METHODOLOGY AND EXPERIMENTS

1. ***Aim of Study***

The aim of this study is to develop and compare machine learning models for predicting machine failures in an industrial setting using the AI4I 2020 Predictive Maintenance (Machine Failures) Dataset. The goal is to identify the most effective model for predictive maintenance strategies.

2. ***Response (Dependent) and Independent Variables***

In this experiment, we distinguish between the dependent variable, which we aim to predict, and the independent variables, which serve as inputs to our predictive models.

Dependent Variable:

- Machine failure (binary: 0 for no failure, 1 for failure)

This is the target variable of our predictive model. It's a binary outcome indicating whether a machine failure occurred (1) or not (0). This variable encapsulates any of the five failure modes described in the dataset: tool wear failure (TWF), heat dissipation failure (HDF), power failure (PWF), overstrain failure (OSF), and random failures (RNF). The goal of our machine learning models is to accurately predict this outcome based on the independent variables.

Independent Variables:

- Type (categorical: L, M, H)
- Air temperature [K]
- Process temperature [K]
- Rotational speed [rpm]
- Torque [Nm]
- Tool wear [min]

These variables serve as the features or predictors in our model:

- **Type:** A categorical variable representing the product quality variant (Low, Medium, High). This impacts the tool wear rate and the threshold for overstrain failure.
- **Air temperature [K]:** The ambient temperature, which can affect machine operation and potential failures, particularly in relation to heat dissipation.
- **Process temperature [K]:** The temperature of the machining process itself, which is related to the air temperature but also influenced by the operation of the machine.
- **Rotational speed [rpm]:** The speed at which the machine's components are rotating. This is crucial for power calculations and can influence heat dissipation failures.
- **Torque [Nm]:** The rotational force applied by the machine. This is important for power calculations and overstrain failures.
- **Tool wear [min]:** The cumulative time the tool has been in use. This directly relates to tool wear failures and contributes to overstrain failures.

These independent variables were carefully selected based on their potential influence on machine failures. They encompass various aspects of the machine's operation and environment, providing a comprehensive set of predictors for our models.

3. ***Factors and Levels***

Factors to be considered in the study include:

In our experiment, the factors are the different algorithms being tested.

- Machine learning algorithms (LightGBM, CatBoost and XgBoost)
- Feature selection methods (Recursive feature elimination with cross-validation to select features)
- Hyperparameter tuning (levels: Default parameters, Grid search optimization, optuna)

4. ***Experiment and Design***

The experiment follows a systematic approach to assess the performance of different machine learning algorithms for predicting machine failures:

a. Data Preprocessing:

- Check for missing values in the dataset and handle them if necessary (e.g., imputation or removal)
- Encode the categorical 'Type' variable using one-hot encoding or label encoding
- Normalize numerical features using techniques like Min-Max scaling or Standard scaling to ensure all features are on the same scale

b. Feature Selection:

- Implement Recursive Feature Elimination with Cross-Validation (RFECV)
    - This method will iteratively remove features, build a model on the remaining features, and evaluate it using cross-validation
    - The process continues until the optimal number of features is found
- The selected features will be used consistently across all models for fair comparison

c. Model Training:

- Split the preprocessed data into training (70%) and testing (30%) sets
- Ensure the split maintains the proportion of failure/non-failure cases (stratified sampling)
- Train three models: LightGBM, CatBoost, and XGBoost

- These are all gradient boosting algorithms known for their high performance in various tasks

d. Hyperparameter Tuning:

- For each algorithm, perform hyperparameter optimization using:
    - **Grid Search:** Exhaustive search over a predefined parameter space
    - **Optuna:** A hyperparameter optimization framework that employs various samplers (e.g., Tree-structured Parzen Estimator) to efficiently search the parameter space

- Key hyperparameters to tune may include:
    - Number of estimators
    - Learning rate
    - Max depth
    - Minimum samples per leaf
    - L1 and L2 regularization terms

e. Cross-validation:

- Implement Recursive feature elimination with cross-validation to select features
- This ensures that the model's performance is consistent across different subsets of the data and helps prevent overfitting.

f. Model Evaluation:

- Evaluate each tuned model on the held-out test set
- Compare performance using the metrics outlined in the next section

5. ***Experiment Performance and Revision***

The experiment will be performed iteratively:

a. Initial run with all algorithms and default parameters

b. Analysis of preliminary results

c. Refinement of feature selection and hyperparameters based on initial performance

d. Re-run experiments with optimized settings

e. Final analysis and comparison of results

6. **Measuring Classifier Performance**

The following metrics will be used to evaluate and compare classifier performance:

a. Precision:

- Measures the proportion of correct positive predictions out of all positive predictions
- Important for minimizing false alarms in maintenance scheduling

b. Recall (Sensitivity):

- Measures the proportion of actual positives that were correctly identified
- Crucial in this context as it represents the ability to detect all potential failures

c. F1-score:

- The harmonic mean of precision and recall
- Provides a balanced measure of the model's performance

d. Area Under the Receiver Operating Characteristic curve (AUC-ROC):

- Represents the model's ability to distinguish between classes
- Less sensitive to imbalanced datasets

7. ***Algorithm Comparison and Selection***

The performance of different algorithms will be compared based on the metrics mentioned above. The selection of the best algorithm will consider:

a. Overall performance across all metrics
b. Consistency of performance (low variance in cross-validation)
c. Computational efficiency (training and prediction time)
d. Interpretability of the model (important for industrial applications)

The final selection will be made based on a weighted combination of these factors, with emphasis on recall and interpretability for practical implementation in an industrial maintenance setting.