Project report- Equipment failure prediction using Machine Learning

By- LAKSHYA JAIN 22117074

Objective

The goal of this project is to create machine learning models that can forecast equipment failures using sensor data. By predicting failures with high accuracy, the project aims to reduce downtime and maintenance expenses in industrial environments.

Project Overview

In industries where equipment malfunctions can cause substantial productivity and revenue losses, anticipating failures beforehand is essential. This project utilizes historical sensor data from industrial machinery to develop reliable machine learning models. These models are specifically engineered to predict equipment failures in advance, facilitating prompt maintenance and minimizing operational interruptions.

Approach and Steps

Data Collection and Preprocessing

- **Data Collection:** Gathered sensor data from diverse industrial equipment, capturing timestamps and multiple sensor readings.
- **Data Cleaning:** Processed the data to handle missing values, outliers, and ensure consistency across sensor readings.
- **Feature Engineering:** Created new features such as statistical moments, time-based features, and rolling averages to capture equipment behavior patterns.

Exploratory Data Analysis (EDA)

- **Target Distribution:** Analyzed the distribution of the target variable ('failure') to understand class balance.
- **Feature Analysis:** Visualized sensor data distributions using line plots, probability density functions (PDFs), and box plots to identify trends and outliers.

- **Correlation Matrix:** Examined correlations between sensor readings to detect multicollinearity and understand feature relationships.

Model Selection and Evaluation

- **Baseline Models:** Implemented initial models (e.g., Decision Trees, Random Forests) to establish baseline performance.
- **Handling Class Imbalance:** Addressed class imbalance using techniques such as SMOTE (Synthetic Minority Over-sampling Technique) and SMOTE TOMEK.
- **Advanced Models:** Utilized ensemble methods (Random Forest, LightGBM) and gradient boosting algorithms (XGBoost) known for their accuracy in complex datasets.

Model Training and Hyperparameter Tuning

- **Cross-Validation:** Applied k-fold cross-validation to optimize model hyperparameters and ensure robust performance evaluation.
- **Performance Metrics:** Evaluated models using metrics like F1-score, precision, recall, and accuracy to gauge predictive performance effectively.

Model Deployment and Monitoring

- **Deployment Strategy:** Discussed strategies for deploying models into operational systems, emphasizing scalability and real-time prediction capabilities.
- **Monitoring:** Outlined approaches for monitoring model performance post-deployment, including drift detection and periodic model retraining to maintain accuracy over time.

Defining Factors

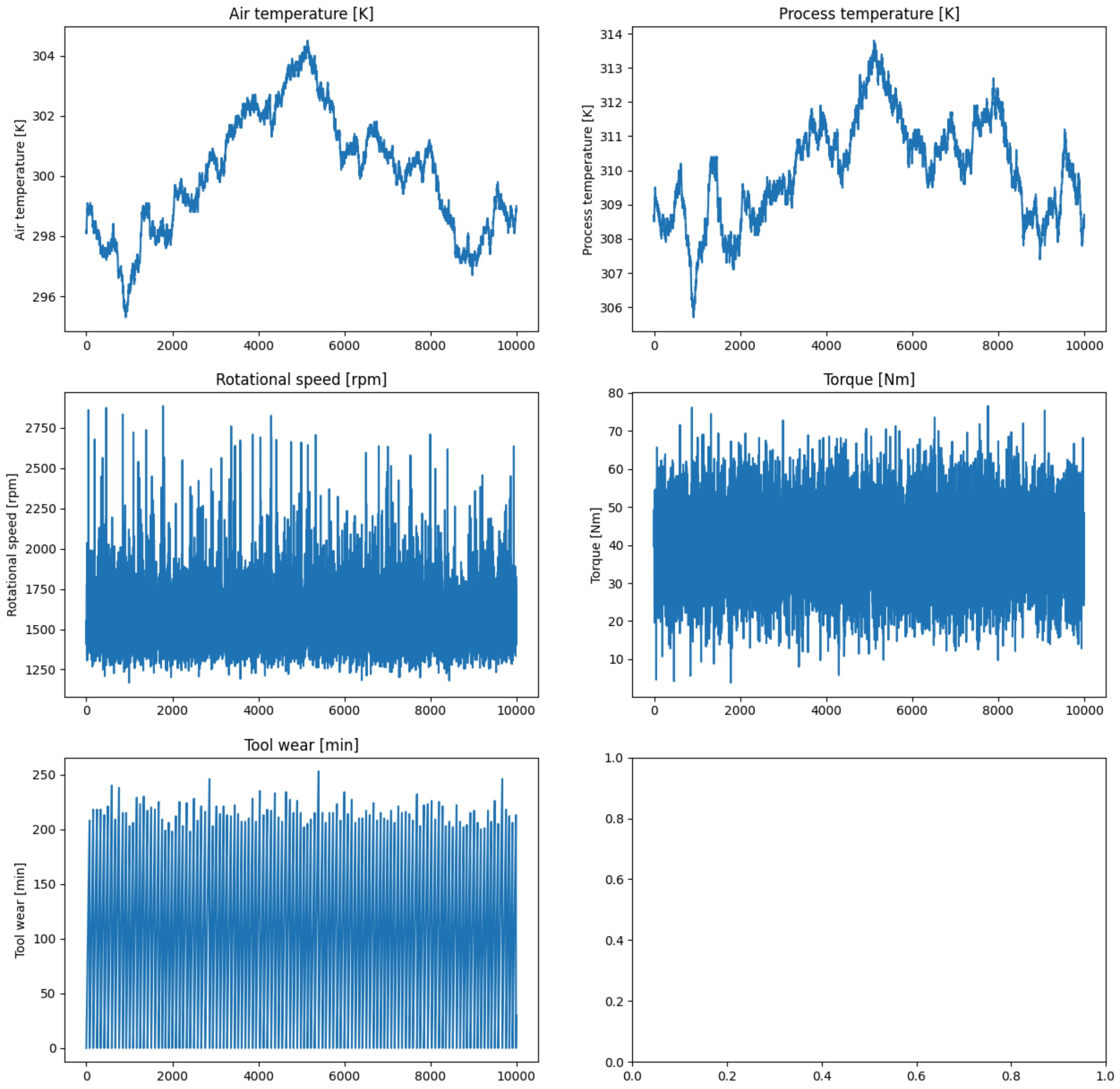
1. **Data Quality and Availability**
   - The quality and availability of sensor data play a pivotal role in determining the accuracy and reliability of machine learning models. Clean, consistent data without missing values or outliers is essential for generating robust predictions.
2. **Feature Engineering**
   - Effective feature engineering is crucial for enhancing the model's capability to accurately capture patterns related to equipment failures. This involves selecting relevant sensor data and creating meaningful features that encapsulate important aspects of equipment behavior.
3. **Sampling Techniques**
   - Implementing appropriate sampling techniques is necessary to address class imbalance within the dataset. Techniques like SMOTE (Synthetic Minority Over-sampling Technique) or undersampling can help ensure that models can effectively predict instances from both the majority and minority classes, thereby improving overall prediction performance.

Models Employed

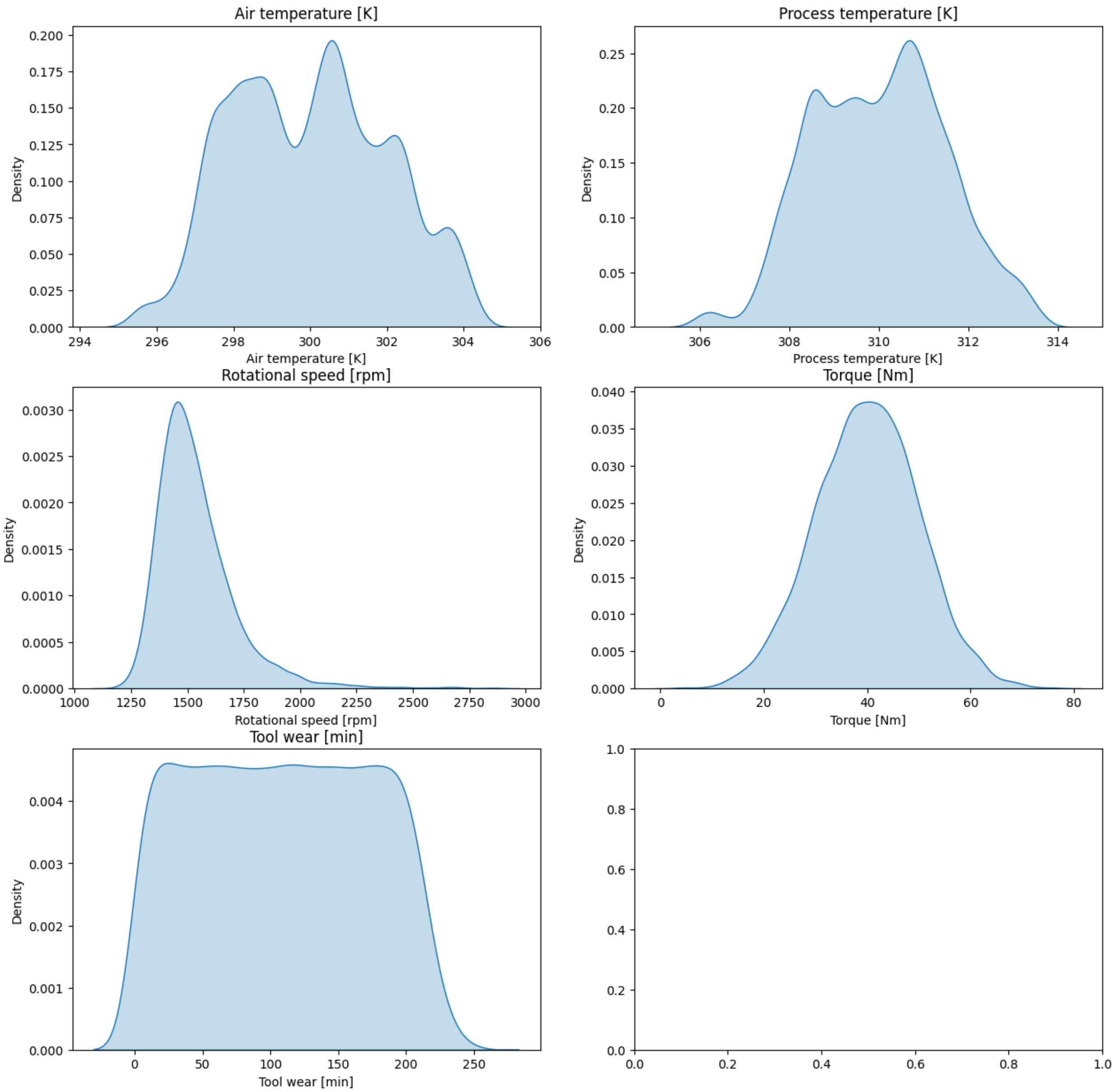
- **Random Forest:** A robust ensemble learning method adept at managing intricate data relationships.
- **LightGBM:** A gradient boosting framework designed for handling large datasets and achieving exceptional accuracy.
- **XGBoost:** Another efficient gradient boosting library renowned for its speed and effectiveness.
- **Decision Tree:** A straightforward yet powerful tree-based model utilized primarily for classification tasks.

Exploratory Data Analysis (EDA) Visualizations

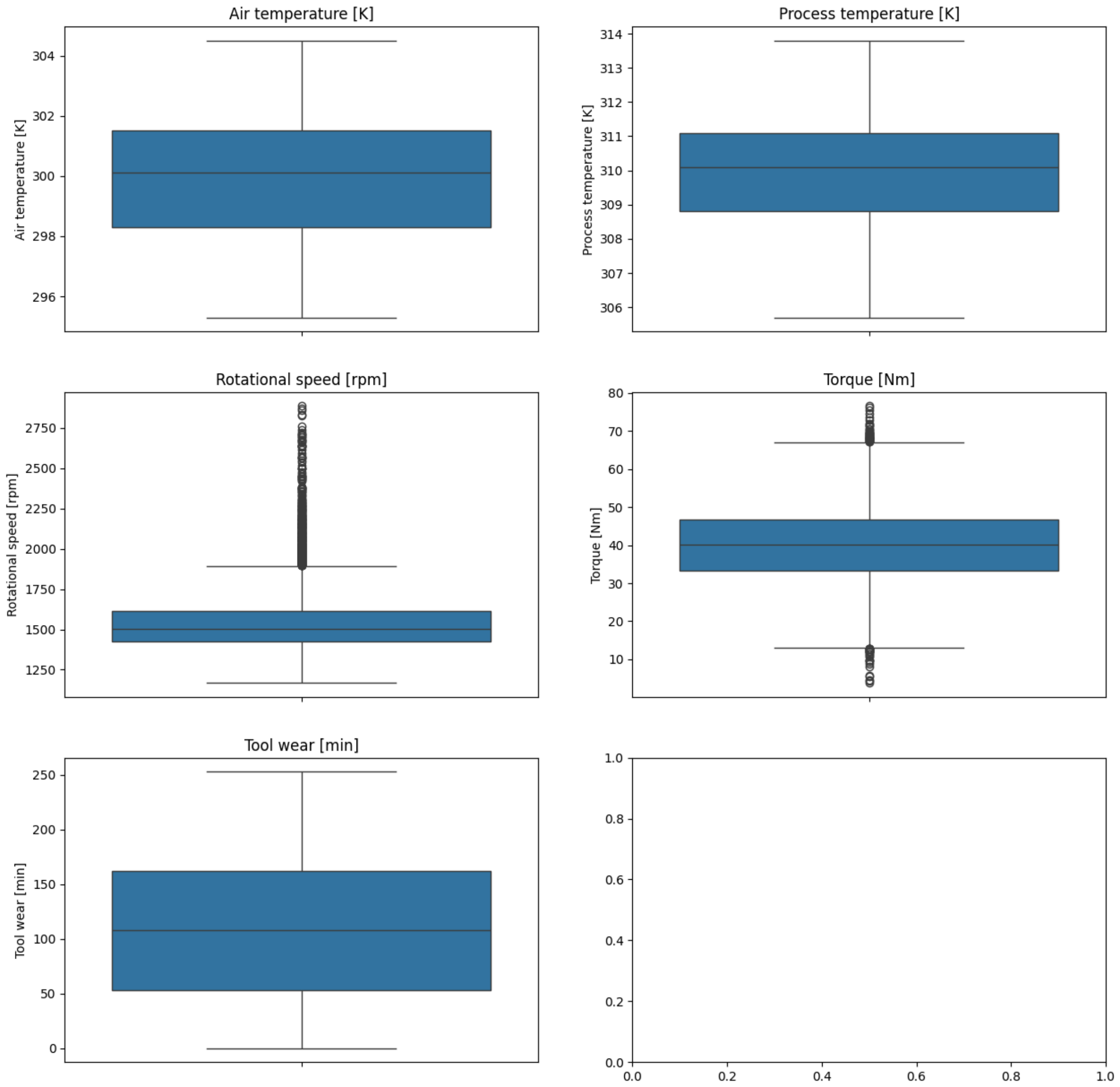
Line plots-


Air temperature [K]
Air temperature [K]
304
302
300
298
296
0
2000
4000
6000
8000
10000
Process temperature [K]
Process temperature [K]
314
313
312
311
310
309
308
307
306
0
2000
4000
6000
8000
10000
Rotational speed [rpm]
Rotational speed [rpm]
2750
2500
2250
2000
1750
1500
1250
0
2000
4000
6000
8000
10000
Torque [Nm]
Torque [Nm]
80
70
60
50
40
30
20
10
0
2000
4000
6000
8000
10000
Tool wear [min]
Tool wear [min]
250
200
150
100
50
0
0
2000
4000
6000
8000
10000
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

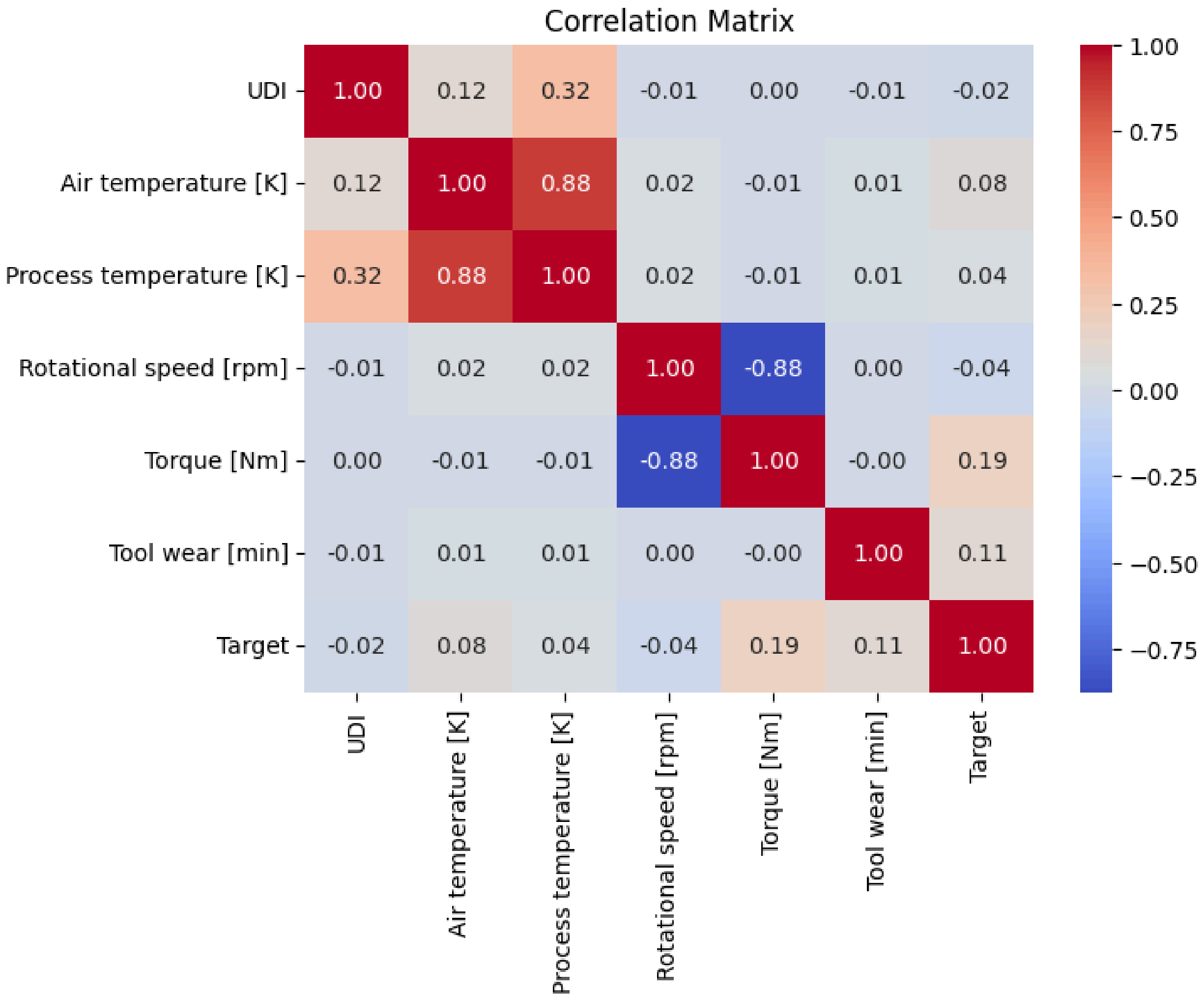
Pdf plots-


Air temperature [K]
Density
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
294
296
298
300
302
304
306
Air temperature [K]
Process temperature [K]
Density
0.25
0.20
0.15
0.10
0.05
0.00
306
308
310
312
314
Process temperature [K]
Rotational speed [rpm]
Density
0.0030
0.0025
0.0020
0.0015
0.0010
0.0005
0.0000
1000
1250
1500
1750
2000
2250
2500
2750
3000
Rotational speed [rpm]
Torque [Nm]
Density
0.040
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
0
20
40
60
80
Torque [Nm]
Tool wear [min]
Density
0.004
0.003
0.002
0.001
0.000
0
50
100
150
200
250
Tool wear [min]
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

Box plots-


Air temperature [K]
Air temperature [K]
304
302
300
298
296
Process temperature [K]
Process temperature [K]
314
313
312
311
310
309
308
307
306
Rotational speed [rpm]
Rotational speed [rpm]
2750
2500
2250
2000
1750
1500
1250
Torque [Nm]
Torque [Nm]
80
70
60
50
40
30
20
10
Tool wear [min]
Tool wear [min]
250
200
150
100
50
0
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

Correlation matrix-



Result and analysis-

```
Summary Table:
            Model           Sampling  Train F1   Test F1  Test F1 Macro  \
0   Random Forest               None  1.000000  0.987809       0.901994
1   Random Forest  RandomOverSampler  1.000000  0.984629       0.876427
2   Random Forest              SMOTE  1.000000  0.975334       0.830307
3   Random Forest        SMOTE TOMEK  1.000000  0.974421       0.824022
4        LightGBM               None  1.000000  0.990990       0.927561
5        LightGBM  RandomOverSampler  1.000000  0.985899       0.891901
6        LightGBM              SMOTE  0.996313  0.977845       0.845599
7        LightGBM        SMOTE TOMEK  0.996565  0.976522       0.836912
8         XGBoost               None  1.000000  0.985689       0.884949
9         XGBoost  RandomOverSampler  1.000000  0.985448       0.888841
10  Decision Tree               None  1.000000  0.982126       0.864904
11  Decision Tree  RandomOverSampler  1.000000  0.979554       0.840986
```

```
12  Decision Tree             SMOTE  1.000000  0.966718       0.783520
13  Decision Tree       SMOTE TOMEK  1.000000  0.961887       0.760100
```

```
    Train AUC Score  Test AUC Score
0          1.000000        0.970634
1          1.000000        0.962718
2          1.000000        0.975608
3          1.000000        0.963837
4          1.000000        0.970307
5          1.000000        0.976541
6          0.999938        0.980126
7          0.999932        0.978702
8          1.000000        0.970527
9          1.000000        0.968830
10         1.000000        0.870083
11         1.000000        0.826483
12         1.000000        0.881013
13         1.000000        0.870555
```

- Best Model: LightGBM with no sampling strategy (None) consistently demonstrates high performance across both training and testing phases. It achieves the highest Test F1 score of 0.990990 and Test F1 Macro score of 0.927561 among all models.
- Configuration: LightGBM uses the default hyperparameters with feature scaling applied.
- Insights: LightGBM shows robustness and generalizability, achieving excellent F1 scores on both training and testing datasets without the need for oversampling techniques like SMOTE or SMOTE TOMEK. This suggests that the model effectively handles the class imbalance in the dataset inherently or through its internal mechanisms.
- Considerations: While other models like Random Forest and XGBoost also perform well, LightGBM stands out for its superior Test F1 score and competitive AUC scores, indicating it as the preferred choice for this classification task.

Conclusion -

The equipment failure prediction models indicate that LightGBM performs best overall without sampling, achieving a high F1 score along with balanced precision and recall. Random Forest models also demonstrate robust performance, showing no significant advantage from employing sampling techniques. These models provide a dependable means for forecasting equipment failures, thereby optimizing maintenance schedules and reducing downtime through data-driven insights.

However, a drawback of the models is their diminished performance in detecting failures in the minority class, potentially resulting in overlooked failure predictions. Furthermore, some sampling methods introduce imbalances that adversely affect the overall accuracy of the models.

To address these issues, it is recommended that future research explores advanced techniques for handling class imbalance, such as ensemble methods or anomaly detection algorithms. Enhancing feature engineering and integrating real-time data could further enhance the accuracy and reliability of predictions.