***Fault Classification and Remaining Useful Life (RUL) Prediction Using Machine Learning in Ball Bearings***

Notebook link: [Ingenium\_final.ipynb](https://colab.research.google.com/drive/1-tszm0n4v8MEe_JQhqqhPL1QMs6NBSYn?usp=sharing)

Dataset link: <https://www.kaggle.com/datasets/brjapon/cwru-bearing-datasets>

Fault classification and Remaining Useful Life (RUL) estimation using a dataset containing various statistical features extracted from sensor data.

The process involves preprocessing, training machine learning models, and evaluating their performance.

The dataset consists of multiple numerical features derived from sensor data, along with a categorical label indicating the type of fault.

Telemetry measurements are come from 3 acceleroments installed on 3 positions in the system:

* Drive end (DE)
* Fan end (FE)
* Base (BA)

The columns in the dataset include:

* max: Maximum value of the signal.
* min: Minimum value of the signal.
* mean: Mean value of the signal.
* sd: Standard deviation.
* rms: Root Mean Square value.
* skewness: Skewness of the signal distribution.
* kurtosis: Kurtosis measure of the signal distribution.
* crest: Crest factor.
* form: A form factor.
* fault: The categorical label representing different fault types (e.g., Ball\_007\_1).

Label Encoder maps the fault types into integers:

'Ball\_007\_1': 0,   
'Ball\_014\_1': 1,   
'Ball\_021\_1': 2,   
'IR\_007\_1': 3,   
'IR\_014\_1': 4,   
'IR\_021\_1': 5,   
'Normal\_1': 6,   
'OR\_007\_6\_1': 7,   
'OR\_014\_6\_1': 8,   
'OR\_021\_6\_1': 9

Ball\_007: Small defect size -> longer life  
Ball\_014: Medium defect size -> medium life  
Ball\_021: Large defect size -> shorter life

[43, 0, 0, 0, 0, 0, 0, 0, 3, 0] for Class 0 (Ball\_007\_1)

This indicates that the model correctly classified 43 samples as Ball\_007\_1, while 3 samples were misclassified as OR\_014\_6\_1.

Process involves:

**Data Preprocessing**

Loading the Dataset,

Feature Selection,

Label Encoding for Fault Classification,

RUL Mapping for Regression,

Feature Scaling,

**Machine Learning Models:**

**Fault Classification Using Random Forest**

Splitting Data for Classification,

Training a Random Forest Classifier,

Evaluating Classification Model,

**RUL Prediction Using Random Forest Regressor**

Splitting Data for Regression,

Training a Random Forest Regressor,

Evaluating Regression Model,

**Time-Series Analysis using ARIMA: Forecasting RMS Feature**

Preparing Data for ARIMA,

Fitting ARIMA Model,

Plotting Forecast Results.

DATA SIMULATION

* Using a Gaussian distribution, we generate new data points for each feature. The mean and standard deviation of each feature in the original dataset are used to define the parameters of the Gaussian distribution, ensuring that the synthetic data follows a similar statistical distribution.
* The newly generated data points are stored in a new DataFrame. This represents the simulated dataset, which has the same number of features as the original dataset but with newly generated values.

1. Random assigning of labels by random sampling

* For each numerical feature, we calculate its mean and standard deviation.
* Using a Gaussian (normal) distribution, we create new data points for each feature. The values are randomly drawn from a normal distribution with the same mean and standard deviation as the original dataset, ensuring the new data is statistically similar.

1. Gaussian Mixture Models (GMM) based labelling

* Extract numerical features (excluding labels).
* Train a GMM model on the original feature set to identify clusters that resemble the existing label structure.
* Generate new synthetic data using Gaussian sampling.
* Predict labels for synthetic data using the trained GMM model.

1. K-Means based labelling

* Extract numerical features (excluding labels).
* Train a K-Means model using the original dataset, with the number of clusters equal to the number of unique labels.
* Generate new synthetic data using Gaussian sampling.
* Predict clusters for synthetic data using K-Means.
* Map clusters to original labels based on the most frequent true label in each cluster.