**Anomaly Detection Using Sensor Data**

Overview: This document explains how to solve the Akasha318 competition task, which involves predicting anomalies using sensor data. The task is structured to evaluate data exploration, modeling techniques, and the ability to deliver a clean, effective Jupyter Notebook.

Objective: Predict whether each data record from the sensor readings is an anomaly or not. The final model will be evaluated using F1 Score and Accuracy.

Dataset Description:

train.parquet: Contains historical sensor readings with labels (anomaly or not).

test.parquet: Contains new, unlabeled sensor data for which predictions need to be made.

sample\_submission.parquet: Format of how the predictions should be submitted.

Step-by-Step Process:

1. Environment Setup

Install required libraries like pyarrow, scikit-learn, xgboost, lightgbm, catboost, etc.

2. Load Data Use pandas.read\_parquet() to load all three files into Pandas DataFrames.

3. Explore the Data

View the first few rows of the dataset.

Check for missing values and datatypes.

Identify the target column (usually named 'target' or 'anomaly').

4. Handle Missing Data

Use SimpleImputer to fill missing values using mean or median strategies.

5. Scale the Features

Normalize the feature values using StandardScaler to prepare them for ML models.

6. Split the Data

Split the training dataset into train and validation sets using train\_test\_split.

7. Build Models

Apply both classical (Logistic Regression, Decision Tree, KNN, SVM) and advanced models (Random Forest, XGBoost, LightGBM, CatBoost).

Fit each model and compare their performance using F1 Score and Accuracy.

8. Tune the Best Model

Use GridSearchCV to tune hyperparameters of the best-performing model.

9. Predict on Test Data

Preprocess the test data using the same imputer and scaler.

Predict using the best model.

Save the predictions in the format given by sample\_submission.parquet.

10. Feature Importance (Optional)

Plot feature importances for better model interpretability.

11. Save the Submission File

Save the predictions to a CSV file (e.g., final\_submission.csv) for uploading.

Deliverables:

A clean, well-commented Jupyter Notebook

A CSV file with the final predictions

Evaluation Criteria:

20%: Data Exploration and Preprocessing

60%: Modeling (Classical + Advanced)

20%: Model Evaluation using F1 Score and Accuracy

Tips:

Focus on creativity in feature engineering.

Use cross-validation for more reliable model results.

Visualize insights clearly with plots.

Final Note: This project is a great opportunity to demonstrate your practical machine learning skills. Follow the steps, comment your code, and make your insights easy to understand.