IoT Analytics Assignment

Predictive Machine Maintenance (AI4I2020 Dataset)



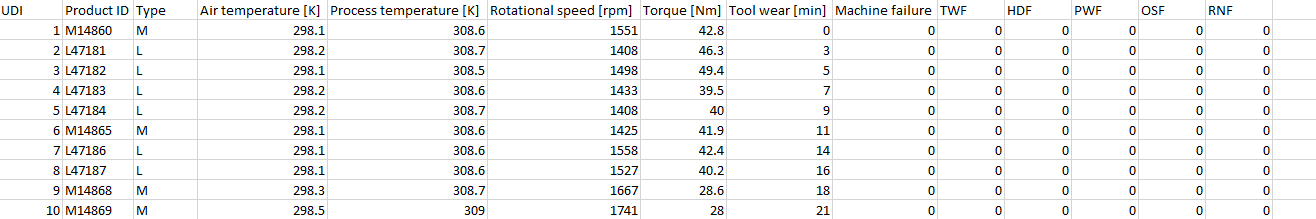
**Group 5**

|  |  |
| --- | --- |
| **Name** | **PRN** |
| Atharva Pandit | 23030242012 |
| Prachi Mishra | 23030242042 |
| Raghav Kapoor | 23030242047 |

Project Overview

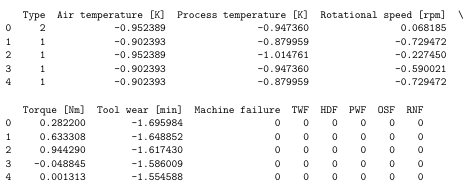
Predictive maintenance is a critical application in industrial IoT, aiming to minimize machine failures, optimize resource utilization, and reduce maintenance costs. This project leverages machine learning techniques to predict machine failures based on IoT sensor data. The objective is to build a classification model that can determine whether a machine is likely to fail based on real-time sensor readings and deploy it using a Streamlit-based web application.

Understanding the Dataset



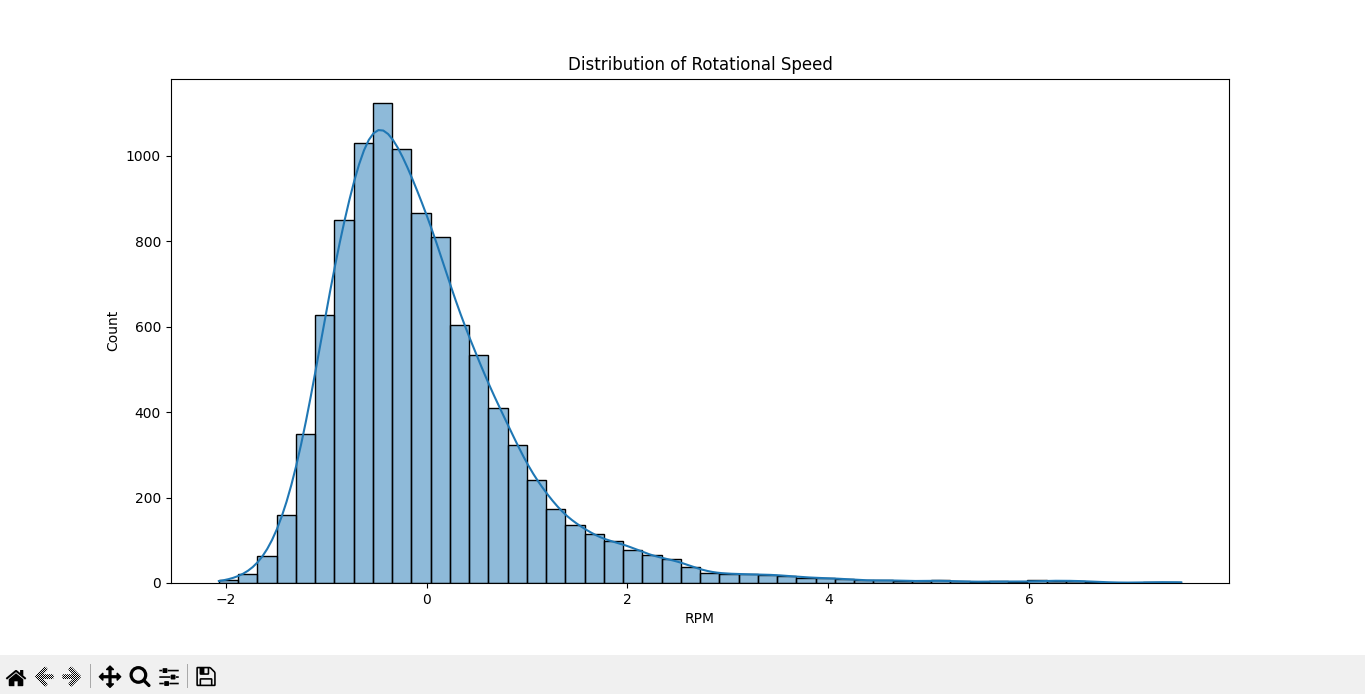
**AI4I 2020 Predictive Maintenance Dataset**

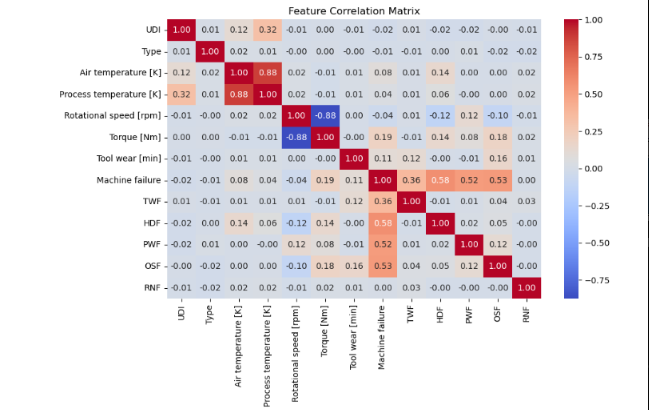
* The dataset contains sensor readings from industrial machines along with machine failure labels.
* It includes **10,000+ records** with multiple machine types and failure modes.
* **Features:**
  + **Continuous Variables:** Air temperature, Process temperature, Rotational speed, Torque, Tool wear
  + **Categorical Variable:** Machine Type (Encoded as 0, 1, 2)
  + **Binary Failure Modes:** TWF, HDF, PWF, OSF, RNF
  + **Target Variable:** Machine Failure (0 = No failure, 1 = Failure)
* **Preprocessing Steps:**
  + Encoded categorical variables using **Label Encoding**.
  + Scaled continuous variables using **StandardScaler**.
  + Dropped irrelevant columns such as ‘**UID’** and ‘**Product ID’**.

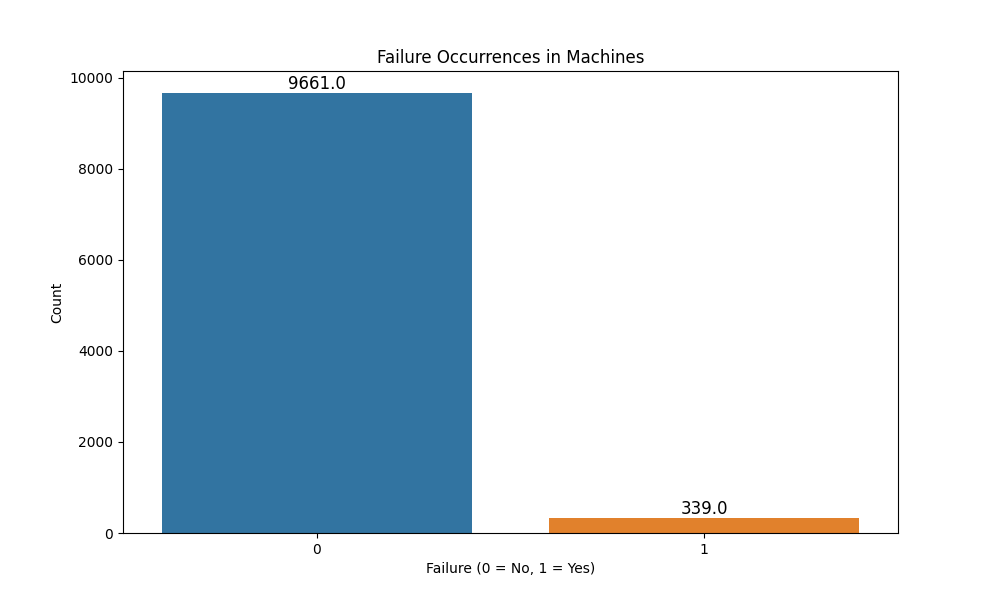
  
Above is the processed data for algorithmic predictions

Exploratory Data Analysis (EDA)

To gain insights into the dataset, we performed the following analyses:

  
**Sensor Readings Distribution (RPM):** Histogram plots for understanding variations in key sensor readings

  
**Feature Correlation Matrix**: Shows **correlation between sensor readings, machine types, and failure events.**

  
**Failure Distribution:** Visualized machine failure occurrences, indicating class imbalance.

Machine Learning Algorithms and Results

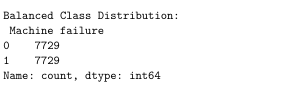
Considering the dataset, we see that there are data imbalances, however, when we are referring to our use case of Machine Failure Prediction imbalances are referred important as there are very less chances of the same. For a safer side, we have also considering balancing the dataset and perform the predictive analysis on the same. Please follow the below segments for the results.

1. Model Selection:  
  
Three different classification models were applied:

1. **Support Vector Machine (SVM)** – Tested for linear separability of classes
2. **Logistic Regression** – Used as a baseline model.
3. **Random Forest Classifier** – Best-performing model due to its robustness.

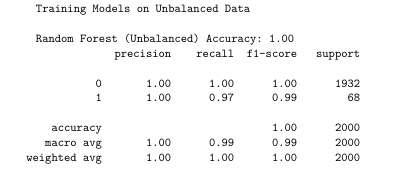
2. Data Balancing using SMOTE:

Used **Synthetic Minority Over-Sampling Technique (SMOTE)** to balance the classes in the training set. This is done to check the predictive capability of the algorithm.

  
Balancing the training dataset for analysis

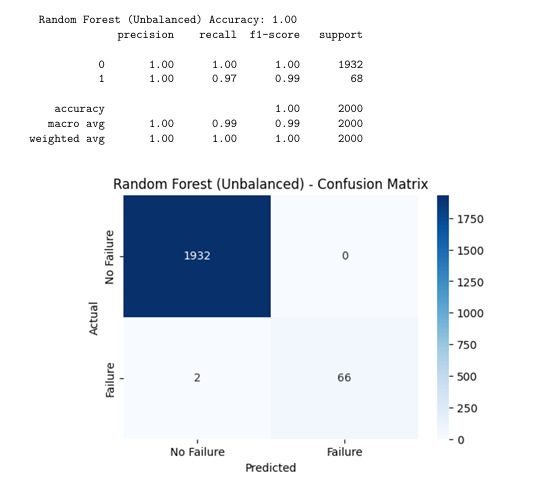
3. Model Evaluation Metrics

* **Accuracy:** Overall model correctness.
* **Confusion Matrix:** Showed correct and incorrect predictions.
* **Precision & Recall:** Evaluated false positives and false negatives.
* **F1-Score:** Balance between precision and recall



RESULTS:

The evaluation metrics performed better on the imbalanced dataset as estimated earlier, therefore, the random forest model was selected based on the following:

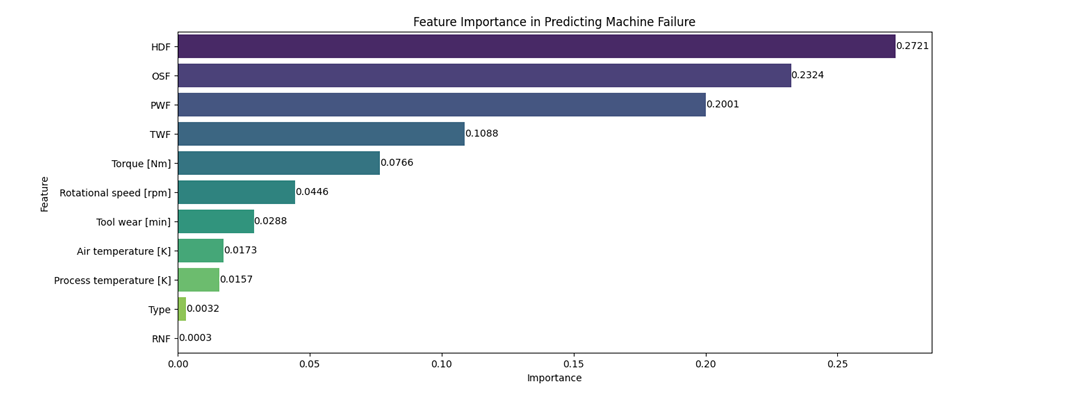


Feature Importance

To understand which factors contribute the most to machine failure, we analyzed feature importance using:

* **Random Forest Feature Importance Scores**
* **Findings:**

🔹 **Failure categories (HDF (Heat Dissipation), OSF (Overstrain Failure) , PWF (Power Failure) are the biggest risk factors.  
🔹 Torque and rotational speed must be controlled to prevent machine damage.  
🔹 Machine type has little impact—maintenance should focus on operational conditions.**



## **Deployment of the Project**

### **Saving the Trained Model:** Used ‘**Pickle’ library** to save the trained Random Forest model.

### ****Building a Web App using Streamlit****

* **User Inputs:** Users enter sensor values and failure indicators.
* **Model Prediction:** The trained model predicts failure probability in real-time.
* **Deployment Process:**
  + Created an interactive UI with **Streamlit**
  + Allowed users to input real-time sensor readings
  + Displayed the prediction (Failure / No Failure)
  + Hosted locally via **streamlit run app.py**

