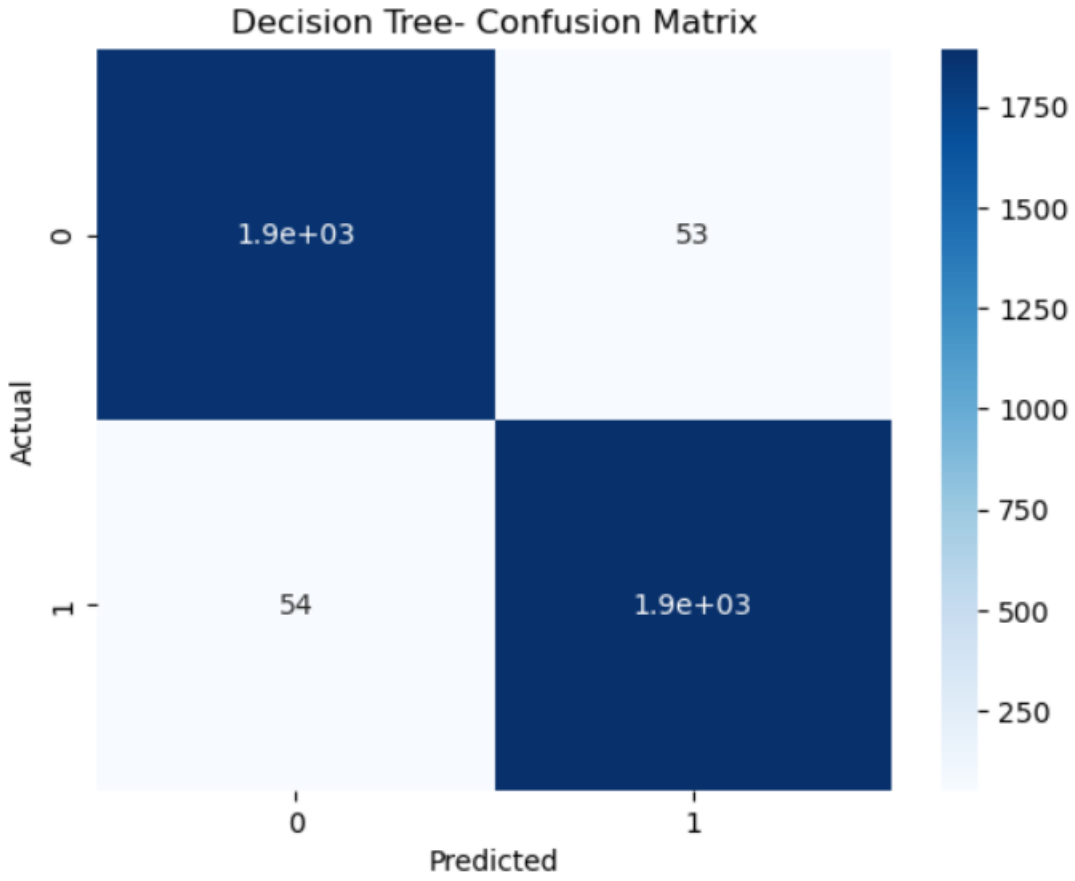
MODEL	Decision Tree	Random Forest	XGBoost
Precision for 0	0.97	0.99	0.99
Precision for 1	0.97	0.98	0.98
Recall for 0	0.97	0.98	0.98
Recall for 1	0.97	0.99	0.99
F1-score for 0	0.97	0.98	0.99
F1-score for 1	0.97	0.98	0.99
Overall Accuracy	0.97	0.98	0.99

So, from above table we can conclude that xgboost has a good accuracy rate with 0.99 so we test the model by saving that xgboost model.

9.SCREEN SHOT OF RESULTS

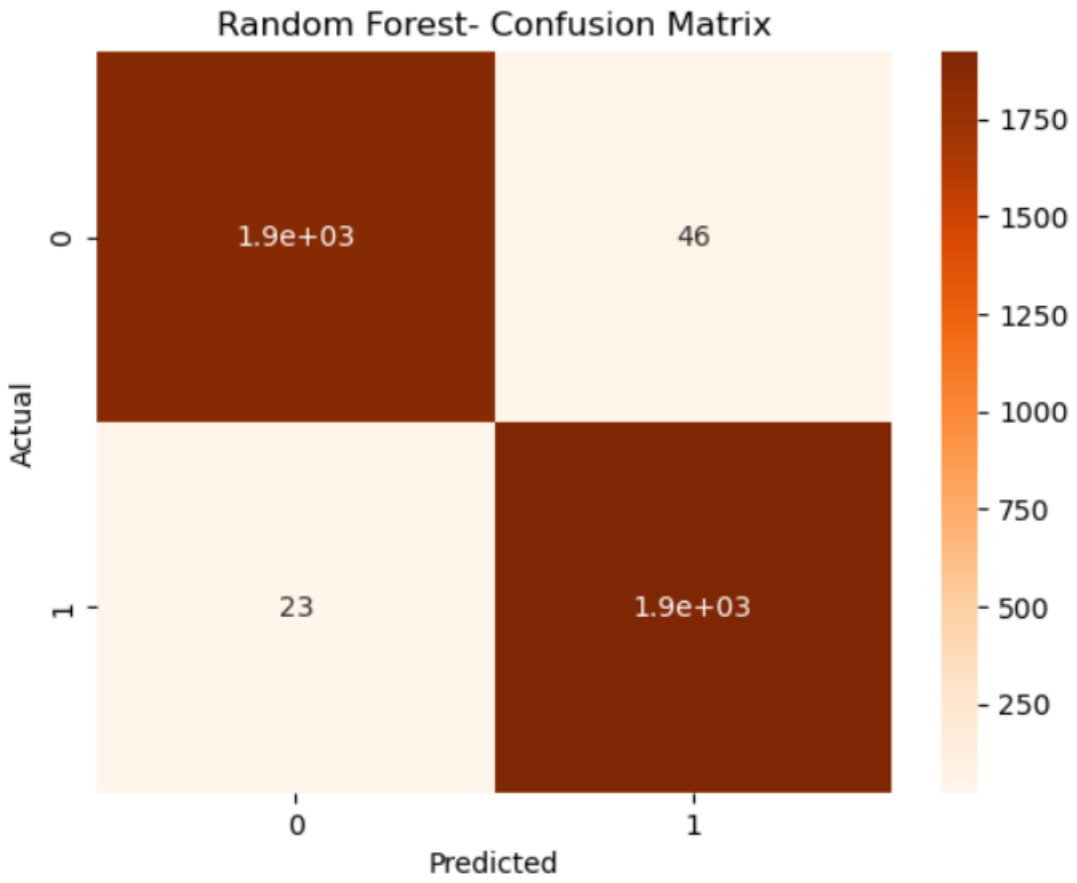
Decision Tree:

Confusion Matrix of a Decision Tree Model



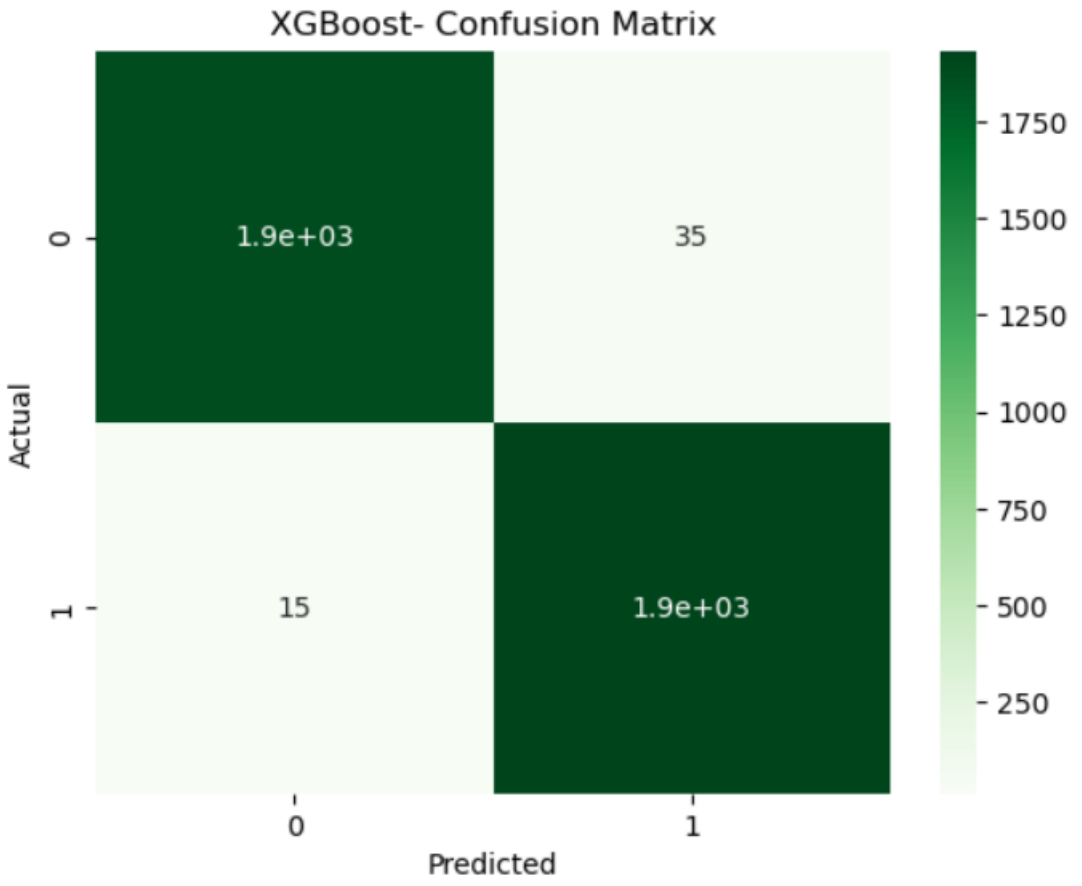
Random Forest:

The confusion matrix for a random forest model

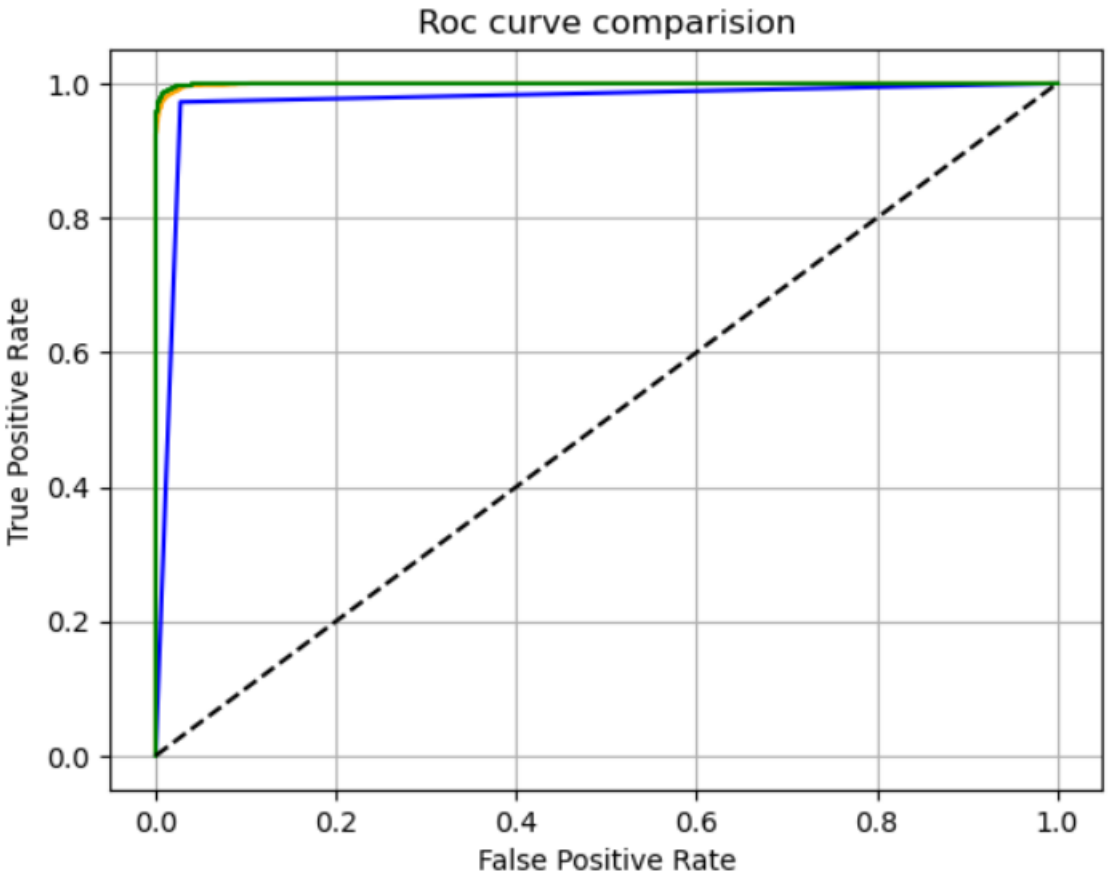


XG-Boost:

Confusion Matrix of a XG-Boost Model



AUC-ROC Curve Comparison screenshot for three models:



10. CONCLUSION

So from above all comparison we can conclude that the XGboost model is the best model for testing the project. And no special scaling made as the accuracy rates for above almost 1.

REFERENCES:

- https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/
- https://www.analyticsvidhya.com/blog/2015/11/beginners-guide-on-logistic-regression-in-r/
- https://www.danielsoper.com/statcalc/calculator.aspx?id=8
- https://sebastianraschka.com/Articles/2014_about_feature_scaling.html#about-standardization
- https://www.statisticshowto.datasciencecentral.com/probability-and-statistics/chi-square/
- http://www.insightsbot.com/blog/2AeuRL/chi-square-feature-selection-in-python
- https://towardsdatascience.com/the-dummys-guide-to-creating-dummy-variables-f21faddb1d4 0
- https://hub.packtpub.com/4-ways-implement-feature-selection-python-machine-learning/
- https://cmdlinetips.com/2018/02/how-to-get-frequency-counts-of-a-column-in-pandas-datafra me/
- https://www.youtube.com/watch?v=V0u6bxQOUJ8