SolarIntel

Solar Energy Generation Prediction

Using EMHIRES and NASA POWER Data

Project Report

Live Demo:	https://huggingface.co/spaces/priyanshutomar2024/SolarIntel
Domain:	Machine Learning, Renewable Energy
Models:	Linear Regression, Random Forest Regressor
Deployment:	Streamlit on Hugging Face Spaces
Coverage:	29 European Countries, 2001–2015, Hourly

March 1, 2026

0 Contents

Abstract

Accurate forecasting of solar energy generation is essential for grid balancing, energy market operations, and renewable infrastructure planning across Europe. This report presents **SolarIntel**, an end-to-end machine learning system that predicts hourly solar capacity factors for 29 European countries by fusing 15 years of photovoltaic generation data from the European Commission's EMHIRES dataset with satellite-derived meteorological observations from NASA's POWER API. The resulting dataset comprises approximately 3.8 million hourly records and 34 features. Two regression models are trained and evaluated: a Linear Regression baseline ($R^2 = 0.788$) and a Random Forest Regressor ($R^2 = 0.911$). The superior model is deployed through an interactive Streamlit web application hosted on Hugging Face Spaces, enabling real-time prediction, 24-hour generation profiling, and cross-country comparison.

1 Introduction

Solar energy is one of the fastest-growing renewable energy sources globally. Accurate forecasting of solar energy generation is critical for grid stability, energy trading, and infrastructure planning. The variability of solar output, driven by weather conditions, geographic location, and time of day, makes prediction a non-trivial challenge.

This project, **SolarIntel**, addresses this challenge by building an end-to-end machine learning pipeline that predicts hourly solar energy capacity factors across 29 European countries. The system fuses 15 years of solar generation data from the European Commission's EMHIRES dataset with hourly meteorological observations from NASA's POWER API. Two regression models are trained, evaluated, and compared. The best-performing model is deployed through an interactive Streamlit web application hosted on Hugging Face Spaces.

1.1 Objectives

- Collect, clean, and merge solar generation and weather datasets spanning 2001–2015.
- Engineer temporal and geographic features suitable for regression modelling.
- Train and evaluate two regression models: Linear Regression and Random Forest Regressor.
- Deploy the trained model as an interactive web application for real-time predictions.

1.2 Scope

The pipeline covers 29 European countries at hourly resolution. The target variable is Capacity Factor, a normalised measure of solar output ranging from 0.0 (no generation) to 1.0 (maximum rated output). The final dataset contains approximately 3.8 million rows and 34 features.

2 Data Sources

Two primary datasets are used. Both are publicly available.

2.1 EMHIRES PV Dataset

The European Meteorological-derived High-resolution RES generation time series for present and future scenarios (EMHIRES) dataset is published by the European Commission's Joint Research Centre (JRC). The PV module provides hourly solar capacity factors for European countries from 1986 to 2015.

Table 1: EMHIRES PV Dataset Summary

Attribute	**Detail**
Source	European Commission, JRC
Temporal Coverage	1986–2015 (hourly)
Spatial Coverage	29 European countries
Format	Wide-format CSV (countries as columns)
Target Variable	Capacity Factor (0.0–1.0)

The dataset is in wide format with each column representing a country code (e.g., AT for Austria, ES for Spain). Each row corresponds to one hour. The data was filtered to the 2001–2015 range to align with the availability of NASA weather data.

2.2 NASA POWER API

The Prediction Of Worldwide Energy Resources (POWER) project by NASA provides satellite-derived meteorological data. Hourly observations were fetched via the POWER API for all 29 EMHIRES countries using geographic centroids.

Table 2: NASA POWER Weather Parameters

API Parameter	**Renamed To**	**Unit**
ALLSKY_SFC_SW_DWN	Irradiance	W/m^2
T2M	Temperature	°C
WS2M	Wind Speed	m/s

Data was collected for each country and each year (2001–2015) individually, with a 0.5-second delay between requests to respect API rate limits. The full fetch takes approximately 20 minutes due to the volume of requests (29 countries $\times$ 15 years = 435 API calls).

3 Data Pipeline

The pipeline follows a sequential flow: data loading, cleaning, transformation, merging, encoding, and export. Each stage is implemented as a standalone Python module and also consolidated in a single end-to-end script. Figure 1 illustrates the full architecture.

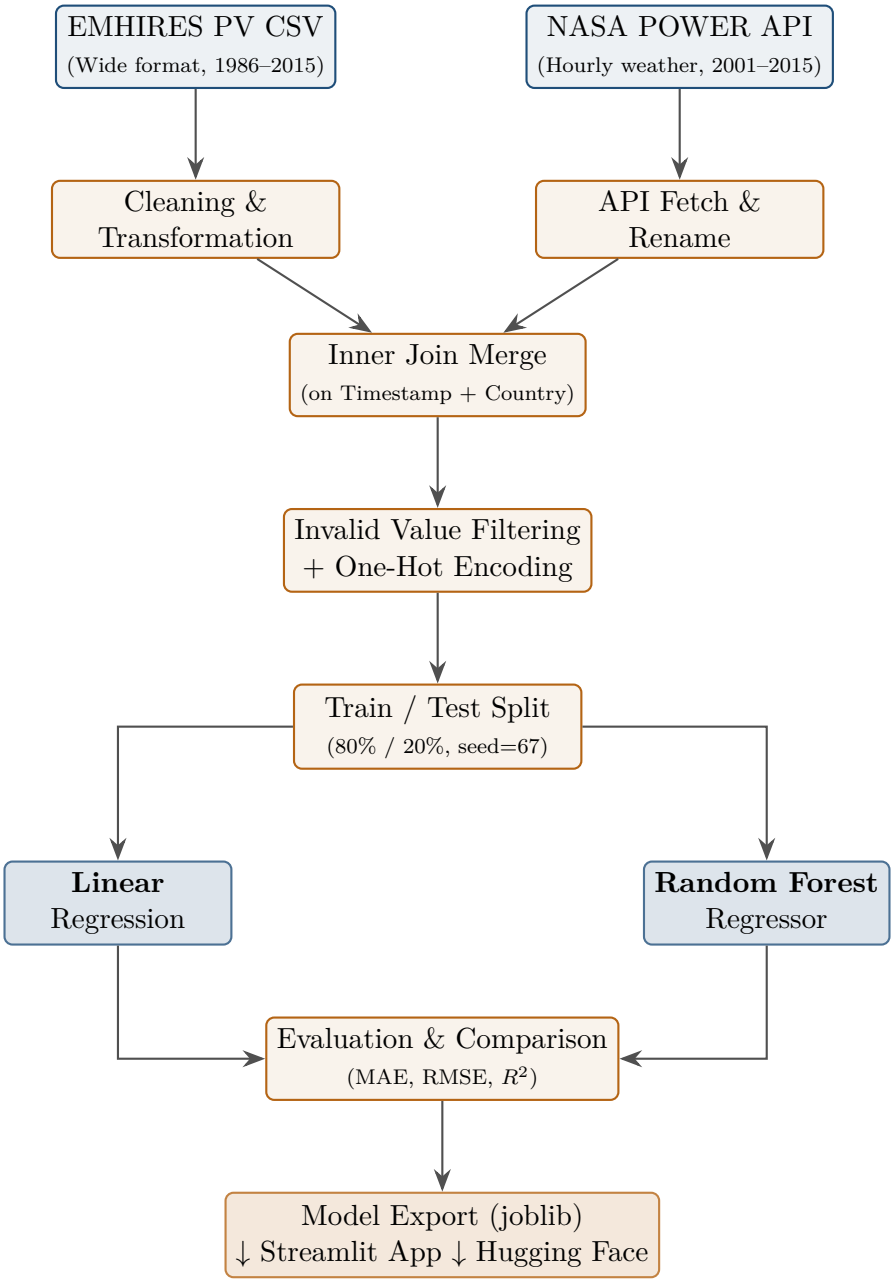

Figure 1: SolarIntel Pipeline Architecture

3.1 Cleaning and Transformation

The raw EMHIRES CSV is in wide format with no explicit timestamps. Timestamps were generated programmatically starting from 1 January 1986 at hourly frequency. The data was then reshaped from wide to long format using `pd.melt()`, producing three columns: `Timestamp`, `Country`, and `Capacity_Factor`.

Key operations:

- Timestamp generation using `pd.date_range()` from 1986-01-01, hourly frequency.
- Wide-to-long reshaping via `pd.melt()`.
- Filtering to 2001–2015 to match NASA POWER data availability.
- Extraction of temporal features: `Hour` (0–23) and `Month` (1–12).

3.2 Merging

The EMHIRES long-format data and NASA weather data were merged on `Timestamp` and `Country` using an inner join. This ensures every row has both a capacity factor value and corresponding weather observations. Float64 weather columns were downcast to float32 to reduce memory consumption.

Post-merge filtering removed invalid weather readings:

- Irradiance < 0 (physically impossible).
- Temperature $= -999.0$ (NASA fill value for missing data).
- Wind Speed $= -999.0$ (NASA fill value for missing data).

3.3 Encoding

The categorical `Country` column was one-hot encoded using `pd.get_dummies()`, producing 29 binary indicator columns (e.g., `Country_AT`, `Country_ES`). This is necessary because the regression models require numerical input. The resulting dataset was exported as `merged_encoded.csv`.

3.4 Final Dataset Summary

Table 3: Final Dataset Characteristics

Property	**Value**
Total Rows	∼3,812,690
Total Features	34
Numerical Features	5 (Hour, Month, Irradiance, Temperature, Wind Speed)
Encoded Features	29 (one per country)
Target Variable	Capacity Factor
Time Range	2001–2015 (hourly)
Countries	29 European nations

4 Feature Engineering

Feature engineering was kept deliberate, deriving only features with clear physical relevance to solar generation.

Table 4: Feature Descriptions

Feature	**Type**	**Range**	**Rationale**
Hour	Temporal	0–23	Solar output follows a diurnal cycle peaking around midday.
Month	Temporal	1–12	Captures seasonal variation in daylight hours and sun angle.
Irradiance	Weather	≥ 0 W/m^2	Primary driver of photovoltaic output.
Temperature	Weather	°C	Higher temperatures reduce PV panel efficiency.
Wind Speed	Weather	≥ 0 m/s	Affects panel cooling and efficiency indirectly.
Country_*	Categorical	0 or 1	Geographic identity capturing latitude, climate, and installed capacity differences.

The feature set was intentionally kept minimal to avoid overfitting and to maintain interpretability. Irradiance is the strongest predictor, with Hour and Month providing temporal context. The country one-hot encoding allows the model to learn location-specific baselines.

5 Model Training and Configuration

Two regression models were trained and compared. Both are provided by scikit-learn.

5.1 Train-Test Split

The dataset was split into 80% training and 20% testing subsets using `train_test_split()` with `random_state=67` for reproducibility. The `Timestamp` and `Capacity_Factor` columns were excluded from the feature matrix.

Table 5: Data Split

Set	Samples
Training	3,050,152
Testing	762,538
Total	3,812,690

5.2 Linear Regression

A standard Ordinary Least Squares (OLS) regression model. It assumes a linear relationship between the features and the target variable. This model serves as a baseline due to its simplicity and interpretability.

```
lr_model = LinearRegression()
lr_model.fit(X_train, y_train)
```

5.3 Random Forest Regressor

An ensemble method that fits multiple decision trees on bootstrapped subsets of the data and averages their predictions. It captures non-linear relationships and feature interactions that linear regression cannot.

```
rfr_model = RandomForestRegressor(
    n_estimators=100,
    random_state=67,
    n_jobs=-1,
    max_depth=12
)
rfr_model.fit(X_train, y_train)
```

Table 6: Random Forest Hyperparameters

Parameter	**Value**
n_estimators	100
max_depth	12
random_state	67
n_jobs	-1 (all CPU cores)

Both trained models were serialised using `joblib` and exported as `.pkl` files for deployment.

6 Model Evaluation

Three standard regression metrics were used to evaluate both models on the held-out test set.

6.1 Evaluation Metrics

Mean Absolute Error (MAE): The average absolute difference between predicted and actual values. Lower is better.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{1}$$

Root Mean Squared Error (RMSE): The square root of the average squared differences. Penalises larger errors more heavily than MAE.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2}$$

R-Squared (R^2): The proportion of variance in the target explained by the model. A value of 1.0 indicates perfect prediction.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{3}$$

6.2 Results

Table 7: Test Set Performance Comparison

Model	**MAE**	**RMSE**	R^2
Linear Regression	0.0534	0.0845	0.7880
Random Forest Regressor	**0.0280**	**0.0547**	**0.9112**

Table 8: Training Set Performance Comparison

Model	**MAE**	**RMSE**	R^2
Linear Regression	0.0534	0.0844	0.7882
Random Forest Regressor	**0.0278**	**0.0543**	**0.9123**

6.3 Analysis

The Random Forest Regressor significantly outperforms Linear Regression across all three metrics. Key observations:

- The Random Forest achieves an R^2 of 0.9112 on the test set, explaining over 91% of the variance in capacity factor. Linear Regression explains approximately 79%.
- The MAE for Random Forest (0.0280) is roughly half that of Linear Regression (0.0534), meaning its average prediction error is about 2.8 percentage points on the capacity factor scale.
- The close alignment between training and testing metrics for Linear Regression (MAE: 0.0534 vs 0.0534) confirms it is not overfitting, but its linear assumption limits its ceiling.
- The close alignment between training R^2 (0.9123) and test R^2 (0.9112) confirms that the Random Forest is generalising well and not overfitting, aided by the `max_depth=12` constraint.

7 Deployment

The trained models are deployed through a Streamlit web application hosted on Hugging Face Spaces.

7.1 Application Architecture

The application consists of two main components:

- **model_loader.py** – Handles model loading from `.pkl` files using `joblib`, constructs feature vectors from user inputs, and returns capacity factor predictions.
- **app.py** – The Streamlit frontend providing interactive controls (country selection, weather parameters, time inputs) and visualisations (prediction gauges, 24-hour generation profiles, country comparisons).

7.2 Features

The deployed application provides:

- Real-time capacity factor prediction based on user-selected country, time, and weather conditions.
- 24-hour generation profile simulation showing expected output across a full day.
- Country comparison tool for benchmarking solar potential across regions.
- Model selection toggle between Linear Regression and Random Forest Regressor.
- Installed capacity scaling to convert capacity factor to absolute power output (kW).

7.3 Hosting

The application is publicly accessible at:

`https://huggingface.co/spaces/priyanshutomar2024/SolarIntel`

Hugging Face Spaces provides free hosting for Streamlit applications with persistent storage for model files.

8 Repository Structure

The project is organised into modular directories, each serving a distinct purpose.

Table 9: Repository Directory Layout

Directory	**Purpose**
Walkthrough/	Complete documented walkthrough (Jupyter notebook and PDF). Recommended starting point.
Pipeline_Modules/	Modular per-stage scripts (8 modules covering visualisation, cleaning, merging, encoding, training, and analysis).
Final_Pipeline/	Single consolidated end-to-end pipeline script.
NASA_Data_Fetch/	NASA POWER API data collection script (`fetcher.py`).
Datasets/	Raw and processed datasets (compressed).
Models/	Trained model files (compressed `.pkl` files).
Demo_and_Hosting/	Streamlit application for deployment.
Report/	This report and reference materials.

9 Technology Stack

Table 10: Technologies Used

Layer	**Technology**
Data Processing	pandas, NumPy
Machine Learning	scikit-learn
Visualisation	matplotlib, seaborn, Plotly
Model Persistence	joblib
Web Application	Streamlit
API Data Source	NASA POWER API
Hosting	Hugging Face Spaces
Version Control	Git, GitHub

10 Future Work

Planned extensions include expanding SolarIntel into an AI agent-based system capable of autonomous decision support. This would involve integrating agentic reasoning to analyse solar generation trends, identify underperformance, and suggest corrective strategies without manual intervention.

Additional directions include:

- Incorporating additional weather parameters and higher spatial granularity.
- Exploring deep learning architectures (e.g., LSTM, Transformer) for sequential forecasting.
- Implementing real-time data ingestion for live prediction updates.

11 Conclusion

Key Result: The Random Forest Regressor achieves an R^2 of **0.9112**, explaining over 91% of variance in hourly solar capacity factor across 29 European countries.

SolarIntel demonstrates a complete machine learning workflow for solar energy generation prediction. The pipeline successfully merges two heterogeneous data sources into a unified dataset of approximately 3.8 million hourly observations. The Random Forest Regressor accurately captures the non-linear relationships between weather conditions, temporal patterns, geographic factors, and solar output. The deployed Streamlit application provides an accessible interface for exploring predictions across 29 European countries.

11 References

1. Gonzalez Aparicio, I., Zucker, A., Careri, F., Monforti, F., Huld, T., Badger, J. (2017). *EMHIRES dataset – Part I: Wind power generation*. European Commission, Joint Research Centre (JRC).
2. NASA Langley Research Center. *POWER – Prediction Of Worldwide Energy Resources*. Available at: https://power.larc.nasa.gov/
3. Pedregosa, F., et al. (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12, 2825–2830.
4. European Commission, Joint Research Centre. *EMHIRES PV Time Series*. Available at: https://setis.ec.europa.eu/european-commission-services/emhires