	Wind Turbine Scada Performance Analysis	ASSIGNMENT - 4
		AI/ML INTERN

### Dataset:

<https://www.kaggle.com/datasets/berkerisen/wind-turbine-scada-dataset>

### Dataset Provided Columns:


- Date/Time
- LV ActivePower (kW)
- Wind Speed (m/s)
- Theoretical\_Power\_Curve (kWh)
- Wind Direction (°)

### Task 1 — EDA

1. Plot the time-series trend for all four parameters and identify any missing or abnormal readings.
2. Visualize the relationship between Wind Speed and LV ActivePower using a scatterplot to observe the basic power curve behavior.

### Task 2 — Supervised Learning (Time-Series Forecasting for ALL 4 Variables)

1. Convert the dataset into a time-series format and prepare windowed data.
2. Build time-series forecasting models to predict:
  - LV ActivePower (kW)
  - Wind Speed (m/s)
  - Theoretical Power Curve (kWh)
  - Wind Direction (°)
3. Train and test the forecasting models and evaluate using suitable error metrics.
  - Plot the predicted vs actual values for all four variables in a user interface with date and Parameters (LV Active Power (kW), Wind Speed (m/s), Theoretical Power Curve (kWh), Wind Direction (°)) filters.

	Wind Turbine Scada Performance Analysis	ASSIGNMENT - 4
		AI/ML INTERN

### Task 3 — Unsupervised Learning

Anomaly Detection in Power Generation - Detect abnormal points where actual power deviates strongly from the theoretical power curve (underperformance detection).

### Task 4 — AI Task

AI Turbine Performance Score Generator

1. Calculate a performance score using the ratio of actual power vs theoretical power.
2. Scale the score between 0–100.
3. Categorize the turbine state as Good / Moderate / Poor.
4. Produce a simple automated suggestion based on the score.

### Task 5 — RAG-Based System

1. Allow users to upload any PDF, DOCX, or TXT files containing wind turbine manuals or maintenance guidelines.
2. Extract text from the uploaded document and store it in a vector database.
3. Build a simple Retrieval-Augmented Generation (RAG) system to answer user questions based on the uploaded content.
4. Demonstrate at least two question–answer examples using the RAG pipeline.
5. Design a user interface that captures user input and presents the corresponding output clearly and interactively.