Interim Report: Week 0 Plan for Solar Data Discovery Challenge

Author: Eyoab Amare
Date: May 18, 2025
Repository: https://github.com/Eyoab11/solar-challenge-week1
Submission Deadline: 11:00 PM EAT, May 18, 2025

Executive Summary

The Solar Data Discovery challenge Week 0 report provides an overview of the preparation and preliminary examination of solar radiation datasets for Togo, Benin, and Sierra Leone. The report outlines both the first task (Git repository and environment configuration) and the second task ("data profiling, cleaning, and exploratory data analysis [EDA] approach" as described above. The work was done in Python 3.10 on a Windows environment, using PowerShell, and involves version control with venv and Git. The plan is designed to handle both the interim submission and any problems that may arise, such as merge conflicts and dependency issues.

Task 1: Git and Environment Setup

Objective

Establish a version-controlled repository and a reproducible Python environment to support data analysis tasks.

Activities and Deliverables

1. **Git Repository Configuration:**

- Initialized a Git repository at D:\Documents\Projects\10 Academy\solar-challenge-week1.
- Linked to the remote repository: https://github.com/Eyoab11/solar-challenge-week1.
- Created branches: eda-togo, eda-benin, and planned eda-sierra_leone.
- Configured .gitignore to exclude data/, venv/, and temporary files.
- Pushed initial commits with project structure (notebooks/, data/, requirements.txt).

Commands:

git init
git remote add origin https://github.com/Eyoab11/solar-challenge-week1.git
git checkout -b eda-togo
echo "data/" > .gitignore
git add .
git commit -m "Initial repository setup with structure"
git push origin eda-togo

2. Python Virtual Environment:

- Created a Python 3.10 virtual environment (venv) for dependency isolation.
- Upgraded pip to version 25.1.1.

Commands:

python -m venv venv
.\venv\Scripts\activate
python -m pip install --upgrade pip

3. Dependency Management:

- Defined requirements.txt with libraries: pandas==2.2.3, numpy==2.2.6, matplotlib==3.10.3, seaborn==0.13.2, scipy==1.14.1, windrose==1.6.8, jupyter==1.1.1, ipykernel==6.29.5.
- Removed pywin32==310 to resolve Linux compatibility issues in GitHub Actions.

Installed dependencies:

pip install -r requirements.txt

4. Jupyter Notebook Setup:

- Configured Jupyter Notebook with a venv kernel for EDA scripts.
- Tested execution of notebooks/togo_eda.ipynb.

Commands:

python -m ipykernel install --user --name=venv --display-name="Python (venv)"
jupyter notebook

Challenges and Mitigations

- Network Issues: Resolved Could not resolve host: github.com by flushing DNS (ipconfig /flushdns) and using HTTPS remote.
- Merge Conflicts: Addressed conflict in notebooks/togo_eda.ipynb by retaining the eda-togo version (git checkout --ours) and committing the merge.
- Dependency Conflicts: Eliminated pywin32 from requirements.txt to fix CI failures.
- Environment Issues: Recreated venv to resolve activation problems.

Outcomes

- Repository: Operational with branches eda-togo, eda-benin, and planned eda-sierra_leone, configured for continuous integration.
- Environment: Python 3.10 venv with all dependencies installed, supporting Jupyter Notebook execution.
- Status: Task 1 completed, enabling Task 2 implementation.

Task 2: Data Profiling, Cleaning, and EDA Approach

Objective

Profile, clean, and analyze solar radiation datasets to identify patterns, quality issues, and relationships for solar farm optimization.

Methodology

Task 2 involves processing datasets (togo-dapaong_qc.csv, Benin, Sierra Leone) using Python libraries (pandas, numpy, matplotlib, seaborn, scipy, windrose). Scripts (togo_eda.ipynb, benin_eda.ipynb, sierraleone_eda.ipynb) are stored in notebooks/, with outputs in data/ (excluded via .gitignore).

1. Data Profiling

- Statistics: Generate summary statistics (mean, median, min, max) for numeric columns (e.g., GHI, DNI, DHI, Tamb) using df.describe(include='all').

- Missing Values: Quantify missing data (df.isna().sum()) and calculate percentages.

- Quality Assessment: Identify negative irradiance values and outliers (Z-scores > 3) in key columns.

2. Data Cleaning

- Negative Values: Set negative GHI, DNI, DHI, ModA, ModB to 0 using np.maximum.

- Missing Values: Impute numeric columns with median; drop rows with missing Timestamp.

- Outliers: Flag rows with Z-scores > 3, with optional removal.

- Output: Save cleaned datasets to data/<country>_clean.csv.

3. Exploratory Data Analysis

- Time Series: Plot GHI, DNI, DHI, Tamb over Timestamp; visualize monthly averages.

- Cleaning Impact: Compare ModA, ModB means by Cleaning status using bar charts.

- Correlations: Compute and visualize correlation matrix for GHI, DNI, DHI, TModA, TModB with seaborn heatmap.

- Relationships: Create scatter plots for WS, WSgust, WD vs. GHI, and RH vs. Tamb/GHI.

- Wind Analysis: Generate wind rose plots for WS and WD using windrose; plot GHI and WS histograms with KDE.

- Temperature and Humidity: Perform regression analysis of RH vs. GHI.

- Multivariate: Produce bubble charts of GHI vs. Tamb, sized and colored by RH.

Implementation Strategy

- Scripts: Develop modular Jupyter notebooks with sections for loading, profiling, cleaning, and EDA, using a COUNTRY_NAME variable for reusability.
- Execution: Run scripts in venv via Jupyter, validating inputs and saving outputs.
- Error Handling: Implement try-except blocks for file operations and log data issues.

Anticipated Challenges and Mitigations

- Merge Conflicts: Resolve using git checkout --ours or manual edits, followed by commit and push.
- Dependency Issues: Ensure requirements.txt is compatible with CI; test windrose locally.
- Data Availability: Secure datasets from facilitators or use placeholders if unavailable.
- CI Failures: Monitor GitHub Actions logs and adjust requirements.txt as needed.

Expected Deliverables

- Profiling Reports: Summary statistics, missing value counts, and quality assessments for each dataset.
- Cleaned Datasets: data/<country>_clean.csv with corrected and imputed data.
- EDA Outputs: Visualizations (time series, wind rose, heatmaps, scatter plots, bubble charts) and insights on solar patterns.
- Repository: Updated branches with scripts, ready for submission.

Conclusion

The Solar Data Discovery challenge is set on a solid foundation by the Week 0 plan. Task 1 resolved technical issues by providing a pre-configured Git repository and Python environment. The task was accomplished successfully. This approach guarantees extensive data analysis.

Action Plan:

- Execute togo_eda.ipynb and validate outputs.
- Resolve eda-togo PR issues and merge to main.
- Submit repository: https://github.com/Eyoab11/solar-challenge-week1.