Equipment Failure Prediction Using Machine Learning

- Madhu Singh

1. Introduction

Predictive maintenance plays a critical role in industrial operations by identifying potential equipment failures before they occur. This report explores the application of machine learning techniques to predict equipment failures based on the operational parameters. Accordingly, the machine downtime and maintenance cost can be reduced significantly while making the maintenance frequency as low as possible.

2. Objectives

The primary objectives of this analysis are:

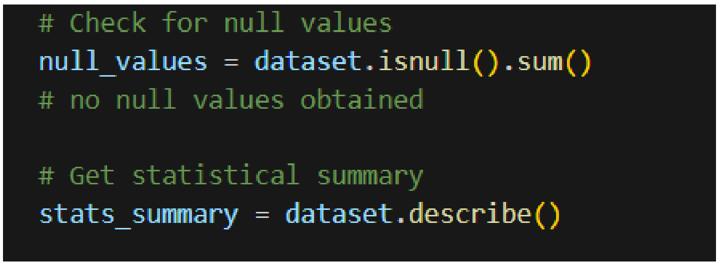
- Explore and understand the dataset.
- Preprocess the data by handling missing values, visualizing distributions, and normalizing features.
- Train and evaluate three classification models (Decision Tree, Random Forest, and K-Nearest Neighbours) to predict equipment failure (Target variable).
- Compare and interpret the performance of these models based on accuracy metrics.

3. Methodology

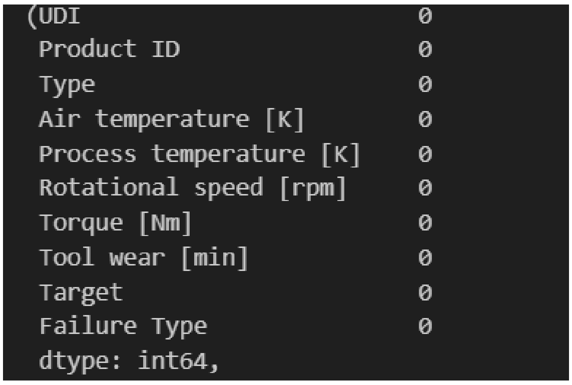
3.1 Data Exploration and Preprocessing

Data Loading and Inspection:

- The dataset is loaded into a pandas DataFrame and basic checks are performed to ensure data integrity (null_values and stats_summary).
- Missing values are checked using **.isnull().sum()** to ensure data completeness. In this specific case, no missing values were found, as indicated by the null_values variable.
- Statistical summaries (stats_summary) are generated using **.describe()** to gain initial insights into the numerical features of the dataset.



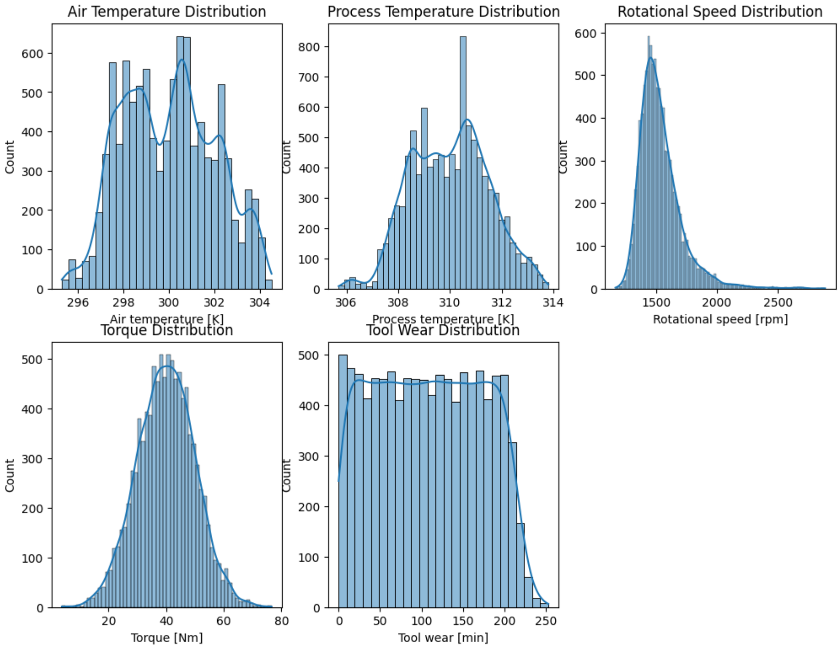
```
# Check for null values
null_values = dataset.isnull().sum()
# no null values obtained

# Get statistical summary
stats_summary = dataset.describe()
```



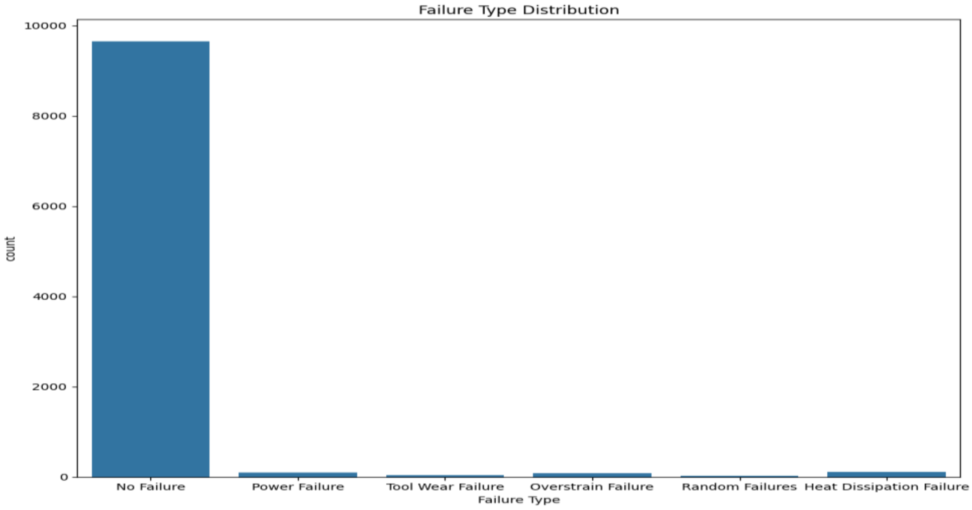
```
(UDI                        0
 Product ID                 0
 Type                       0
 Air temperature [K]        0
 Process temperature [K]    0
 Rotational speed [rpm]     0
 Torque [Nm]                0
 Tool wear [min]            0
 Target                     0
 Failure Type               0
 dtype: int64,
```

Data Visualization:

- Histograms are used to visualize the distributions of key operational parameters (Air temperature [K], Process temperature [K], Rotational speed [rpm], Torque [Nm], Tool wear [min]).



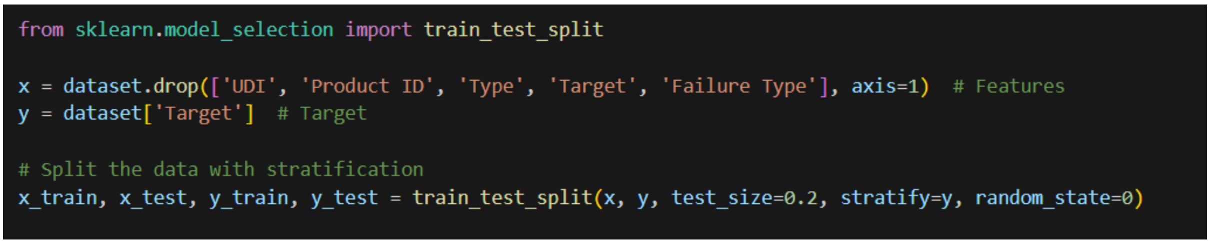
- The following plot illustrates the distribution of the Failure Type.



Data Splitting and Normalization:

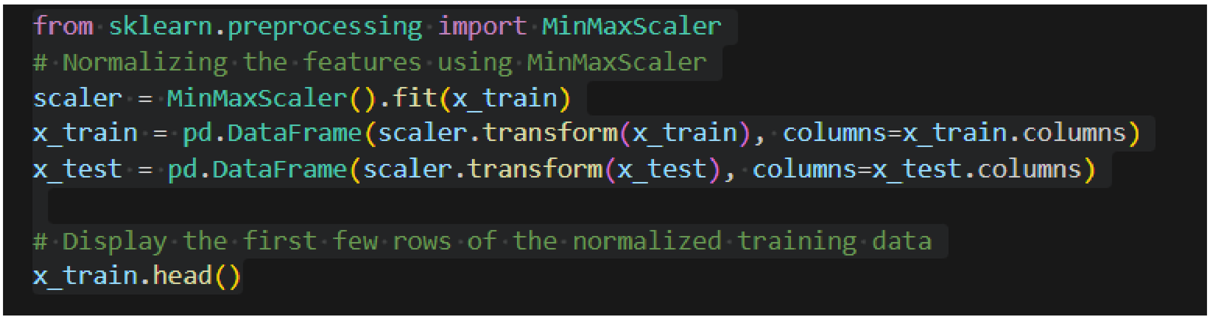
- Data Splitting: The dataset is split into features (x) and the target (y). Features like UDI, Product ID, Type, Target and Failure Type are excluded from features (x) using .drop().
- Target Variable: The target variable (y) is set to Target, which represents the binary outcome to be predicted.

- Train-Test Split: Using train_test_split, the dataset is divided into training (x_train, y_train) and testing (x_test, y_test) sets. Stratified sampling is used (stratify=y) to maintain the proportion of classes in the target variable (Target).



```
from sklearn.model_selection import train_test_split

x = dataset.drop(['UDI', 'Product ID', 'Type', 'Target', 'Failure Type'], axis=1)  # Features
y = dataset['Target']  # Target

# Split the data with stratification
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, stratify=y, random_state=0)
```

- Features are normalized using MinMaxScaler to standardize the scale across different features.



```
from sklearn.preprocessing import MinMaxScaler
# Normalizing the features using MinMaxScaler
scaler = MinMaxScaler().fit(x_train)
x_train = pd.DataFrame(scaler.transform(x_train), columns=x_train.columns)
x_test = pd.DataFrame(scaler.transform(x_test), columns=x_test.columns)

# Display the first few rows of the normalized training data
x_train.head()
```

3.2 Model Training and Evaluation

Model Selection and Training:

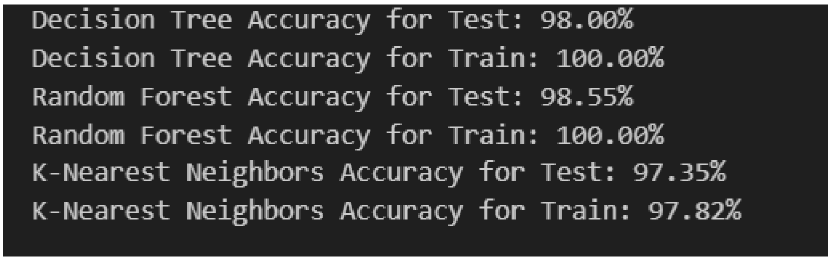
- Three classifiers are selected and trained on the normalized training data:
  - **Decision Tree**
  - **Random Forest**
  - **K-Nearest Neighbors**

Model Evaluation:

- The accuracy of each model is evaluated using accuracy_score on both training and testing sets to assess performance.

4. Results

The Random Forest Classifier exhibited the highest accuracy on the test set, suggesting it is the most effective in predicting equipment failures in unseen data.



```
Decision Tree Accuracy for Test: 98.00%
Decision Tree Accuracy for Train: 100.00%
Random Forest Accuracy for Test: 98.55%
Random Forest Accuracy for Train: 100.00%
K-Nearest Neighbors Accuracy for Test: 97.35%
K-Nearest Neighbors Accuracy for Train: 97.82%
```

5. Conclusion

This analysis demonstrates the feasibility and effectiveness of using machine learning models for equipment failure prediction. The Random Forest classifier showed the highest accuracy on the test set, indicating its potential for practical deployment in predictive maintenance strategies. These findings underscore the potential of machine learning in enhancing operational efficiency and reducing costs associated with unplanned equipment downtime.

6. References

https://yavisankar.medium.com/equipment-failure-prediction-using-machine-learning-models-8be0206eed6b