Hackathon Part A – Pipeline Design

Objectives

In this exercise, you will:

- Design and implement a modular Python data pipeline to process multiple related datasets.
- Use Pandas and other tools to ingest, clean, transform, and merge real-world data.
- Generate a minute-by-minute temperature estimate for 2012 using a model based on daily weather summaries.
- Document your pipeline logic, assumptions, and outputs clearly for reuse by other developers.

Overview

You have been provided with two zipped files in the folder PartA_Data. The first contains data about the time of sunrise, sunset and the length of the day in Edinburgh. The second contains weather data from a weather station in Scotland.

Your goal is to build a Python data pipeline that merges these sources, models temperature changes over time, and outputs a **synthetic dataset estimating temperature every minute for 2012**. This data will later be used to analyse electronic equipment performance under environmental stress.

Pipeline Structure

Your repository should contain the following:

```
/data-pipeline-hackathon/
|
├── data/
|   ├── Edinburgh-daytime.xlsx
|   └── Strathspey-weather.xlsx
|
├── pipeline/
|   ├── ingest.py
|   ├── clean.py
|   ├── transform.py
|   ├── merge.py
|   └── resample.py
|
├── output/
|   └── merged_summary.csv
|
├── main.py
└── requirements.txt
```

Each module should perform a clear function:

- **ingest.py** – reads Excel data, handling multiple worksheets dynamically.
- **clean.py** – removes empty rows, fixes headers, and standardises column names.
- **transform.py** – converts date fields and derives useful numerical values.
- **merge.py** – joins astronomical and weather data on date.
- **resample.py** – interpolates daily data to one-minute intervals using a sinusoidal temperature model.

Temperature Estimation Model

To estimate temperatures between the daily minimum and maximum, use a **sinusoidal model**:

$$T(t) = T_{avg} + A \cdot \sin\left(\frac{\pi(t - t_{peak})}{12}\right)$$

Where:

- T_{avg} is the average of daily min and max temperature.
- A is half the difference between max and min.
- t_{peak} is the hour of peak temperature, typically around 15:00.

This generates a smooth temperature curve that peaks mid-afternoon and reaches its minimum near dawn.

Output

Your final dataset should:

- Contain 527,040 rows (one per minute from midnight 31 December 2011 to 23:59 on 31 December 2012).
- Include columns for datetime, estimated_temp, and optionally metadata such as Location.

Deliverables

1. **Working pipeline** – all stages functional and reproducible.
2. **Test process** – confirm row counts and data consistency.
3. **Code quality** – meaningful variable names and clear comments.

Hackathon Part B – Pipeline Design Karma

Objectives

In this exercise you will:

- Adapt your pipeline to process a different pair of datasets.
- Test whether your code handles variations in worksheet names and structures.
- Update documentation and discuss any necessary code changes.

Overview

You have been provided with two zipped files in the folder PartB_Data. The first contains data about the time of sunrise, sunset and the length of the day in London. The second contains weather data from a weather station in Hertfordshire.

Your brief is to estimate the temperature every minute for the year 2012. This will be used in a further analysis of operational data of electronic equipment in the vicinity.

Key Tasks

1. Modify ingest.py so it accepts parameters for **location** and **month_code**.
2. Ensure column names are standardised across all regions (e.g. "Max. Temp. (°C) 09/09h" → "Max").
3. Update the merge logic to include a "Location" column.
4. Test your pipeline with both pairs of files:
    - Edinburgh + Strathspey
    - London + Royston

Document any differences you encounter and describe how you resolved them.

Output

Your pipeline should again produce a one-minute interval temperature estimate for 2012, now specific to each location.

Hackathon Part C – Big Data model building

Objectives

In this exercise, you will:

- Build a model linking ambient temperature to equipment performance.
- Explore whether temperature fluctuations impact fan speeds, workloads, or failure rates.
- Present findings visually using graphs or dashboards.

Overview

You have been given equipment performance data for a telecommunications mast in the Strathspey area of Scotland. Combine this with your temperature dataset from Part A to explore how environmental changes affect operational parameters such as fan speed or power use.

The data details the workloads being undertaken by each of the 10 transmitters on the mast and the speed at which their cooling fans are spinning.

- Each transmitter has 3 fans in an N+1 configuration
- Minimum fan speed is 120 RPM delivering 3 CFM
- Maximum fan speed is 800 RPM delivering 20 CFM
- Maximum number of connections per transmitter is 168
- Maximum data bandwidth per transmitter is 20Mhz
- Fan failure rate per minute is nominally 1 in 3,285,000 (approx. 50,000 MTTF)
- Each connection uses 1/4 W and each 1Mhz of data bandwidth uses 2W

You also have error log, service log and customer complaint data files.

Guidance

- Use Python (Pandas, Matplotlib, or PySpark if appropriate) to join datasets and model relationships.
- Consider regression or correlation analysis between temperature and system performance.
- Discuss potential effects of global warming scenarios (+1°C, +3°C, +5°C).

Hackathon Part D – Compare Big Data sets

Objectives

In this exercise you will:

- Apply your model to multiple datasets.
- Compare results between masts in different regions.
- Identify and interpret anomalies.

Overview

You have been given performance data for several telecommunications masts in the Royston area. Use your model from Part C, adjusting for local temperature profiles derived from Part B.

Investigate:

- Whether any masts show unusual performance patterns.
- If these differences may relate to environmental or equipment factors.

Deliverable

Write a short report or create a presentation summarising:

- The methodology used to compare masts.
- Observed anomalies and possible causes.
- How your model adapts to new environmental data.

Extension Discussion

Parts **C** and **D** are designed as open-ended exploration challenges. You are encouraged to experiment with additional tools such as:

- Plotly or Dash for interactive dashboards
- Azure or AWS for scaling data analysis
- ML frameworks for predictive modelling

Final Notes

By completing this hackathon, you will have demonstrated:

- End-to-end pipeline development skills.
- The ability to generalise and adapt code.
- An understanding of data modelling and environmental impact analysis.
- Clear technical documentation and collaborative practices.