**Relevant Problem - Project 1**

**AnomaData** **(Automated Anomaly Detection for Predictive Maintenance)**

**Problem Statement:**

Many different industries need predictive maintenance solutions to reduce risks and gain actionable insights through processing data from their equipment.

Although system failure is a very general issue that can occur in any machine, predicting the failure and taking steps to prevent such failure is most important for any machine or software application.

Predictive maintenance evaluates the condition of equipment by performing online monitoring. The goal is to perform maintenance before the equipment degrades or breaks down.

This Capstone project is aimed at predicting the machine breakdown by identifying the anomalies in the data.

The data we have contains about 18000+ rows collected over few days. The column ‘y’ contains the binary labels, with 1 denoting there is an anomaly. The rest of the columns are predictors.

**Steps**

* **Data Quality Check:**
  + Check for missing values.
  + Verify data types and correct them if needed (e.g., ensure time is a datetime type).
* **Descriptive Statistics and Visualizations:**
  + Load the data and convert the time column to datetime format.
  + Generate and print descriptive statistics for the dataset.
  + Visualize the distribution of the target variable y and selected features using histograms and KDE plots.
  + Identify outliers using boxplots for the selected features.
* **Data Cleaning:**
  + Handle missing values.
  + Address duplicate columns by removing the duplicate target column y.1.
  + Standardize the data (features) using StandardScaler to ensure they have zero mean and unit variance.
  + Recreate the cleaned dataframe with the standardized features along with the time and y columns.
* **Feature Engineering**:
  + Extract hour, day, and day of the week from the time column.
* **Standardize the Data**:
  + Standardize all features to have zero mean and unit variance.
* **Feature Selection**:
  + Use a Random Forest classifier to determine the importance of each feature.
  + Print the feature ranking based on importance.
  + Use Recursive Feature Elimination (RFE) to select the top 10 features based on their importance.
* This approach helps in identifying the most relevant features, reducing the dimensionality of the data, and potentially improving model performance.
* **Train/Test Split:**
  + Split the data into training and test sets using train\_test\_split with stratify=y to ensure the class distribution is maintained.
* **Model Training**: Train a Random Forest classifier on the training set.
* **Predictions**: Make predictions on the test set.
* **Confusion Matrix**: Compute and plot the confusion matrix.