Week 7 – Workflow Optimization and Scheduling

Introduction:

Scheduling in smart manufacturing involves more than just allocating tasks; it also involves dynamically optimizing them in response to changing machine circumstances. Using real-time health data, such failure likelihood and urgency, to intelligently reschedule workloads among machines is the main emphasis of week seven. At this point, proactive operational control replaces passive monitoring (Week 6).

Objective:

- Make judgments about task scheduling based on health indicators received from sensors.
- Create a rescheduling algorithm that takes urgency and failure levels into account.
- Use clear charts and Gantt plots to see delay patterns and the results of task rescheduling.
- In Week 8, provide a task schedule that is optimized and legible by humans.

Methodology:

- **Ingestion and Preparation of Data:**
  - The script loads the Week 2 output, processed_sensor_data.json, and sorts it by timestamp and machine_id.
  - In order to facilitate chronological work tracking, timestamps are aligned and processed.
- **Simulation of Failure Probability:**
  - A faux failure score is produced since true failure prediction has not yet been modeled:
    - Give rows with failure == 1 a score in the range of 0.85 to 1.0.
    - Assign between 0.0 and 0.6 for records that are in good health.
  - The behavior of a trained predictive maintenance model is modeled by this probabilistic labeling.
- **Engineering Urgency Scores:**
  - MinMaxScaler is used to normalize all six sensor measurements.
  - The formula for an urgency score is:
    (failure_probability + vibration + temperature) / 3

- This serves as a composite risk score for scheduling choices and assigns greater weight to mechanical stress indications.
- **Simulating a job queue:**
  - Every 20th timestamp is used to simulate jobs for each machine, which correspond to production line task allocations.
  - Job_id, machine_id, scheduled_time, randomized_estimated_duration, and priority_level (Low/Medium/High) are all included for each work.
  - The queue simulates actual task scheduling with fluctuating demands.
- **Algorithm for Intelligent Rescheduling:**
  Intelligent_scheduler() defines:
  - It returns the most recent machine state for every task (timestamp ≤ job's planned time).
  - Failure probability > 0.6 results in a 60-minute delay, whereas urgency score > 0.5 results in a 30-minute delay.
  - produces a table for rescheduling that includes:
    - New versus original time
    - Risk of failure and urgency score
    - Duration and delay indicators

  High-risk computers are dynamically deprioritized thanks to this method.

Results & Insights:

- **Analysis of Delay Distribution:**
  - The failure probability histogram exhibits a distinct tail above 0.6, which is in good agreement with the risk threshold.
  - A priority vs. delay countplot shows that:
    - Medium/High priority projects have higher delays since they are closer to important risk areas.
    - Rarely are low-priority tasks postponed unless there are significant dangers involved.
- **Gantt chart that is interactive:**
  - A visual work execution timeline is created using Plotly's Gantt chart:
    - Jobs that are on time or delayed are categorized by priority.
    - Enables visual inspection of bottlenecks, overlaps, or idle periods by manufacturing management.
- **Exported Outputs:**
  - The optimized_schedule.json file contains the final optimized schedule.
  - Every record includes:

- Times that have been rescheduled
- Flags of delay
- Scores of risk
- Priority metadata

Week 8 will see the direct integration of this organized format with a production management system.

Summary:

By enabling the factory to modify its work queues in response to the health of live machines, Week 7 brings about a significant improvement in system intelligence. Important lessons learned:

- Scheduling no longer follows set plans but instead responds to predicted insights.
- Prioritization honors technical (risk measurements) and commercial (priority levels) limitations.
- Both operational execution and diagnostic exploration are supported by visualization tools.

Conclusion:

By Week 7, SmartFactory.AI has advanced past detection and is able to intelligently control activities. A real-time work queue and delay modeling system helps manufacturers reduce cascade failures, increase uptime, and move operations away from high-risk equipment in advance. The framework for Week 8's system integration and orchestration logic, which connects scheduling and monitoring to a closed-loop execution environment, is now established.

URL to Week 7 – Workflow Optimization and Scheduling.ipynb