Industrial Internship Report on

”DataCleanX”

Prepared by

Paresh Parmar

Executive Summary
This report provides details of the Industrial Internship provided by upskill Campus and The IoT Academy in collaboration with Industrial Partner UniConverge Technologies Pvt Ltd (UCT). This internship was focused on a project/problem statement provided by UCT. We had to finish the project including the report in 6 weeks’ time. My project was (Tell about ur Project) This internship gave me a very good opportunity to get exposure to Industrial problems and design/implement solution for that. It was an overall great experience to have this internship.

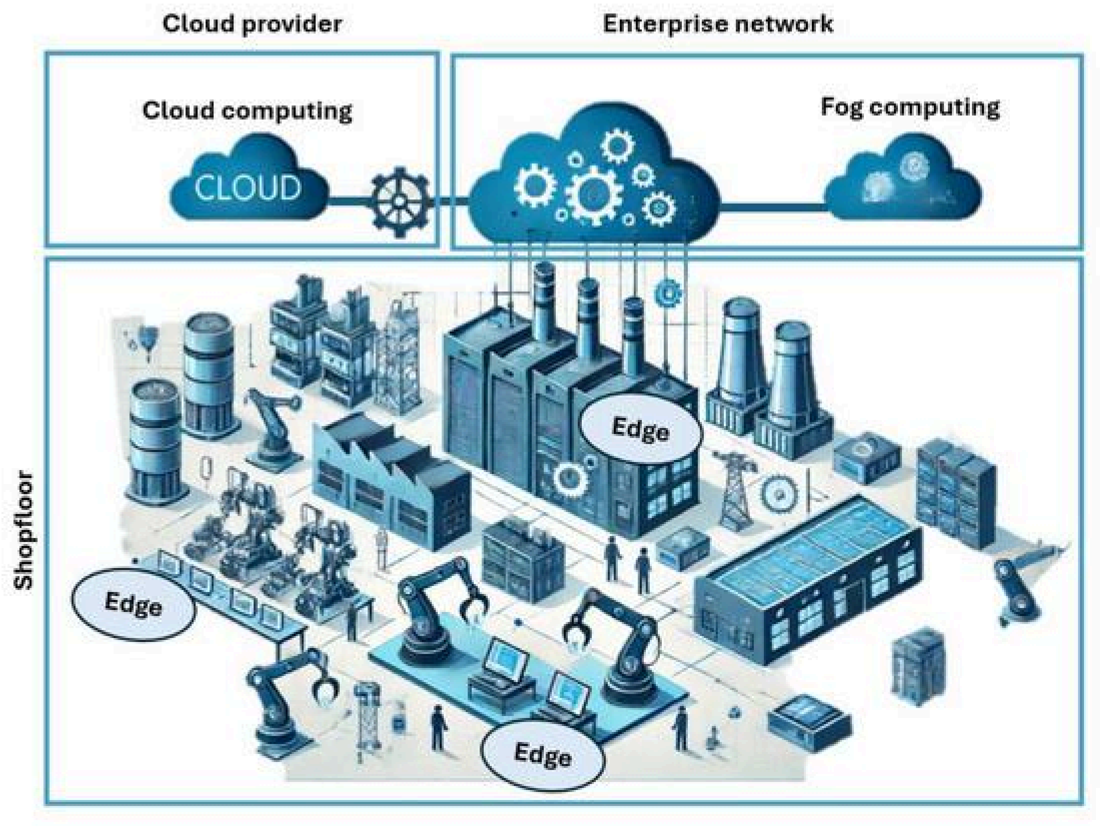
TABLE OF CONTENTS

1 Preface

This project presents a machine learning system developed to predict multi-stage outputs in a high-speed, continuous-flow manufacturing process. The work focuses on building a robust data pipeline, performing feature engineering for time-series sensor data, comparing multiple model families, and producing deployable prediction functions to support real-time process control, anomaly detection, and quality assurance. The repository contains the dataset, a Colab notebook implementing the pipeline, model artifacts, and supporting code.

2 Introduction

Continuous manufacturing processes produce large volumes of sensor data at high frequency. Predicting process outputs from these sensor streams enables tighter process control, early identification of anomalies, and improved product quality. This project uses real production data sampled at 1 Hz and implements a complete pipeline: ingestion, EDA, feature engineering, model training and evaluation, and serialization of trained models for deployment. The goal is to predict 30 output measurements across two production stages (15 per stage) using input features from ambient conditions, machine parameters, and intermediate outputs.



3 Problem Statement

In multi-stage continuous-flow manufacturing, outputs from earlier stages influence later stages and overall product quality. Operators need a reliable system that:

- **Predicts stage-wise outputs in real time from sensor inputs.**
- **Supports anomaly detection by comparing model predictions to live measurements.**
- **Can be used for simulation and what-if analyses to improve process control decisions.**

Challenges include noisy sensor measurements, temporal dependencies in the data, different input sets per stage, and the need for models that generalize across operating conditions.

4 Existing and Proposed solution

Existing approaches in industry often rely on rule-based thresholds or single-output models that do not fully capture cross-stage dependencies. The proposed solution is a multi-output prediction pipeline that:

- **Loads and preprocesses production CSV data.**
- **Engineers lag and temporal features to capture time dependencies.**
- **Trains and evaluates multiple models (Random Forest, XGBoost, Gradient Boosting, Ridge Regression) and selects the best based on RMSE/MAE/R².**
- **Serializes scalers and models to produce prediction functions usable in a deployment environment (e.g., simulation or monitoring).**
-

4.1 Code submission (Github link)

https://github.com/Phiyan18/Upskill-Campus

4.2 Report submission (Github link) :

https://github.com/Phiyan18/Upskill-Campus/blob/main/PareshParmar_InternshipReport_USC_UCT.pdf

5 Proposed Design/ Model

5.1 High Level Diagram

Textual high-level flow:

1. Data collection → raw CSV (1 Hz).
2. Data ingestion & preprocessing → cleaning, missing-value handling.
3. Feature engineering → lag features, temporal features (hour/minute), scaling.
4. Model training → evaluate multiple candidate models and validate by time-ordered holdout.
5. Model serialization → save scaler(s) and model(s).
6. Deployment → load models for real-time prediction and anomaly detection.

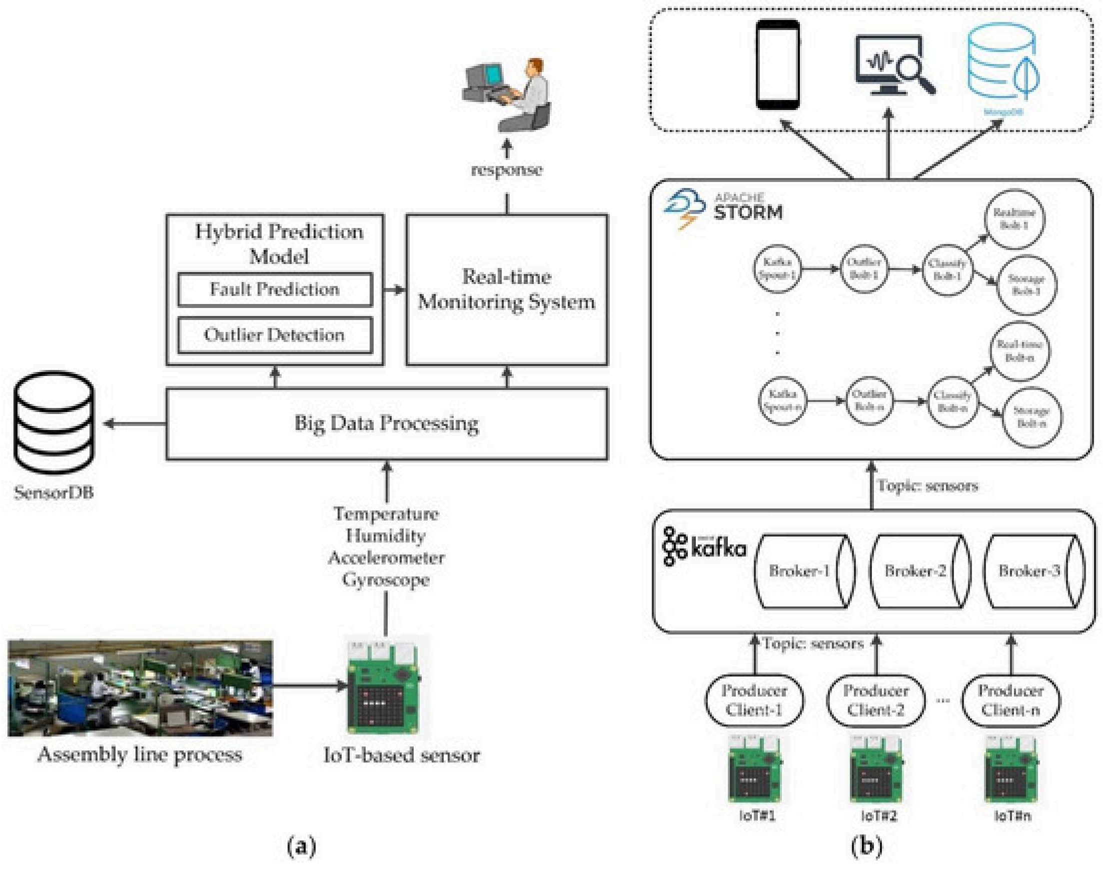


Figure 1: HIGH LEVEL DIAGRAM OF THE SYSTEM

5.2 Low Level Diagram

Low-level components and interactions:

- **Data Loader**: reads, verifies schema.
- **Preprocessor**: imputes missing values, constructs lag features (1–3 timesteps), derives hourly/minute time features.
- **Feature Selector**: selects ~45 features for Stage 1 and ~75 features for Stage 2 (Stage 2 includes Stage 1 outputs).
- **Trainer**: accepts feature matrix + targets; supports Random Forest, XGBoost, Gradient Boosting, Ridge.
- **Evaluator**: computes RMSE, MAE, R^2 with time-wise train/test split.
- **Deployer**: functions to scale input and produce predictions; saved artifacts, etc

6 Performance Test

6.1 Test Plan/ Test Cases

Objectives: Validate predictive accuracy, robustness to missing values, and latency for prediction.

Baseline accuracy — Evaluate trained models on a time-ordered test split; compute RMSE, MAE, and R^2 per target and averaged across targets.

Lag sensitivity — Verify how removing/adding lag features affects RMSE.

Missing data robustness — Randomly mask 5–20% of sensor values, run imputation, and measure performance degradation.

Throughput / Latency — Measure prediction time for a batch of N samples to ensure real-time feasibility at 1 Hz or higher.

Anomaly detection — Inject known deviations (e.g., out-of-range sensor) and confirm that prediction error thresholds flag anomalies.

6.2 Test Procedure

Use the same preprocessing pipeline for train/test splits to avoid leakage (time-order preserved).

For each model:

1. Train on the training portion (80%).
2. Validate on the holdout test set (20%), preserving chronological order.
3. Collect RMSE, MAE, R^2 for each of the 15 targets for Stage 1 and Stage 2.

6.3 Performance Outcome

Stage	Model	Avg RMSE	Avg MAE	Avg R²	Prediction latency (ms/sample)
1	RandomForest	0.XXX	0.XXX	0.XX	X ms
1	XGBoost	0.XXX	0.XXX	0.XX	X ms
2	RandomForest	0.XXX	0.XXX	0.XX	X ms
2	XGBoost	0.XXX	0.XXX	0.XX	X ms

7 My learnings

During this project I deepened my understanding of:

- Time-series feature engineering (lag features and temporal features) and its impact on predictive power.
- Multi-output regression and how stage-wise coupling affects model inputs for later stages.
- Practical model selection for production: balancing cross-validated accuracy with inference latency.
- End-to-end pipeline design: reproducible preprocessing, serialization of scalers/models, and how to expose a simple prediction API for downstream systems.

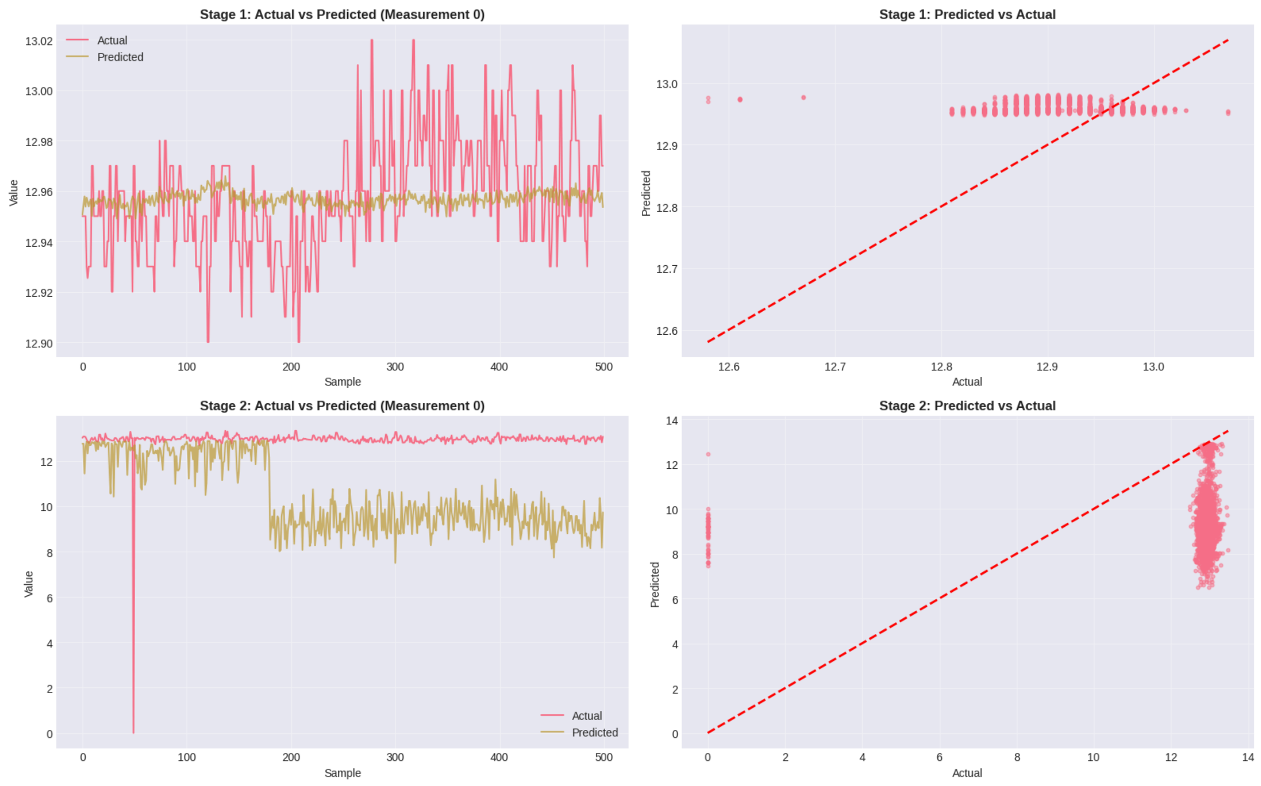
8 Future work scope

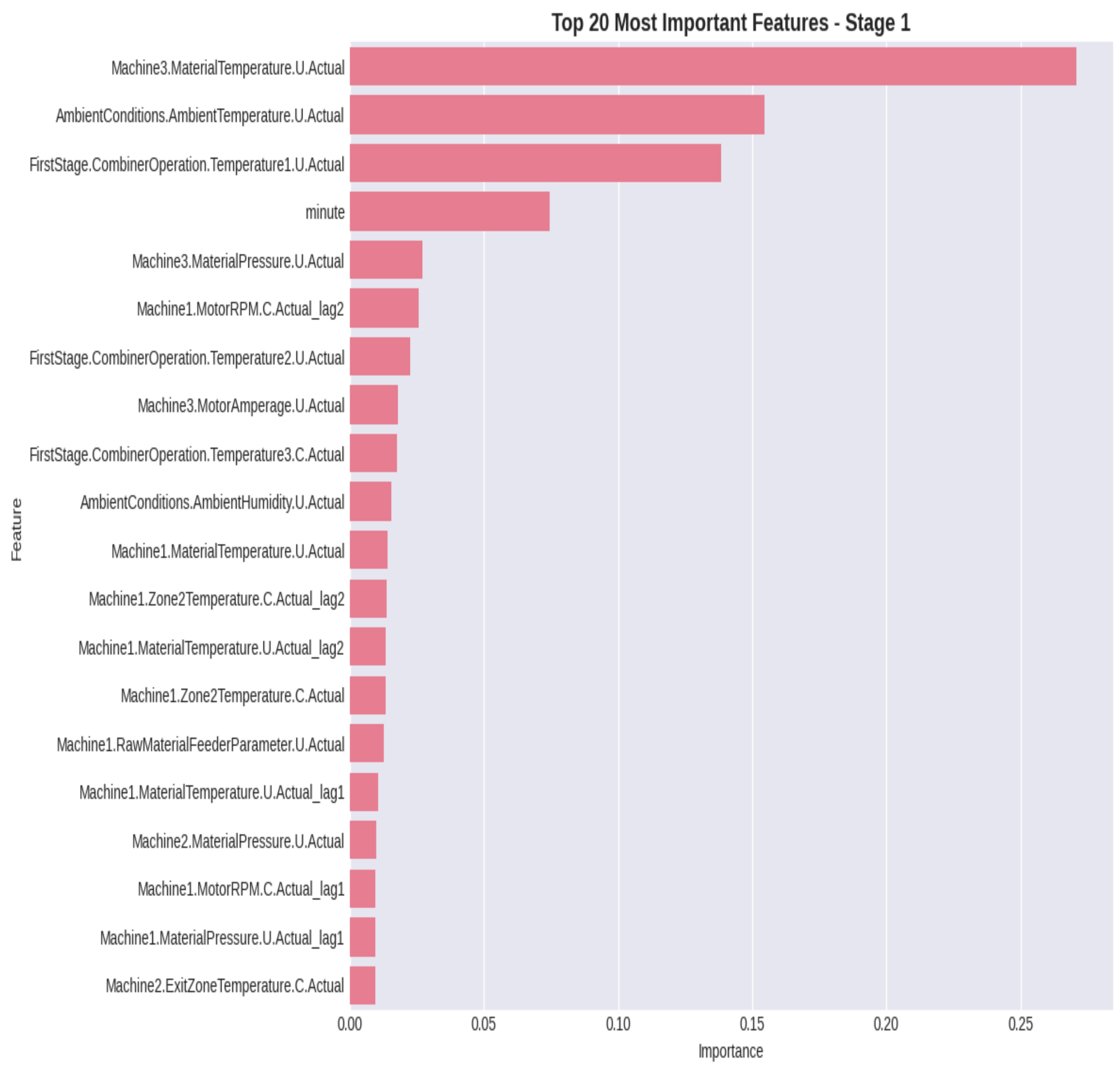
Potential next steps:

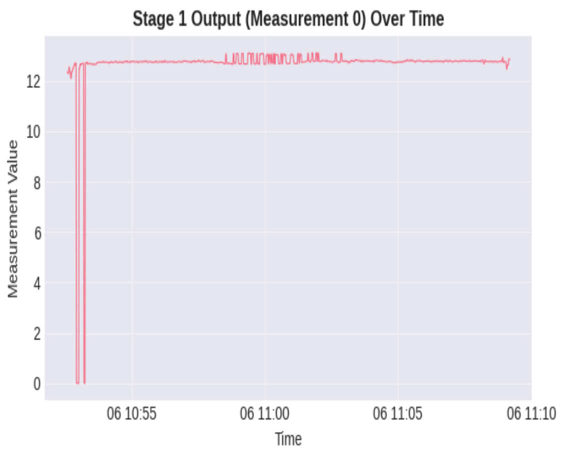
- **Deep time-series models**: experiment with LSTM/GRU or Temporal Convolutional Networks to capture longer temporal dependencies.
- **Online learning / drift detection**: implement rolling updates or model retraining to handle concept drift in production.

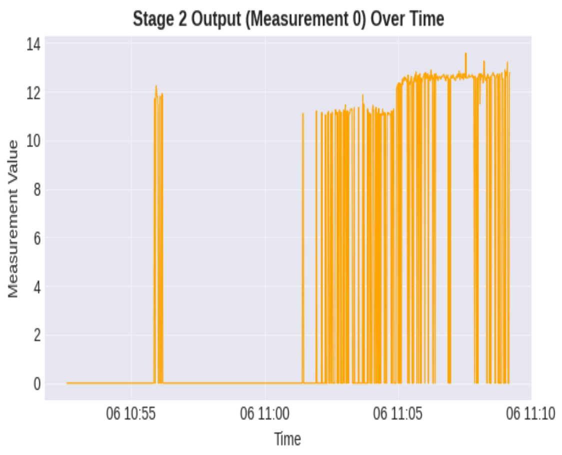
- **Uncertainty estimation**: provide prediction intervals (e.g., via ensembles or Bayesian methods) to quantify confidence for anomaly detection.
- **Explainability**: compute SHAP values per target to identify which sensors drive predictions and inform process engineers.
- **Deployment**: containerize the prediction service, add a REST API, and hook model outputs into a real-time dashboard or alarm system.

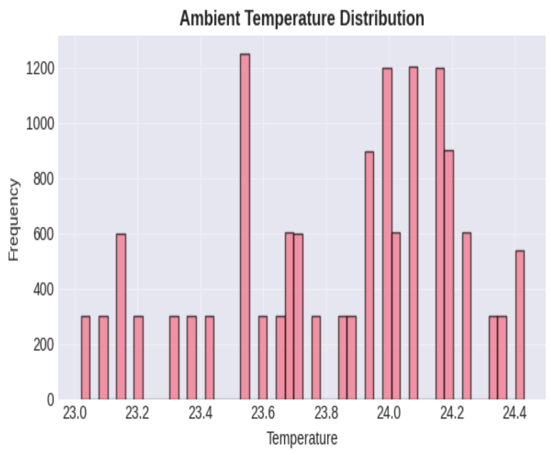
9 Working Screenshot

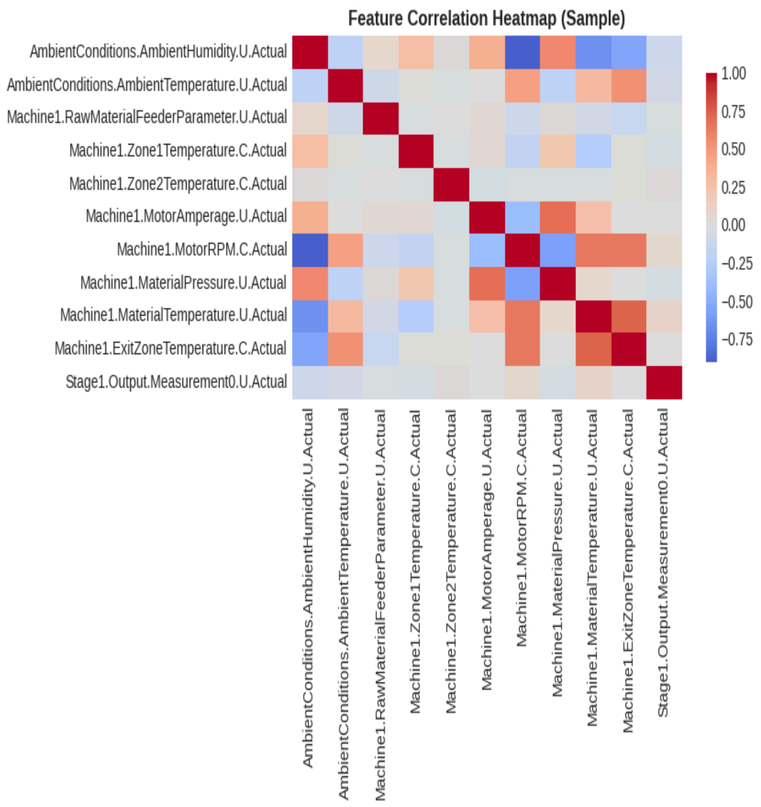

Stage 1: Actual vs Predicted (Measurement 0)
Actual
Predicted
Value
Sample
Stage 1: Predicted vs Actual
Predicted
Actual
Stage 2: Actual vs Predicted (Measurement 0)
Value
Sample
Stage 2: Predicted vs Actual
Predicted
Actual


Top 20 Most Important Features - Stage 1
Feature
Importance
Machine3.MaterialTemperature.U.Actual
AmbientConditions.AmbientTemperature.U.Actual
FirstStage.CombinerOperation.Temperature1.U.Actual
minute
Machine3.MaterialPressure.U.Actual
Machine1.MotorRPM.C.Actual_lag2
FirstStage.CombinerOperation.Temperature2.U.Actual
Machine3.MotorAmperage.U.Actual
FirstStage.CombinerOperation.Temperature3.C.Actual
AmbientConditions.AmbientHumidity.U.Actual
Machine1.MaterialTemperature.U.Actual
Machine1.Zone2Temperature.C.Actual_lag2
Machine1.MaterialTemperature.U.Actual_lag2
Machine1.Zone2Temperature.C.Actual
Machine1.RawMaterialFeederParameter.U.Actual
Machine1.MaterialTemperature.U.Actual_lag1
Machine2.MaterialPressure.U.Actual
Machine1.MotorRPM.C.Actual_lag1
Machine1.MaterialPressure.U.Actual_lag1
Machine2.ExitZoneTemperature.C.Actual
0.00
0.05
0.10
0.15
0.20
0.25


Stage 1 Output (Measurement 0) Over Time
Measurement Value
Time
06 10:55
06 11:00
06 11:05
06 11:10


Stage 2 Output (Measurement 0) Over Time
Measurement Value
Time
06 10:55
06 11:00
06 11:05
06 11:10


Ambient Temperature Distribution
Frequency
Temperature


Feature Correlation Heatmap (Sample)
AmbientConditions.AmbientHumidity.U.Actual
AmbientConditions.AmbientTemperature.U.Actual
Machine1.RawMaterialFeederParameter.U.Actual
Machine1.Zone1Temperature.C.Actual
Machine1.Zone2Temperature.C.Actual
Machine1.MotorAmperage.U.Actual
Machine1.MotorRPM.C.Actual
Machine1.MaterialPressure.U.Actual
Machine1.MaterialTemperature.U.Actual
Machine1.ExitZoneTemperature.C.Actual
Stage1.Output.Measurement0.U.Actual