Week 10 – User Interface and Visual Analytics

Introduction:

Real-time monitoring and dynamic production process adjustments are essential in contemporary smart manufacturing to preserve productivity and minimize downtime. The SmartFactory's last deployment and monitoring stage is represented by this script. AI project that provides thorough system performance insights by combining processed sensor data with schedule validation data. By offering dashboards, notifications, and feedback analysis to support decision-making, it seeks to support continuous operational optimization.

Objective:

The main objective is to put in place a framework for live system performance monitoring that combines sensor-based machine health indicators with task scheduling results. This structure ought to offer:

- Quantitative measures of machine operations and scheduling performance.
- Interactive, visual dashboards providing a daily rundown of operations.
- Automated notifications for unusual machine failures and major project delays.
- Analytical visualisations to comprehend trends in machine health and delay patterns.
- To improve continuous improvement, operator input on delay cause is incorporated.

Methodology:

- **Preparation and Data Loading:**
  - Processed sensor data from JSON files and loaded verified work schedules.
  - For time-series analysis, convert timestamps and pertinent information to the proper datetime forms.

- **Calculating Performance Metrics:**
  - Utilizing sensor failure flags, determine the total number of planned jobs, the average and maximum job delays, the percentage of delayed jobs, and the overall machine failure rate.

- **Building a Time Series Dashboard:**
    - Combine schedule information into daily metrics (planned jobs, average delay, percentage of delayed jobs).
    - Combine sensor data to determine the number of failures per day.
    - To get a single daily performance report, combine these datasets.
    - To visually monitor tasks, delays, and failures, create interactive multi-axis time series plots using Plotly.
- **Alert Generation:**
    - Establish cutoff points for failure counts (5 per day) and task delays (30 minutes).
    - Create notifications when jobs and days surpass failure and delay criteria, respectively.
    - For operational visibility, save notifications in JSON format.
- **In-depth Illustrations:**
    - Boxplots are used to examine the distribution of delays across work priorities and show how delays affect key activities.
    - To visualize machine health trends and spot failing equipment, roll the 7-day failure rates per machine.
- **Integration of Feedback Loops:**
    - For a subset of jobs, simulate operator feedback on the system, external, and unknown sources of delays.
    - To help direct next root cause analysis, visualize feedback proportions on a pie chart.
- **Report Exporting:**
    - For reporting and archiving purposes, save the performance measurements as CSV files.
    - Add calculated real start/end timings, delay durations, and work status to the final schedule and save it.

Results & Observations:

- **System Metrics:** Key performance metrics such as the overall work throughput, average delay of a few minutes, and the proportion of jobs that suffered delays were calculated by the script. Despite being relatively low, the machine failure rate offered vital context for operational stability.
- **Dashboard Insights:** Interactive charts showed temporal patterns, including failures, delays that spiked on particular days, and variations in the number of

planned jobs. This visual style facilitates the quick identification of irregularities or deterioration in performance.

- **Alerts:** Jobs with excessive delays and days with unusual failure spikes were successfully highlighted by alert generation. Proactive management actions before problems worsen are made easier by this feature.
- **Priority-Based Delay Distribution:** The visualization revealed that while higher priority projects often have lower delay medians, outliers may still occur, pointing to specific areas that need improvement.
- **Machine Health Trends:** Machines with rising failure rates were identified using rolling failure rate graphs, which indicated potential candidates for scheduling preventive maintenance.
- **Feedback Loop:** Despite using a fictitious dataset, the operator feedback simulation demonstrated how crucial it is to identify the reasons for delays in order to improve system modifications and take into account outside influences.
- **Final Schedule Augmentation:** To calculate actual delays and classify task status, actual start and finish timings were genuinely randomized within a range. Retraining machine learning models and further operational analysis are supported by the enhanced schedule dataset.

Summary:

This Week 10 script successfully integrates machine sensor feedback and work scheduling data to offer a complete monitoring solution for the smart manufacturing environment. Data-driven decision-making, prompt anomaly identification, and real-time operational monitoring are made possible by the integrated dashboards and alarm systems. Furthermore, even if operator input is only simulated, it represents a significant advance in bridging the gap between automated systems and human insights. Predictive maintenance planning and continuous performance improvement are firmly established by the final expanded dataset and reports.

Conclusion:

In an autonomous manufacturing environment, the deployment and monitoring system created here provides vital insight into the operation of workflows and the condition of equipment. This strategy improves the factory's resistance to delays and failures by integrating statistical measurements, interactive visualizations, automated alarms, and feedback loops. It illustrates the feasibility of combining real-time sensor data with AI-driven scheduling for ongoing operational improvement. By adding sophisticated anomaly

detection, predictive failure predictions, and adaptive scheduling adjustments based on real-time input, future research may build upon this foundation.

URL To Week 10 – User Interface and Visual Analytics.ipynb