Rayfield Systems: Technical Track — Week 2

WEEK 2: DATA PIPELINE DESIGN

GOAL:
By the end of this week, you will build your first working prototype data pipeline. This pipeline should load real energy data, clean it, and prepare it for downstream AI/ML analysis. You will also begin mapping out how this system will operate automatically, and what AI capabilities will enhance it. Your work this week will be foundational for Weeks 3–5.

Note: Forecasting workflow was used to structure this week's deliverables. You are free to change up some of the tasks outlined below to fit your chosen workflow better.

DELIVERABLES:

By Friday, you will submit:

1. A functioning data pipeline prototype (even if basic)
2. A clear system flow diagram (CSV → Processing → Output)
3. Documentation of dataset choices, cleaning logic, and feature engineering
4. Draft ideas for automation and AI/ML integration

Day 1 – Dataset Selection & Import

Task 1.1: Choose Your Dataset

You will begin by selecting a real-world dataset relevant to clean energy, such as:

- Solar generation (e.g., hourly kWh output)
- Wind production
- Hydroelectric output

Recommended sources:

- EIA.gov
- OpenEI
- Kaggle
- NREL.gov

Choose one dataset and document:

- Why this dataset was selected

- What time span and granularity it covers
- What energy workflow it could support (e.g., forecasting, anomaly detection, compliance)

Task 1.2: Load and Inspect the Data

You will now load the dataset into your development environment (Python, Jupyter Notebook, R, etc.). Preview the data and answer:

- How many rows/columns does it contain?
- Are there missing values or inconsistent formats?
- What fields stand out as useful or problematic?

Day 2 – Data Cleaning & Exploration

Task 2.1: Clean the Dataset

You will clean the dataset using basic preprocessing methods:

- Drop or fill missing values (NAs)
- Format column types (e.g., datetime, numeric)
- Rename headers to readable names
- Convert units if needed (e.g., MWh to kWh)

Use code comments to explain your cleaning steps. Save a cleaned version of your dataset (cleaned_data.csv).

Task 2.2: Explore and Structure the Data

Perform basic exploratory analysis:

- Use describe(), .info(), or visuals (e.g., line plots, histograms)
- Identify trends or outliers
- Find useful metrics (e.g., daily output average, max/min, percent change)

Task 2.3: Identify Key Metrics

Make a short list of metrics or columns your AI/ML tools will likely use:

- E.g., “Rolling 7-day average production,” “Hourly % change,” or “Time since last peak”

Document how these features might help automate monitoring, forecasting, or reporting workflows.

Day 3 – Pipeline Mapping

Task 3.1: Create a Data Pipeline Diagram

You'll now design the structure of your data pipeline. Use draw.io or a Figma board to draft a visual that includes:

- Input: CSV file or API
- Processing: Cleaning, transformations, feature generation
- Output: Dashboard, AI model input, automated alert
- Optional: Storage layers, scheduling triggers, report generation

Export or screenshot your pipeline and upload it to your shared project folder.

Task 3.2: Annotate Each Step

Label each component of your pipeline:

- What tool or method is being used (e.g., Pandas, Airflow, CRON)?
- When does it run? (daily/hourly/trigger-based?)
- What are the expected inputs/outputs?

Task 3.3: Think Through Edge Cases

Document at least three "what-if" scenarios. Examples:

- What happens if a new data column appears?
- How do you handle missing or duplicated data on a holiday?
- Can this pipeline scale for multiple datasets?

Day 4 – Feature Planning for AI/ML

Task 4.1: Identify AI/ML Opportunities

Now that your data is structured, list ways AI/ML can improve your workflow:

- GPT-generated summaries of daily energy reports
- Forecasting next day's output using regression or time series
- Anomaly detection for outlier days

- Alert generation when metrics go out of bounds

Document the “why” behind each one — how will it help a real energy role?

Task 4.2: Match Features to User Roles

Return to your personas from Week 1. For each persona, ask:

- What would they gain from this pipeline?
- Which feature solves a real pain point?

Example:
“The Asset Performance Analyst would benefit from a GPT-generated summary to speed up daily performance reviews, especially when reviewing 10+ sites at once.”

Task 4.3: Map AI into the Pipeline

Draft a second version of your pipeline that includes AI/ML components:

- Where does the model plug in?
- What output does it generate?
- Where does the user review or act on the result?

Label all components clearly. This will guide your ML integration in Week 3.

Day 5 – Modeling Strategy & System Logic

Task 5.1: Choose Your First Model

Pick one simple model to implement next week. Examples:

- Linear regression for daily forecasting
- Isolation Forest for anomaly detection
- LLM (e.g., GPT-4) for summarizing daily logs

Explain:

- Why this model?
- What are its inputs and outputs?
- What data preprocessing will it require?

Task 5.2: Define System Logic & User Interaction

Sketch out how your system will behave with the model integrated. For example:

1. User uploads data
2. AI flags an anomaly
3. Summary and risk score are emailed to team

Decide:

- What triggers the pipeline?
- When do humans step in?
- How are alerts or results shared?

Task 5.3: Document the Flow in Figma or Diagrams.net

Turn this logic into a clear, visual flowchart or low-fidelity UI. Include:

- Upload screen or API intake
- Results display (table/chart/summary)
- Trigger options (button, schedule, threshold breach)
- Output options (email, report export, Slack)

This can serve as your AI pipeline's first interactive prototype.

WEEK 2 GOALS RECAP

1. Choose and clean a relevant dataset tied to a real energy role or workflow.
2. Build a visual map of your data pipeline, including AI/ML touchpoints.
3. Prepare to implement one AI or ML feature that clearly solves a user problem.
4. Develop system-level thinking and documentation habits that prepare you for full automation in Weeks 3–5.

Individual Deliverables (for someone working alone):

Note: Working solo is only permitted if your teammates are unresponsive *and* you've made a genuine effort to find at least two other people to collaborate with but have failed.

Day 1 – Dataset Selection & Import (Solo Version)

Deliverable 1.1 – Dataset Justification Document

Create a short write-up (Markdown or PDF) that includes:

- Dataset name and source (e.g., EIA, Kaggle)
- Why you picked it (e.g., high granularity, relevant to solar forecasting)
- What task this supports (e.g., anomaly detection, load prediction)
- Time coverage and frequency (e.g., hourly, daily, monthly)

Deliverable 1.2 – Data Loading Notebook (`1_data_import.ipynb`)

- Load dataset using pandas
- Use `.info()`, `.head()`, `.describe()` to inspect structure
- Include markdown cells answering:
    - Rows/columns
    - Any nulls or strange formats
    - First thoughts on useful fields

Day 2 – Data Cleaning & Exploration (Solo Version)

Deliverable 2.1 – Cleaned Data File

- Code in a notebook (`2_cleaning.ipynb`) with clear code comments
- Steps must include:
    - Dropping/filling NAs
    - Renaming columns
    - Formatting datetime + units
- Save cleaned output as `cleaned_data.csv`

Deliverable 2.2 – Quick Analysis Notebook (`2_exploration.ipynb`)

- Create simple visualizations (line chart, histogram)
- Include summary stats: mean, max, min, % change
- Markdown answers to:
    - What trends or outliers are visible?
    - What surprises you?

Deliverable 2.3 – Feature Planning Doc

- Write a list of ~5 engineered metrics or features for ML use
- Example:
  - 7-day rolling avg
  - Hourly % change
  - Time since last peak
- Briefly explain how each might help forecasting or anomaly detection

Day 3 – Pipeline Mapping (Solo Version)

Deliverable 3.1 – Pipeline Diagram (v1)

- Use **diagrams.net**, **Figma**, or hand-drawn + scanned
- Include:
  - Input (CSV/API)
  - Cleaning step
  - Feature generation
  - Output (e.g., chart, ML-ready CSV, alert)

Deliverable 3.2 – Annotated Diagram Notes (`pipeline_notes.md`)

- For each box in diagram, explain:
  - Tool used (e.g., Pandas, CRON, Supabase)
  - Trigger (daily? manual?)
  - Input/output file names

Deliverable 3.3 – Edge Cases Document

- Markdown file: `edge_cases.md`
  - What if data format changes?
  - What if there's no data one day?
  - How will you scale for multiple sites?

Day 4 – Planning for AI/ML (Solo Version)

Deliverable 4.1 – AI/ML Use Case List (`ml_ideas.md`)

- Pick 2–3 features you could build
  - Daily forecast model (e.g., Linear Regression)
  - GPT summary of daily output
  - Outlier detection (e.g., Isolation Forest)
- For each:

    - What's the input/output?
    - Who does it help?

Deliverable 4.2 – Persona Matching

- Revisit Week 1 personas or create one if missing
- For each AI idea above, write 2–3 lines on how it helps one persona
    - "Operations Manager could use daily forecast to plan ahead…"

Deliverable 4.3 – Pipeline Diagram v2 (with ML)

- Update your pipeline diagram
    - Show where model runs
    - Where the summary/output appears
    - Label what gets stored/shared
- Save as `pipeline_with_ml.png` or `.pdf`

Day 5 – Model & System Logic (Solo Version)

Deliverable 5.1 – Model Strategy Doc (`model_plan.md`)

- Pick one model you'll build next week
    - Regression, Anomaly detection, or GPT summary
- For it, answer:
    - What are the inputs?
    - What features does it need?
    - Why did you choose it?

Deliverable 5.2 – System Logic Sketch

- Sketch your full solo system logic
    - What triggers pipeline?
    - When does model run?
    - How is output shared?
- Write it out or include a logic diagram

Deliverable 5.3 – System Flowchart / UI Mockup

- Low-fidelity diagram of:
    - User uploading data (or API fetch)
    - ML output display (table/summary/graph)
    - Options: auto-email, export report
- Save as: `system_flowchart.pdf` or `ml_ui_mockup.png`