



Manufacturing Downtime Project Documentation

1. Executive Summary

A comprehensive project package for analyzing and forecasting machine downtime in a manufacturing plant. The package includes: data cleaning and preprocessing, a star-schema data model, exploratory analyses, predictive models (time-series and classification), Power BI dashboards, wireframes, visuals, step-by-step tutorials, and formal reports for stakeholders.

Key outcomes:

- Cleaned machine_downtime_schema_v.1.2.xlsx ready for analytics.
- Star-schema data model [one Fact table + six Dimension tables].
- Exploratory analyses answering key business questions [downtime counts by machine, month, sensor-statistics before/during failures].
- Predictive pipelines: RandomForest for day-ahead failure counts [regression], Logistic Regression & classification for next-day failure probability.
- Power BI dashboards: Overview, Machine Performance, Data Explorer with KPI tiles, treemaps, matrices and drillable visuals.

2. Project Scope & Objectives

Scope: Clean raw machine sensor logs and downtime flags; build analytics and models to forecast downtime and produce interactive dashboards for maintenance planning.

Objectives:

1. Clean and standardize raw data.
2. Build a usable star-schema data model for fast reporting.
3. Answer business questions [downtime by machine, time patterns, sensor behavior around failures].
4. Build forecasting and classification models for near-term downtime risk.
5. Deliver Power BI dashboards and explanatory reports with recommendations.

3. Deliverables

1. machine_downtime_schema_v.1.2.xlsx

2. Jupyter Notebooks:
 - 00_data_cleaning.ipynb [pandas cleaning steps]
 - 01_exploratory_analysis.ipynb [visual EDA]
 - 02_forecasting_analysis.ipynb [forecasting models]
3. Power BI file: Downtime_Dashboard.pbix with Overview, Machine Performance, Data Explorer pages.
4. Documentation: this master PDF + presentation slides.

4. Data Cleaning & Preprocessing [Design & Steps]

Steps implemented

- Import libraries: pandas, numpy, matplotlib, sklearn.
- Load CSV/Excel and inspect: .head(), .info(), .describe(), .shape, .columns.
- Missing values:
 - Numeric → fill with median.
 - Categorical/text → fill with mode.
- Duplicates → drop_duplicates().
- Trim text columns and standardize case [.str.strip(), .str.title()].
- Convert date column to datetime and extract year, month, day.
- Create Date_Dim with DateKey, Year, Quarter, Month, Day, WeekOfYear.
- Outlier detection [IQR] + iterative replacement by median until none remain.

5. Data Model [Star Schema]

Fact Table: Fact_Table

- Keys: Date, Machine_ID, Spindle_Temp_Status_ID, Spindle_Status_ID, Cutting_Status_ID, Hydraulic_Status_ID
- Measures: Hydraulic_Pressure[bar], Spindle_Bearing_Temperature, Spindle_Vibration, Cutting[kN]

Dimension Tables [6]:

1. Date_Dim — Date[PK], Year, Month, Day
2. Machine_Dim — Machine_ID [PK], Assembly_Line_No
3. Hydraulic_Dim — Hydraulic_Status_ID [PK], Hydraulic_Status
4. Spindle_Temp_Dim — Spindle_Temp_Status_ID [PK], Spindle_Temp_Status
5. Spindle_Dim — Spindle_Status_ID [PK], Spindle_Status
6. Cutting_Dim — Cutting_Status_ID [PK], Cutting_Status

6. Wireframes & Visual Design [Power BI]

6.1 Overview Page [Header + KPIs + Trend]

- Top: Project title badge, date slicer, Assembly Line dropdown.
- Left KPI column [vertical]: KPI cards—Avg Tool Vibration, Avg Voltage, Avg Torque, Avg RPM, Downtime %.

- Middle: Area chart Number_of_Failures_by_Month (drillable by machine/line).
- Right: Treemap Count of Downtime by Month + small table Machine Health (Summary).
- Footer: Last data refresh timestamp and quick filters.

6.2 Machine Performance Page

- Top filter bar (Machine selector, Date range).
- Left: KPI tiles repeated for selected machine.
- Middle: Bar chart Downtime Count by Hydraulic_Status - Spindle_Status with drill up.
- Right: Matrix: Month vs Downtime count heatmap style.
- Bottom: Table Per Machine Breakdown with conditional formatting for high downtime.

6.3 Data Explorer Page

- Search box for Machine_ID, table view, and slicers for status categories.
- A small chart to preview sensor distributions and selectable rows to inspect raw records.

Theme & Styling: crimson/dark red + cream gradient, rounded cards, consistent fonts, shadow depth for KPI tiles.

7. Analytics Questions & Implementation Map

1. Total downtime per machine — SQL aggregation: `SELECT Machine_ID, SUM[Downtime_Flag] FROM fact GROUP BY Machine_ID` → bar chart in Power BI.
2. Which machine has highest downtime — same as [1] + highlight/highlight color in visual.
3. Average sensor values before vs during downtime — SQL window functions or grouped joins; implement comparisons and boxplots in EDA notebook.
4. Which month has highest downtime — group by `MONTH(DateKey)` → area/bar chart.
5. Downtime trend over months — time-series aggregation + trend line (linear fit) in notebook and line visual in Power BI.
6. Do newer machines have less downtime — join `Machine_Dim.Install_Date` to compute age and compare downtime rate by age groups.

8. Modeling & Machine Learning (Pipeline Summaries)

8.1 Regression: Daily Failures Forecast

- Data: daily aggregation of Downtime (failures per day).
- Features: Lag_1, Lag_2, Lag_3.
- Model: `RandomForestRegressor(n_estimators=200)`.
- Split: time-ordered (train first 80%, test last 20%).
- Metrics: MAE, R2.
- Output: next_day forecast and actual vs predicted plot.

8.2 Classification: Next-Day Failure per Machine

- Label: Next_Day_Downtime created by shifting Downtime_Flag per machine.

- **Features:** recent sensor readings + operational params (list in v3).
- **Models:** LogisticRegression, KNN (for comparison).
- **Split:** stratify by label if class balance is imbalanced.
- **Metrics:** Accuracy, Precision, Recall, F1-score; ROC-AUC.
- **Output:** risk table with Failure_Probability per machine and bar chart.

8.3 KNN Regression: Weekly Machine Predictions

- **Data:** weekly failures per machine.
- **Encoding:** LabelEncoder for machine IDs.
- **Model:** KNeighborsRegressor(n_neighbors=3).
- **Output:** predicted failures next week for each machine and bar visualization.

9. Tutorials & How-to Guides

Each tutorial corresponds to a deliverable notebook and includes commands and step-by-step notes.

9.1 Notebook: Data Cleaning [00_data_cleaning.ipynb]

- Load dataset
- Run the exact cleaning pipeline (missing value imputation, outlier treatment IQR loop, trimming, type conversion)
- Save cleaned CSV

9.2 Notebook: EDA & Charts [01_exploratory_analysis.ipynb]

- Load cleaned CSV
- Produce: downtime counts by machine, by month, boxplots of sensors pre/post downtime, correlation heatmap
- Save charts as PNGs for Power BI and report

9.3 Notebook: Forecasting [02_forecasting.ipynb]

- Create daily aggregation, lag features, train RandomForest, evaluate and produce next-day forecast

9.4 Notebook: Classification [02_forecasting.ipynb]

- Feature selection, train LogisticRegression and KNN, evaluation, produce risk table (latest record prediction)

9.5 Notebook: Downtime analysis & forecasting [Analysis_Questions.sql].

- **Analysis_Questions**
Load SQL exports, create daily aggregations and machine-level features from the Fact_Table\$, build lag / rolling features (1–7 days), split train/validation by time, train a RandomForest classifier/regressor to predict next-day Machine_Failure (or next-day

downtime count], evaluate [confusion matrix / ROC-AUC or RMSE / MAE] and produce next-day forecasts [table + CSV].

10. Timeline & Milestones

- **Week 1** — Data cleaning, model build, deliver cleaned dataset and data model
- **Week 2** — EDA, answer business questions, design Power BI wireframes
- **Week 3** — Train models, validate, build predictions pipeline
- **Week 4** — Build Power BI dashboards, final report, and presentation