Decisions:

1. Identify machine parts at high risk of failure.
2. Prioritize maintenance schedules to avoid breakdowns.
3. Reduce unplanned downtimes and increase efficiency.
4. Allocate resources effectively for proactive maintenance.

ML Tasks:

1. Classification → Predict part failure (Yes/No).
2. Regression → Predict time until failure.
3. Anomaly Detection → Detect unusual sensor readings

Value Propositions:

1. A predictive maintenance system to detect failing machine parts in advance.
2. Reduces machine downtime and maintenance costs.
3. Prevents unexpected failures that disrupt operations.
4. Improves overall factory efficiency and reliability.
5. Factory operators to schedule maintenance.
6. Maintenance engineers to prevent failures.
7. Production managers to optimize machine uptime.
8. Manufacturing companies (reduced costs, higher efficiency).
9. Factory workers (safer working conditions).
10. Customers (more reliable product delivery).

Data Source:

1. IoT sensors (temperature, vibration, pressure).
2. Machine maintenance records.
3. Operator logs (manual inspections, past failures).
4. Usage history (hours operated, workload).

Data Collection:

1. Real-time IoT sensors → Automated data collection.
2. Manual inspection logs → Technician inputs.
3. Historical maintenance reports → Past failure analysis.
4. Feedback loop from predictions → Model updates based on actual outcomes.

Offline Evaluation:

1. Train models using historical data.
2. Cross-validation to prevent overfitting.
3. Compare different algorithms (Random Forest, XGBoost, LSTMs).
4. Classification → Precision, Recall, F1-score.
5. Regression → RMSE (Root Mean Square Error), MAE (Mean Absolute Error).
6. Anomaly Detection → ROC-AUC, False Positive Rate.

Features:

1. Temperature fluctuations (indicate overheating issues).
2. Vibration intensity (detects mechanical wear and tear).
3. Previous failure frequency (parts with recurring issues).
4. Machine usage hours (wear and tear over time).
5. Pressure and load variations (detects stress levels on parts).

Building Models:

Model Updates:

1. Initial training on historical data.
2. Regular retraining every month using new failure records.
3. Continuous learning with real-time data streams.
4. Automated model tuning based on recent failure patterns.

Models Used:

1. Random Forest, XGBoost, Neural Networks for classification.
2. LSTMs, ARIMA for time-series failure prediction.
3. Deep Learning for anomaly detection (detecting unusual sensor readings).

Making Predictions:

1. Continuously in real-time using IoT sensor data.
2. Scheduled at regular intervals for preventive maintenance.
3. Before critical failures occur to provide early warnings.
4. Downtime tolerance varies based on machine importance (e.g., critical vs. non-critical parts).

Evaluation and Monitoring:

Methods: Monitor real-time predictions and update models periodically.

1. Compare predicted failures vs. actual failures to measure accuracy.
2. Set up automated alerts for maintenance teams.

Metrics: Model drift detection (does accuracy degrade over time?).

1. Precision, Recall, RMSE updates based on new failure reports.
2. Real-time anomaly tracking for early detection.