10 Academy: Artificial Intelligence Mastery

Solar Data Discovery: Week 0 Challenge

Kickstart Your AI Mastery with Cross-Country Solar Farm Analysis

Date: 15 May - 22 May 2025

Table of Contents

Challenge Overview

This week's challenge is focused on understanding, exploring, and analyzing solar farm data found in Benin, Sierra Leone, and Togo. The challenge aims to evaluate candidates for the 12-week training program in Data Engineering (DE), Financial Analytics (FA), and Machine Learning Engineering (MLE).

Applicants who proceed to the next level by demonstrating sufficient performance in this week's challenge will have a clear picture of the required discipline, resilience, proactivity, talent diversity, and other essential elements of the 10 Academy training. Those who can not make it to the limited spots available will gain a clear understanding of the direction they should improve to prepare for FA, DE & MLE job positions in the future. Everyone will gain project experience to showcase in their professional profile.

This week, therefore, is a **win-win** for everyone. We advise you to put your best effort into completing as many tasks as possible. We know that the number of tasks you are required to complete is a lot, and you **will not** have time to build intuition or to be comfortable with the new concepts and skills you are exposed to with this week's challenge. Please **note that** building a deeper understanding is not the purpose for this week's project. Moreover, you may have never done or attempted to do some of the tasks before this training. If you are confused and **overwhelmed**, know that it **is expected**.

The tutors, community managers, and all other teams are there to support you as best as they can. Be proactive in asking questions, provide resources that may help others, and above all **persist**!

Business Objective

MoonLight Energy Solutions aims to develop a strategic approach to significantly enhance its operational efficiency and sustainability through targeted solar investments. As an Analytics Engineer at MoonLight Energy Solutions, your task is to perform a quick analysis of an environmental measurement provided by the engineering team and translate your observation as a strategy report. Your analysis should focus on identifying key trends and learn valuable insights that will support your data-driven case - your recommendation based on the statistical analysis and EDA. In particular, your analysis and recommendation must present a strategy focusing on identifying high-potential regions for solar installation that align with the company's long-term sustainability goals. Your report should provide an insight to help realize the overarching objectives of MoonLight Energy Solutions.

Dataset Overview

The data for this week's challenge is extracted and aggregated from Solar Radiation Measurement Data. Each row in the data contains the values for solar radiation, air temperature, relative humidity, barometric pressure, precipitation, wind speed, and wind direction, cleaned and soiled radiance sensor (soiling measurement) and cleaning events.

. The structure of the data is as follows

- **Timestamp (yyyy-mm-dd hh:mm)**: Date and time of each observation.
- **GHI (W/m^2)**: Global Horizontal Irradiance, the total solar radiation received per square meter on a horizontal surface.
- **DNI (W/m^2)**: Direct Normal Irradiance, the amount of solar radiation received per square meter on a surface perpendicular to the rays of the sun.
- **DHI (W/m^2)**: Diffuse Horizontal Irradiance, solar radiation received per square meter on a horizontal surface that does not arrive on a direct path from the sun.
- **ModA (W/m^2)**: Measurements from a module or sensor (A), similar to irradiance.
- **ModB (W/m^2)**: Measurements from a module or sensor (B), similar to irradiance.
- **Tamb (°C)**: Ambient Temperature in degrees Celsius.
- **RH (%)**: Relative Humidity as a percentage of moisture in the air.
- **WS (m/s)**: Wind Speed in meters per second.
- **WSgust (m/s)**: Maximum Wind Gust Speed in meters per second.
- **WSstdev (m/s)**: Standard Deviation of Wind Speed, indicating variability.
- **WD (°N (to east))**: Wind Direction in degrees from north.
- **WDstdev**: Standard Deviation of Wind Direction, showing directional variability.
- **BP (hPa)**: Barometric Pressure in hectopascals.
- **Cleaning (1 or 0)**: Signifying whether cleaning (possibly of the modules or sensors) occurred.
- **Precipitation (mm/min)**: Precipitation rate measured in millimeters per minute.
- **TModA (°C)**: Temperature of Module A in degrees Celsius.
- **TModB (°C)**: Temperature of Module B in degrees Celsius.

- **Comments**: This column is designed for any additional notes.

Week's Topics Covered

1. **Python Programming:**
   - Task-specific programming assignments.
2. **GitHub Commands:**
   - Continuous committing and repository management.
3. **Data Understanding and Exploration:**
   - Applying exploratory data analysis techniques.
4. **CI/CD:**
   - Understanding continuous integration and continuous deployment.
5. **Streamlit**
   - Creating a dashboard using Streamlit.

Team

Facilitators :

- Yabebal
- Mahlet
- Kerod
- Rediet
- Rehmet

Key Dates

- **Challenge Introduction** - 8:30 AM UTC time on Thursday 08 May 2025.
- **Interim Submission** - 8:00 PM UTC time on Sunday 11 May 2025.
- **Final Submission** - 8:00 PM UTC time on Wednesday 14 May 2025.

Instructions

Task 1: Git & Environment Setup

Objective: Get everyone comfortable with version control before touching data.

- **Initialize Repository**
    - Create a new GitHub repo named solar-challenge-week1.
    - Clone it locally and set up a Python virtual environment (venv or conda).
- **Branching & Commits**
    - Create a branch called setup-task.
    - Commit at least **3 times** with messages like “init: add .gitignore”, “chore: venv setup”, “ci: add GitHub Actions workflow”
        - .gitignore (include data/ and any .csv/.ipynb_checkpoints/)
        - requirements.txt
        - GitHub Actions workflow (.github/workflows/ci.yml) that runs pip install -r requirements.txt.
- **Basic CI**
    - Add a GitHub Actions workflow file (.github/workflows/ci.yml) that runs python --version or pip install -r requirements.txt.
- **README**
    - In README.md, document how to reproduce the environment:
- Merge setup-task into main via a Pull Request.
- Suggested folder structure:

```
├── .vscode/
│   └── settings.json
├── .github/
│   └── workflows
│       ├── unittests.yml
├── .gitignore
├── requirements.txt
├── README.md
 |------ src/
├── notebooks/
│   ├── __init__.py
│   └── README.md
├── tests/
│   ├── __init__.py
└── scripts/
    ├── __init__.py
    └── README.md
```

Key Performance Indicators (KPIs):

- Dev Environment Setup.

Task 2: Data Profiling, Cleaning & EDA

Objective: Profile, clean, and explore each country's solar dataset end-to-end so it's ready for comparison and region-ranking tasks.

Create branch**:** eda-<country> (e.g. eda-benin)

Notebook**:** <country>_eda.ipynb

Perform Exploratory Data Analysis (EDA) analysis on the following:

- **Summary Statistics & Missing-Value Report**
  - df.describe() on all numeric columns., df.isna().sum() and list any column with >5% nulls.
- **Outlier Detection & Basic Cleaning**
  - Look for missing values, outliers, or incorrect entries, especially in columns like GHI, DNI, and DHI and check for outliers, especially in sensor readings (ModA, ModB) and wind speed data (WS, WSgust).
  - Compute Z-scores for GHI, DNI, DHI, ModA, ModB, WS, WSgust; flag rows with |Z|>3.
  - Drop or impute (median) missing values in key columns.
  - **Export** cleaned DataFrame to data/<country>_clean.csv (ensure data/ is in .gitignore and never commit CSVs).
- **Time Series Analysis**
  - Line or bar charts of GHI, DNI, DHI, Tamb vs. Timestamp.
  - Observe patterns by month, trends throughout day, or anomalies, such as peaks in solar irradiance or temperature fluctuations.
- **Cleaning Impact**
  - Group by Cleaning flag and plot average ModA & ModB pre/post-clean.
- **Correlation & Relationship Analysis**
  - Heatmap of correlations (GHI, DNI, DHI, TModA, TModB).
  - Scatter plots: WS, WSgust, WD vs. GHI; RH vs. Tamb or RH vs. GHI.
- **Wind & Distribution Analysis**
  - Wind rose or radial bar plot of WS/WD.
  - Histograms for GHI and one other variable (e.g. WS).
- **Temperature Analysis**
  - Examine how relative humidity (RH) might influence temperature readings and solar radiation.
- **Bubble Chart**

    - GHI vs. Tamb with bubble size = RH or BP.

Key Performance Indicators (KPIs):

- Proactivity to self-learn - sharing references.
- EDA techniques to understand data and discover insights,
- Demonstrating Stats understanding by using suitable statistical distributions and plots to provide evidence for actionable insights gained from EDA.

Task 3: Cross-Country Comparison

Objective: Synthesize the cleaned datasets from Benin, Sierra Leone, and Togo to identify relative solar potential and key differences across countries.

Branch: compare-countries
Notebook: compare_countries.ipynb

- Load each country's cleaned CSV (data/benin_clean.csv, etc.) locally.
- **Metric Comparison**
    - **Boxplots** of GHI, DNI, DHI side-by-side (one plot per metric, colored by country).
    - **Summary Table** comparing mean, median, and standard deviation of GHI, DNI, DHI across countries.
- **Statistical Testing** *(optional but recommended)*
    - Run a one-way ANOVA (or Kruskal–Wallis) on GHI values to assess whether differences between countries are significant.
    - Briefly note p-values.
- **Key Observations**
    - A markdown cell with **3 bullet points** summarizing what stands out (e.g., "Country X shows highest median GHI but also greatest variability").
- **(Bonus) Visual Summary**
    - A small bar chart ranking countries by average GHI.

Key Performance Indicators (KPIs):

- Inclusion of all three countries in each plot
- Correct implementation and reporting of p-values
- Relevance and actionability of those insights
- Use of summary table comparing mean/median/SD for each metric

Bonus (Optional): Interactive Dashboard

Objective: Build a Streamlit app—code only, no data—to visualize your insights.

- Designing and developing a dashboard using Streamlit to visualize data insights.
- Integrating Python scripts to fetch and process data dynamically.
- Implementing interactive features (e.g., sliders, buttons) to allow users to customize visualizations.
- Deploying the Streamlit dashboard to [Streamlit Community Cloud](https://streamlit.io/cloud).
- Suggested Additional Folder Structure

```
├── app
│   ├── __init__.py
│   ├── main.py  # main Streamlit application script
│   ├── utils.py  # utility functions for data processing and visualization
└── scripts
    ├── __init__.py
    └── README.md
```

Key Performance Indicators (KPIs)

- **Dashboard Usability**: Ease of use, with intuitive navigation and clear labels.
- **Interactive Elements**: Effective use of Streamlit widgets to enhance user engagement.
- **Visual Appeal**: Clean and professional design that effectively communicates data insights.
- **Deployment Success**: Fully functional deployment, accessible via a public URL.

Minimum Essential To Do:

1. Create branch: dashboard-dev
2. App: app/main.py with:
   - Widgets to select countries.
   - Boxplot of GHI or other plots .
   - Top regions table.
3. Design and develop the Streamlit dashboard to visualize the dataset with interactive elements
4. Git Hygiene:
   - Keep data/ ignored; app reads local CSVs.
5. Commit & PR:
   - 1 commit (“feat: basic Streamlit UI”).
   - Commit your work with a descriptive commit message.

6. Document the development process and usage instructions in the README.md file.

Due Date (Submission)

Sunday: (May, 18, 20258:00 PM (UTC)

What to Submit:

- GitHub link to your main branch.
- An interim report covering your Week 0 plan, including:
  - ○ Task 1 summary (Git & environment setup)
  - ○ Task 2 approach (profiling, cleaning & EDA outline)

Wednesday: (May, 21, 20258:00 PM (UTC)

What to Submit:

- GitHub link to your main branch.
- A final report covering all Week 0 work, written in a Medium-blog style (markdown or PDF).
- Your dashboard screenshot is placed in the repository (e.g. under dashboard_screenshots).

Other Considerations:

- **Documentation:** Encourage detailed documentation in code and report writing.
- **Collaboration:** Emphasise collaboration through Github issues and projects.
- **Communication**: Regular check-ins, Q&A sessions, and a supportive community atmosphere.
- **Flexibility:** Acknowledge potential challenges and encourage proactive communication.
- **Professionalism:** Emphasise work ethics and professional behavior.
- **Time Management:** Stress the importance of punctuality and managing time effectively.

Tutorials Schedule

In the following, the color **purple** indicates morning sessions, and non-purple indicates afternoon sessions.

- Day 1 (Thursday 15 May 2025):
    - Introduction to the Challenge(Mahlet)
    - Python Environment, Git & GitHub Basics + CI/CD (Kerod)
- Day 2 (Friday 16 May 2025):
    - Data Science Workflow & CRISP-DM Basics (Rediet)
    - Data profiling and Exploratory Data Analysis Techniques (Kerod)
- Day3 (Saturday 17 May 2025)
    - Dashboard development using Streamlit (Rehmet)
- Day 4 (Monday 19 May 2025)
    - Q&A
- Day 5 (Tuesday 20 May 2025)
    - Q&A

Feedback

You will receive comments/feedback in addition to a grade.

References

- **Python Testing**
  - https://machinelearningmastery.com/a-gentle-introduction-to-unit-testing-in-python/
  - https://docs.python-guide.org/writing/tests/
  - https://realpython.com/python-testing/
- **Dashboard design and implementation**
  - Get started - Streamlit Docs
  - Streamlit 101: An in-depth introduction | by Shail Deliwala | Towards Data Science
  - StreamLit - Data Scientists tool for developing web apps | by INSAID | Medium
  - Streamlit Community Cloud.
- **Python Programming:**
  - Object Oriented Programming
  - Python Courses and Tutorials: Online and On Site (python-course.eu)
- **Data Engineering**
  - What is Data Engineer: Role Description, Skills, and Background | AltexSoft
- **Version control – Git**
  - What is version control | Atlassian
  - Learn Git branching -- interactive way to learn Git
  - Git with large files
  - Which files to not track and how to not track them? | Atlassian
  - .gitignore docs
  - Conventional commits -- lightweight convention on top of commit messages.
- **CI/CD**
  - What is Continuous Integration | Atlassian
  - DevOps Pipeline | Atlassian
  - 7 Popular Open Source CI/CD Tools - DevOps.com
  - Setting up a CI/CD pipeline on Github