This assignment is a two-part challenge designed to evaluate your capabilities across a full-stack data science workflow.

1. **Task 1: Machine Data Analysis** – Analyze 3 years of cyclone machine sensor data to detect shutdowns, operational states, anomalies, and perform short-term forecasting.
2. **Task 2: RAG + LLM System Design** – Design and prototype a Retrieval-Augmented Generation (RAG) system to query technical documents using open-source models.

You are expected to complete both parts to the best of your ability. The focus of the evaluation is on your problem-solving process, design decisions, and the ability to produce a clean, well-documented solution.

# Part 1: Machine Data Analysis (Timeseries)

## Dataset Provided: A 3-year time-series dataset of cyclone sensor data, comprising approximately 370,000 records at a 5-minute interval.

## Variables:

* Cyclone\_Inlet\_Gas\_Temp – Temperature of hot gas at the cyclone inlet.
* Cyclone\_Gas\_Outlet\_Temp – Temperature of hot gas at the cyclone outlet.
* Cyclone\_Outlet\_Gas\_draft – Gas draft (pressure) at the cyclone outlet.
* Cyclone\_cone\_draft – Gas draft (pressure) at the cyclone cone section.
* Cyclone\_Inlet\_Draft – Gas draft (pressure) at the cyclone inlet.
* Cyclone\_Material\_Temp – Temperature of the material at the cyclone outlet.

## Tasks (detailed)

### 1) Data Preparation & Exploratory Analysis

* Load dataset; handle missing values, timestamp gaps, and outliers. Ensure strict 5-minute indexing.
* Provide summary statistics and correlation matrix.
* Visualize representative slices (e.g., one week and one year) to show normal behavior and variance.

### 2) Shutdown / Idle Period Detection

* Programmatically detect and segment **shutdown or idle periods**.
* Compute total downtime and number of shutdown events across 3 years.
* Provide a visualization of one full year with shutdowns clearly highlighted.

### 3) Machine State Segmentation (Clustering)

* Exclude shutdown periods; use only active operation data for clustering.
* Apply multivariate clustering (examples: KMeans, DBSCAN, HDBSCAN) on relevant features/aggregates (raw variables, lags, deltas, rolling stats).
* Determine an interpretable set of states (e.g., Normal, Startup/Shutdown, High Load, Degraded).
* For each cluster/state, provide:
  + Summary statistics (means, std, relevant percentiles)
  + Typical behavior / short description (e.g., “High Inlet Temp, moderate draft → High Load”)
  + Frequency and duration statistics (how often and how long each state persists)

### 4) Contextual Anomaly Detection + Root Cause Analysis

* Build anomaly detection that is **contextual to cluster/state**:
  + Example approach: for each cluster, train a cluster-specific Isolation Forest / LOF or use state-conditional statistical thresholds (rolling MAD) and flag anomalies relative to that cluster’s distribution.
* Produce a consolidated list of anomalous events (start, end, duration, most implicated variables).
* For **each selected anomaly** (choose 3–6 interesting ones), propose a **root cause hypothesis** based on data evidence. For example:  
  + “A sudden drop in Cyclone\_Gas\_Outlet\_Temp coincided with a spike in Cyclone\_Inlet\_Draft → likely upstream draft surge or partial inlet blockage clearing.”
  + Explain the causal reasoning (timing, variable relationships, cluster state before/after).
* Visualize each chosen anomaly with context (time series plots, cluster labels, and additional annotations).

### 5) Short-Horizon Forecasting (Required)

* Forecast Cyclone\_Inlet\_Gas\_Temp for the next **1 hour (12 steps)**.
* Compare at least two methods (e.g., persistence baseline + one model such as ARIMA/Prophet or a simple LSTM/RandomForest regressor on lag features).
* Evaluate on a held-out test split using **RMSE** and **MAE** (report both).
* Discuss performance differences and practical forecasting challenges (shutdowns, regime changes, non-stationarity).

### 6) Insights & Storytelling

* Provide **3–5 concise insights** connecting shutdowns, clusters, anomalies, and forecasting results.  
  + Example: “Cluster X shows higher anomaly rate and lower efficiency metrics; many anomalies in Cluster X occur 10–30 minutes before shutdowns.”
* Give actionable recommendations (e.g., monitoring rules, triggers for operator alerts, further data collection).

### 

### 

### Deliverables (Task 1)

All outputs for **Task 1** should be placed inside a folder named **Task1**. The folder must contain:

1. **Code** – Jupyter Notebook (task1\_analysis.ipynb) or Python script (task1\_analysis.py), runnable with clear steps.
2. **Outputs (CSV)**
   * shutdown\_periods.csv – Detected shutdowns (start, end, duration).
   * anomalous\_periods.csv – Anomalies with metadata (time, duration, variables, state).
   * clusters\_summary.csv – Summary statistics for each operational state.
   * forecasts.csv – True vs predicted values for forecasting tasks.
3. **Plots** (saved as PNG inside plots/)
4. **README.md** – Instructions on running the code and description of deliverables.

# Part 2: RAG + LLM System Design

## Context - The operations team needs a system to query 50+ PDFs of technical documentation (operating manuals, SOPs, troubleshooting guides) in natural language (e.g., *“What does a sudden draft drop indicate?”*). The system should return reliable, cited answers and be scalable, robust, and resistant to common LLM failure modes.

## Objective - Design and prototype a Retrieval-Augmented Generation (RAG) system that enables efficient document search and LLM-based answering with guardrails.

## **Corpus:** Use the documents located in the following Google Drive folder:<https://drive.google.com/drive/folders/1JVpmDQNrRcq3GByOvsPaapsx2Im6YbMT?usp=drive_link>

## Tasks

### 1. System Architecture

* Propose a complete architecture for the RAG system.
* Include components for:
  + Document ingestion & preprocessing
  + Chunking strategy
  + Embeddings & indexing (vector database)
  + Retrieval layer
  + LLM layer for answer generation
  + Guardrails for safe and faithful responses
* Provide a clear architecture diagram (slide or image).

### 2. Retrieval Strategy

* Discuss:
  + Document chunking approach (size, overlap, granularity).
  + Choice of embedding model (e.g., sentence-transformers models, or any free embedding model from Hugging Face).
  + Retrieval method (dense vector search, BM25, or hybrid).
* Suggest methods for ensuring relevance and faithfulness, e.g., reranking, enforcing citations, or returning source snippets.

### 3. Guardrails & Failure Modes

* Address how the system will handle:
  + No relevant answers (graceful fallback message).
  + Hallucinations (enforcing source citations).
  + Sensitive queries (blocking or filtering).
* Propose monitoring metrics to track retrieval quality (e.g., precision@k, recall@k, response faithfulness).

### 4. Scalability Considerations

* Explain how your design would handle:
  + A 10x increase in documents.
  + 100+ concurrent users.
  + Cloud deployment under cost constraints (e.g., serverless or GPU-efficient scaling).

### 5. Prototype (Minimal but Runnable)

* Implement a minimal prototype of the system using free and open-source tools only.
* Requirements:
  + Ingest a few sample documents (synthetic or provided).
  + Index them in a local vector store (e.g., FAISS, Chroma, Weaviate local mode).
  + Use a free/open LLM (e.g., flan-t5-small, opt-125m, Llama 2 7B (local inference), or any Hugging Face-hosted free model).
  + Allow natural language Q&A with retrieved sources cited.
* Bonus: Include a small evaluation (e.g., precision@k, recall@k).

### Deliverables (Part 2):

Inside the /Task2/ folder:

* **architecture\_diagram.pptx** — visual system architecture.
* **notes.md** — brief write-up of design trade-offs, retrieval strategy, guardrails, and scaling plan.
* **prototype/** (folder):
  + rag\_prototype.py OR rag\_demo.ipynb (minimal runnable demo)
  + README.md (exact run steps, including which **free/open models** to download and how to run locally)
  + docs/ (sample input docs)
  + *(optional)* evaluation.csv (retrieval metrics)

# Final Expected output:

Your final submission must be a compressed folder named:  
 **FirstName\_LastName\_DataScience**

## PPT (Final\_Presentation.pptx)

Your presentation should have **6–9 slides** covering both tasks:

1. **Data Preparation (Task 1)** – How you treated raw data (missing values, shutdown detection).
2. **Analysis Strategy (Task 1)** – Methods for clustering, anomaly detection, and forecasting.
3. **Insights (Task 1)** – Key findings (shutdown patterns, anomalies, operational states).
4. **System Architecture (Task 2)** – RAG pipeline diagram + explanation.
5. **RAG Challenges & Guardrails (Task 2)** – Retrieval strategy, hallucination prevention, scalability.

## Notes

* **Task 1** code must be runnable end-to-end.  
  **Task 2** must use **free/open-source models and tools** (e.g., Hugging Face models, FAISS/Chroma, LangChain/LlamaIndex).
* Include clear instructions in every README.md.