**MAJOR PROJECT REPORT - DATA SCIENCE SEPTEMBER BATCH**

**Machine Failure Prediction using Sensor Data**

## **1. Introduction**

Machine maintenance and failure prediction are crucial to industrial productivity, as unexpected machine failures can lead to costly downtime. This project aims to predict potential machine failures in advance using a variety of sensor readings. By leveraging historical sensor data from machines, we can apply machine learning techniques to anticipate when a machine is likely to fail. This predictive capability allows for proactive maintenance and resource allocation, minimizing downtime and reducing costs.

The dataset used in this project contains sensor data collected from various machines, including information about environmental conditions, operating settings, and machine performance metrics.

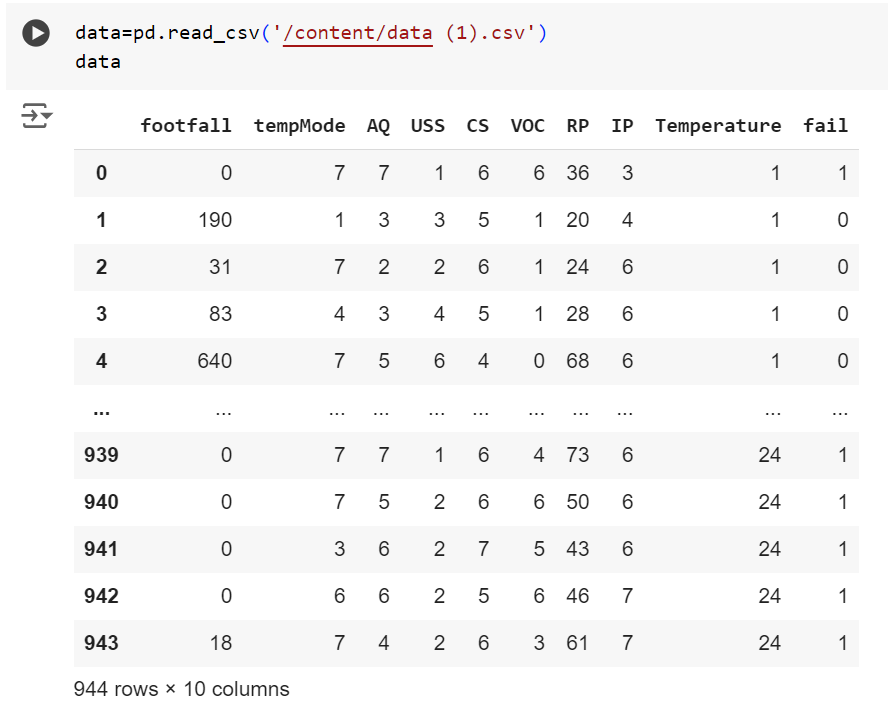
**2. Dataset Overview**

### **Dataset Description**

This dataset includes sensor readings and binary indicators of machine failures, collected to facilitate predictive maintenance. The main objective is to predict the occurrence of machine failures (fail column) based on the readings from other sensors.

### **Features in the Dataset**

* **footfall**: The number of people or objects passing by the machine.
* **temp Mode**: The temperature mode or setting of the machine.
* **AQ**: Air quality index near the machine.
* **USS**: Ultrasonic sensor data, indicating proximity measurements.
* **CS**: Current sensor readings, indicating the electrical current usage of the machine.
* **VOC**: Volatile organic compounds level detected near the machine.
* **RP**: Rotational position or RPM (revolutions per minute) of the machine parts.
* **IP**: Input pressure to the machine.
* **Temperature**: The operating temperature of the machine.
* **fail**: Binary indicator of machine failure (1 for failure, 0 for no failure).

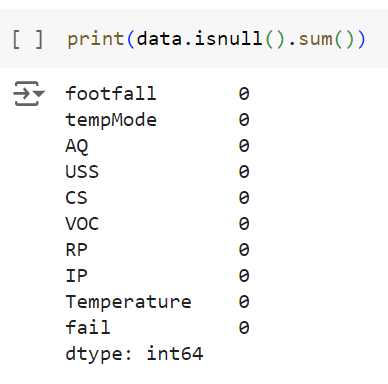


**3. Data Preprocessing**

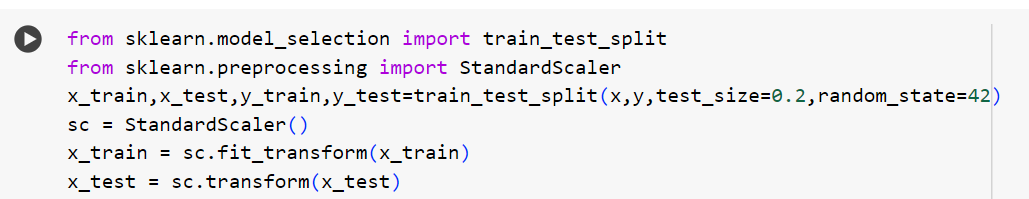
Data preprocessing is a critical step in machine learning projects as it ensures that the data is clean, consistent, and suitable for training a model. The steps we applied include:

### **Data Cleaning and Handling Missing Values**

1. **Handling Missing Values**: We checked for missing values in the dataset. But we don’t have any mising values here.



1. **Data Normalization**: Sensor data often varies in scale (e.g., temperature readings vs. air quality index). To ensure uniformity across features, we normalized or scaled the data. For instance, we used StandardScaler to transform each feature so that they contribute equally to the model, preventing any single feature from dominating the prediction.



### **Correlation Check:**

We performed a correlation check in this project to identify and remove features that are weakly correlated with the target variable (fail, which indicates machine failure). Here’s why this step is important:

### **1. Feature Selection:**

Correlation analysis helps us understand how strongly each feature is associated with the target variable. Features with low or no correlation are less likely to have predictive power, meaning they don’t provide meaningful information for predicting machine failures. By removing these features, we can focus on the variables that matter most, making the model simpler and often more accurate.

### **2. Reducing Noise:**

Features with low correlation can introduce noise into the model, making it harder for the algorithm to identify relevant patterns. This noise can lead to overfitting, where the model performs well on training data but poorly on new, unseen data. Removing such features helps reduce overfitting and improves the model's generalization.

### **3. Improving Model Efficiency:**

By removing irrelevant features, we also reduce the dimensionality of the data, which can speed up the training process and make the model more efficient. This is especially useful in complex models like Random Forest, where each additional feature can increase computation time and complexity.

### **How Correlation Check Works:**

During correlation analysis, we calculate the correlation coefficient between each feature and the target variable. If a feature has a very low correlation (close to zero), it means there's no linear relationship between that feature and the target. Such features are often considered irrelevant for the prediction task and are candidates for removal. Here the correlation of the columns tempMode and CS is very low.

In summary, performing a correlation check helps streamline the dataset by selecting only the most relevant features, which contributes to a more efficient, accurate, and interpretable model.

### **3. Column Removal**

During data preprocessing, we identified two columns that had a low correlation with the target variable (fail). Since the columns (‘tempMode’ & ‘CS’) did not provide significant predictive value, they were removed from the dataset to reduce noise and improve model performance. Feature selection is an important step in machine learning as it helps to remove irrelevant features that may introduce unnecessary complexity or reduce accuracy.



### **Feature Selection**

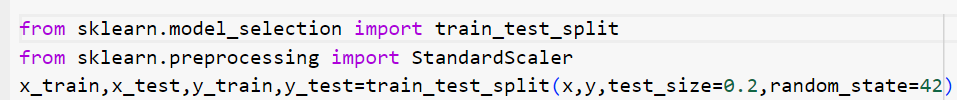
Based on correlation analysis, we removed two columns that had a poor correlation with the target variable. Features with low correlation may not add predictive value, and removing them can help improve model accuracy by reducing noise.

### **4. Encoding Categorical Variables**

If the dataset contained any categorical variables, we would apply encoding techniques (such as One-Hot Encoding) to transform these categorical features into a numerical format suitable for machine learning models. But we don’t have any categorical values here.

### **5. Data Splitting**

We divided the dataset into training and testing sets (e.g., 80% for training and 20% for testing). This separation allows us to train the model on one part of the data and evaluate it on an unseen portion to gauge its generalization performance.



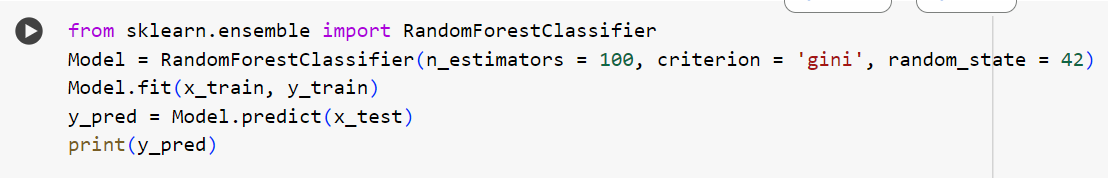
**Model Selection and Training**

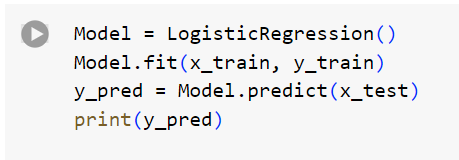
### **Choice of Model**

We initially experimented with two different models:

1. **Logistic Regression**: A simple and interpretable model that is often used for binary classification problems. It achieved an accuracy of 87.301% on this dataset.
2. **Random Forest Classifier**: An ensemble model that constructs multiple decision trees and combines their results to improve accuracy and reduce overfitting. Random Forest achieved a higher accuracy of 89.417%.

**Why Random Forest?** The Random Forest Classifier outperformed Logistic Regression in this case due to its ability to handle complex, non-linear relationships in the data. It also effectively deals with noise and irrelevant features, which can be common in sensor data. By aggregating multiple decision trees, Random Forest minimizes the risk of overfitting, providing more robust and reliable predictions for this dataset.



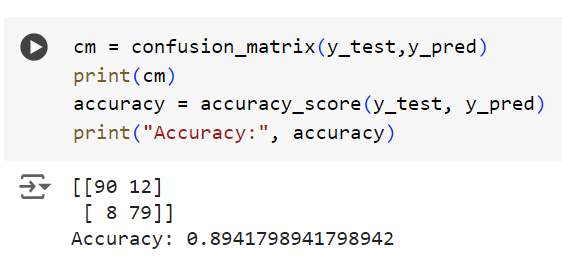


**5. Model Evaluation**

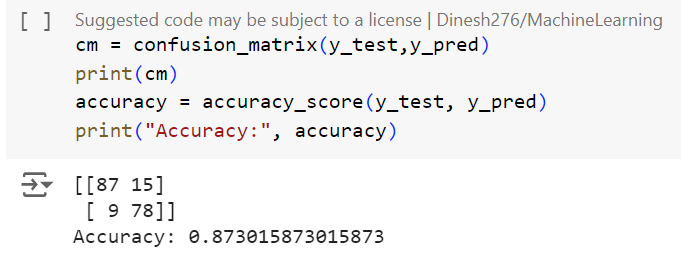
To evaluate the model's performance, we used the following metrics:

* **Accuracy**: Measures the percentage of correct predictions out of all predictions. Random Forest achieved 89.417% accuracy, which is a notable improvement over Logistic Regression.
* **Confusion Matrix**: Shows the true positives, true negatives, false positives, and false negatives, providing insights into model performance on each class.

**For Random Forest:**



**For Logistic Regression:**

****

Given that Random Forest outperformed Logistic Regression, we used it as the final model for this project.

**6. Conclusion**

In this project, we used sensor data to predict machine failures, aiming to facilitate predictive maintenance and reduce machine downtime. By preprocessing the data, removing irrelevant features, and experimenting with different models, we achieved an accuracy of 89.417% using a Random Forest Classifier. The model can serve as a valuable tool for industrial maintenance teams, helping them detect and address potential failures before they occur.