



Predictive Maintenance

DePaul University, Chicago, Illinois, USA

Group #11: Adithya Harsha, Joao Vitor Lira de Carvalho Firmino, Beemnet Desta, Tejas

About our Dataset



- The dataset we selected is the Predictive Maintenance Dataset, sourced from Kaggle, which contains 10,000 data points and 14 features.
- These features capture various operational conditions, such as air temperature, process temperature, rotational speed, torque, and tool wear, along with a machine failure label.
- Machine failures are categorized into five types: tool wear failure, heat dissipation failure, power failure, overstrain failure, and random failures.


Goal:

The goal of this project is to predict machine failures using operational data. By doing so, we hope to improve maintenance schedules, reduce downtime, and avoid premature replacements.

Use:

This prediction model will assist industries in implementing predictive maintenance strategies, ensuring that machines are serviced at the most appropriate time. This reduces operational expenses, improves machine reliability, and avoids unexpected breakdowns. Ultimately, it will result in increased production and cost savings for companies that rely on heavy gear.

Pre processing and Cleaning



- There were no null/missing values.
- Checked for empty strings in the 'Product ID' and 'Type' columns, and found none.
- Detected outliers in 'Rotational Speed [rpm]' (418 outliers) and 'Torque [Nm]' (69 outliers) using the IQR method.
- Applied log transformation to 'Rotational Speed [rpm]' and 'Torque [Nm]' to reduce the impact of extreme values and normalize the distribution.
- Converted the categorical 'Type' feature into numerical binary features using One-Hot Encoding.
- Standardized numerical features including 'Air temperature [K]', 'Process temperature [K]', 'Rotational Speed [rpm]', 'Torque [Nm]', and 'Tool wear [min]' using StandardScaler to ensure equal contribution during modeling.
- Altogether, we now have transformed and scaled features ready for machine learning models.

Outlier Detection & Handling



- Outlier Detection:
 - We applied the Interquartile Range (IQR) method to identify outliers in the numerical columns:
 - Rotational Speed [rpm]: 418 outliers detected.
 - Torque [Nm]: 69 outliers detected.
 - No outliers found in 'Air temperature [K]', 'Process temperature [K]', or 'Tool wear [min]'.
- Handling Outliers:
 - To reduce the impact of extreme values, we applied log transformation to the 'Rotational Speed [rpm]' and 'Torque [Nm]' columns:
 - Rotational Speed [rpm]: After transformation, the data is concentrated around a tighter range, with fewer extreme outliers.
 - Torque [Nm]: The transformed data is more compressed, with smaller impacts from outliers on the lower end.

Feature Scaling

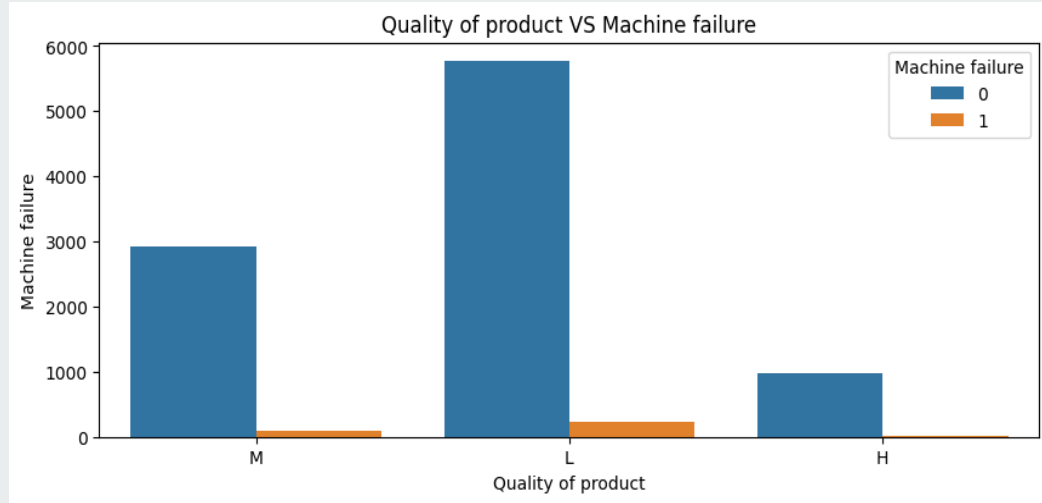
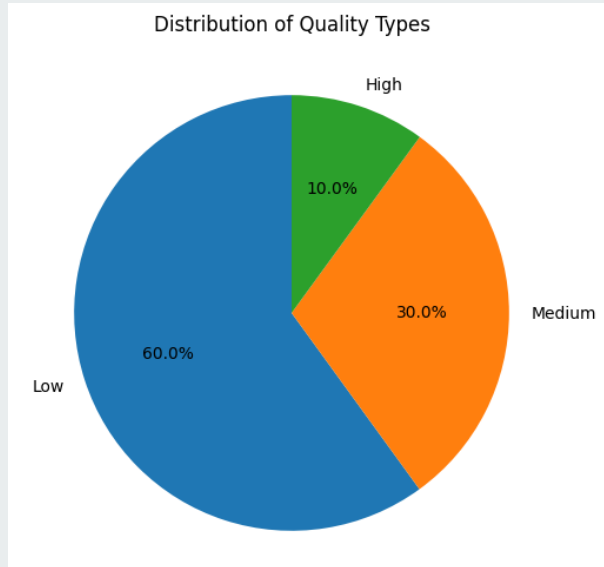
The features in the dataset have different ranges, which can affect the performance of machine learning models. To ensure all features contribute equally, we applied standardization. After considering several options, we decided on using StandardScaler, which standardizes features by giving them: Mean = 0 and Standard Deviation = 1

We made sure that scaling was applied only to the numeric features and not to the categorical ones.

- Features Scaled:
 - 'Air temperature [K]'
 - 'Process temperature [K]'
 - 'Rotational speed [rpm]'
 - 'Torque [Nm]'
 - 'Tool wear [min]'

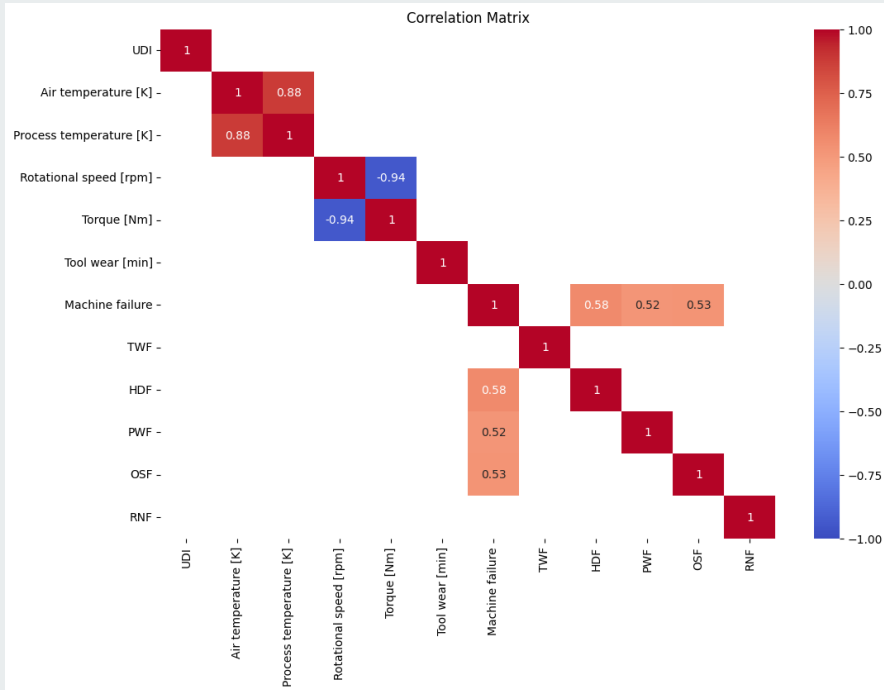
Exploratory Data Analysis(EDA)

Pie chart illustrates that, within this dataset, 60% of the manufactured products are of low quality, 30% are of medium quality, and 10% are of high quality.



Machine failure based on product Quality (Bar Chart): In this graph, it is possible to see that almost no high-quality products are related to machine failure. On the other hand, almost all failures occur during the manufacturing of low-quality products.

Correlation Matrix:

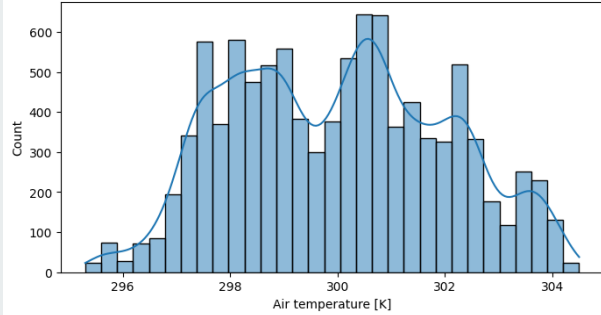


- **Air Temperature [K] & Process Temperature [K] (0.88 correlation):** These variables are linked to the machine's thermal dynamics. Higher air temperatures make cooling harder, leading to increased process temperatures.
- **Rotational Speed [rpm] & Torque [Nm] (-0.94 correlation):** Typically, as rotational speed increases, torque decreases, indicating an inverse relationship crucial for maintaining consistent power output.
- **Different Failure Types (HDF, PWF, OSF):** Failure modes like HDF, PWF, and OSF are interconnected. For instance, poor heat dissipation (HDF) can cause electrical issues, leading to power failures (PWF)

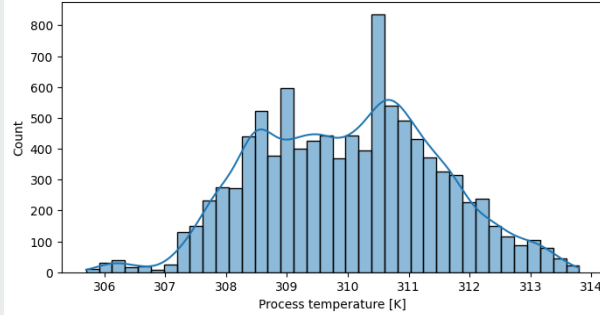
Histograms of features:



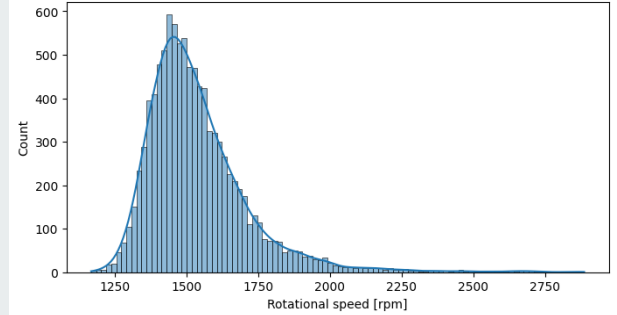
Histogram of Air temperature [K]



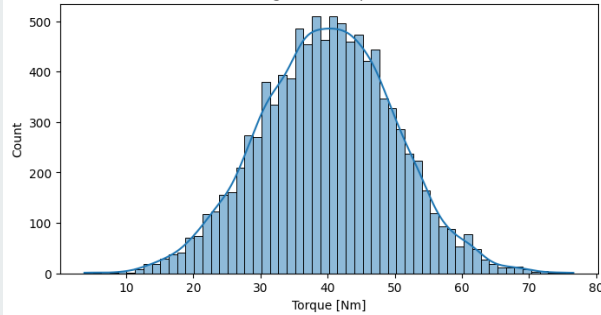
Histogram of Process temperature [K]



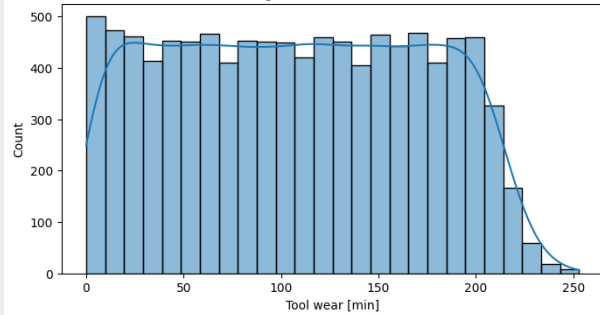
Histogram of Rotational speed [rpm]



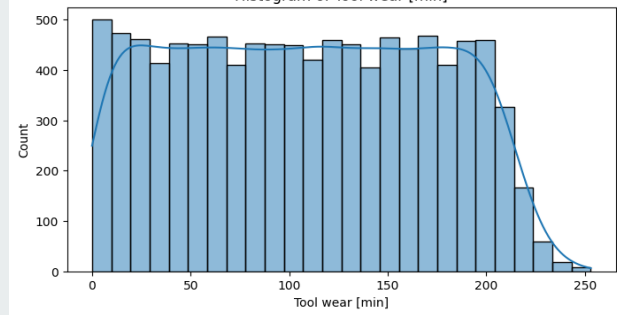
Histogram of Torque [Nm]



Histogram of Tool wear [min]




Histogram of Tool wear [min]





- **Air Temperature Histogram:** The distribution is bimodal, with concentrations around 298 K and 301 K.
- **Process Temperature Histogram:** Values range from 306 K to 314 K, with most clustered between 309 K and 312 K. A prominent peak between 310 K and 311 K suggests most temperatures are centered here, though higher temperatures occur less often.
- **Rotational Speed Histogram:** Right-skewed with a peak around 1500 rpm, indicating most data points cluster at lower speeds, with fewer high-speed instances.
- **Torque Histogram:** Displays a normal distribution, with a mean torque of approximately 40 Nm.
- **Tool Wear Histogram:** Tool wear ranges from 0 to 200 minutes, with a nearly uniform distribution and a sharp decline after 200 minutes.

Feature Engineering



- **New Features Created:**
 - **Mechanical_Power:**
 - Captures the machine's output using the formula: $\text{Mechanical Power} = \text{Rotational Speed} \times \text{Torque}$
 - **Key Insight:** Strong inverse correlation between Rotational Speed and Torque (**-0.94**).
 - **Temp_Diff:**
 - Highlights potential overheating issues with the formula: $\text{Temp_Diff} = \text{Process Temperature} - \text{Air Temperature}$
 - **Key Insight:** Strong positive correlation between Air and Process Temperatures (**0.88**).
- Log Transformation Applied After feature creation to avoid distorting Mechanical Power.
 - This maintains the accuracy of features like Mechanical Power, which could become negative or zero, distorting the data before transformation.

Model Preparation and Training



Scaling Removed for Tree-Based Models:

- Random Forest, Gradient Boosting, and XGBoost don't require scaling, simplifying preprocessing.
- Tree-based models split features based on order, so the magnitude of values (e.g., speed or temperature) doesn't affect their performance.


Column Renaming:

- Renamed columns to remove special characters like '[', ']', and spaces. This was necessary for **XGBoost** to avoid errors during model training, as XGBoost requires column names without special characters.

Train-Test Split:

- Used a train-test split of 80% training data and 20% test data with a `random_state=42` for reproducibility.

Scaling Selected Features



- The goal was to Standardize features to improve model performance and convergence.
- Columns Scaled: Air_temperature_K, Process_temperature_K, Rotational_speed_rpm, Torque_Nm, Tool_wear_min.
- We used StandardScaler to apply scaling.
- This ensures features are on a similar scale, reducing bias toward larger values.
- Post-scaling, mean values are close to 0, and standard deviations are close to 1, indicating successful normalization.

Train/Test Split



- The purpose is to separate the data for model evaluation and to avoid overfitting.
- Target variable (Any_Failure) indicates any type of machine failure.
- Feature selection:
 - Dropped columns unrelated to model training (Machine_failure and target column).
 - Ensures only predictive features are included in X.
- Outcome:
- The split ensures the model is trained on a larger portion of data while retaining enough for reliable testing.

Model Selection



- Tree-based models were selected due to their ability to handle complex interactions between features.:
 - Random Forest;
 - Gradient Boosting;
 - XGBoost.
- Additionally, other models were applied to evaluate performance across different types of algorithms:
 - AdaBoost;
 - Logistic Regression.

Model Training, Prediction, and Evaluation



- We trained the models on the SMOTE-balanced data using only the selected features.
- This step ensured that the training data was balanced and tailored to the important features, enhancing the model's ability to accurately detect instances of the minority class.
- After training, predictions were made on the test set, and the model's performance was evaluated using metrics such as accuracy, precision, recall, and F1-score.

Model Training, Prediction, and Evaluation



Random Forest:

- Final Accuracy: 0.965
- Precision: 0.47
- Recall: 0.68
- F1: 0.56

Results for Random Forest:

	precision	recall	f1-score	support
0	0.99	0.97	0.98	1935
1	0.47	0.68	0.56	65
accuracy			0.96	2000
macro avg	0.73	0.83	0.77	2000
weighted avg	0.97	0.96	0.97	2000

Accuracy: 0.965

Model Training, Prediction, and Evaluation



Gradient Boosting Model:

- Final Accuracy: 0.938
- Precision: 0.32
- Recall: 0.78
- F1: 0.45

Results for Gradient Boosting:

	precision	recall	f1-score	support
0	0.99	0.94	0.97	1935
1	0.32	0.78	0.45	65
accuracy			0.94	2000
macro avg	0.65	0.86	0.71	2000
weighted avg	0.97	0.94	0.95	2000

Accuracy: 0.938

Model Training, Prediction, and Evaluation



XGBoost Model:

- Final Accuracy: 0.978
- Precision: 0.64
- Recall: 0.72
- F1:0.68

```
Results for XGBoost:
              precision    recall  f1-score   support

      0         0.99      0.99      0.99     1935
      1         0.64      0.72      0.68        65

 accuracy          0.98     2000
 macro avg         0.82     0.85     0.83     2000
weighted avg         0.98     0.98     0.98     2000

Accuracy: 0.978
```

Model Training, Prediction, and Evaluation

Threshold Tuning on XGBoost Model: It was employed to balance recall and precision further, enhancing sensitivity to true failure cases.

- Final Accuracy: 0.967
- Precision: 0.49
- Recall: 0.74
- F1: 0.59

Results for XGBoost with SMOTE and Threshold 0.3:

	precision	recall	f1-score	support
0	0.99	0.97	0.98	1935
1	0.49	0.74	0.59	65
accuracy			0.97	2000
macro avg	0.74	0.86	0.79	2000
weighted avg	0.97	0.97	0.97	2000

Accuracy: 0.967

Model Training, Prediction, and Evaluation



AdaBoost Model:

- Final Accuracy: 0.904
- Precision: 0.22
- Recall: 0.75
- F1:0.34

Results for AdaBoost Classifier:

	precision	recall	f1-score	support
0	0.99	0.91	0.95	1935
1	0.22	0.75	0.34	65
accuracy			0.90	2000
macro avg	0.60	0.83	0.64	2000
weighted avg	0.97	0.90	0.93	2000

Accuracy: 0.904

Model Training, Prediction, and Evaluation



Logistic Regression Model:

- Final Accuracy: 0.705
- Precision: 0.07
- Recall: 0.63
- F1:0.12

```
Results for LogisticRegression Classifier:
              precision    recall  f1-score   support

         0           0.98        0.71        0.82        1935
         1           0.07        0.63        0.12          65

 accuracy                   0.70        2000
 macro avg           0.53        0.67        0.47        2000
weighted avg           0.95        0.70        0.80        2000

Accuracy: 0.705
```

Conclusion



In this predictive maintenance project, multiple models were evaluated to predict failures. In conclusion, several key points can be highlighted:

- **Top Performing Model:**
 - XGBoost achieved the highest performance with an accuracy of 0.978.
 - Threshold tuning further improved recall to 0.74, optimizing sensitivity to true failures.
- **Model Comparisons:**
 - Random Forest also performed well but had slightly lower precision and recall.
 - Gradient Boosting and AdaBoost provided acceptable recall but lacked precision balance.
 - Logistic Regression showed the lowest metrics and was not suited for this task.

Conclusion



- **Main Insights:**
 - XGBoost, especially with threshold tuning, is the most effective model for this dataset.
 - A balanced approach between recall and precision was essential for reliable failure prediction.
- **Practical Implications:**
 - The model's reliability can enhance maintenance scheduling and reduce unexpected downtimes..



Thank You