

PLANTPREDICT UPDATE & SDK DEVELOPMENT FOR API

Stephen Kaplan
Performance & Prediction Engineer

2018 PVPM Conference



LEADING THE WORLD'S
SUSTAINABLE ENERGY FUTURE



NOTICE TO RECIPIENTS

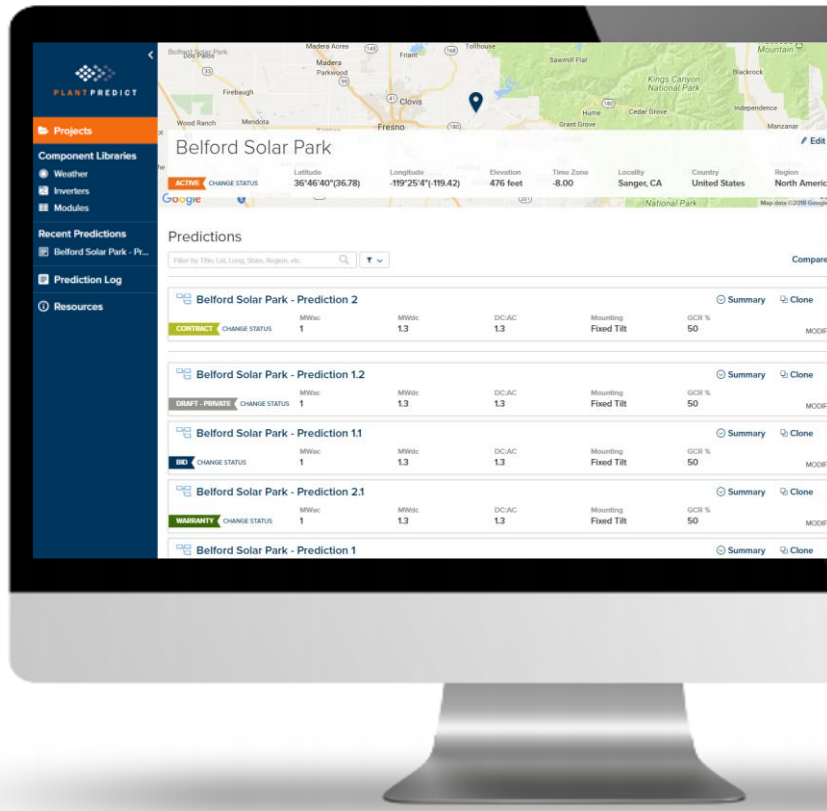
The Recipient acknowledges that (i) this presentation is provided to the Recipient for informational purposes only, (ii) neither First Solar, Inc. (“First Solar”) nor any of its affiliates bears any responsibility (and shall not be liable) for the accuracy or completeness (or lack thereof) of any information contained within this presentation and the oral statements made in connection therewith and this presentation and the oral statements made in connection therewith are not to be relied upon as a substitute for independent review of such other information as the Recipient may deem appropriate or prudent to review, (iii) no representation regarding the material within this presentation and the oral statements made in connection therewith is made by First Solar or its affiliates, (iv) neither First Solar nor any of its affiliates have made an independent verification as to the accuracy or completeness of the information within this presentation and the oral statements made in connection therewith, and (v) neither First Solar nor any of its affiliates shall have any obligation to update or supplement any information contained within this presentation and the oral statements made in connection therewith or otherwise provide additional information.



PLANTPREDICT OVERVIEW

INTRODUCING PLANTPREDICT

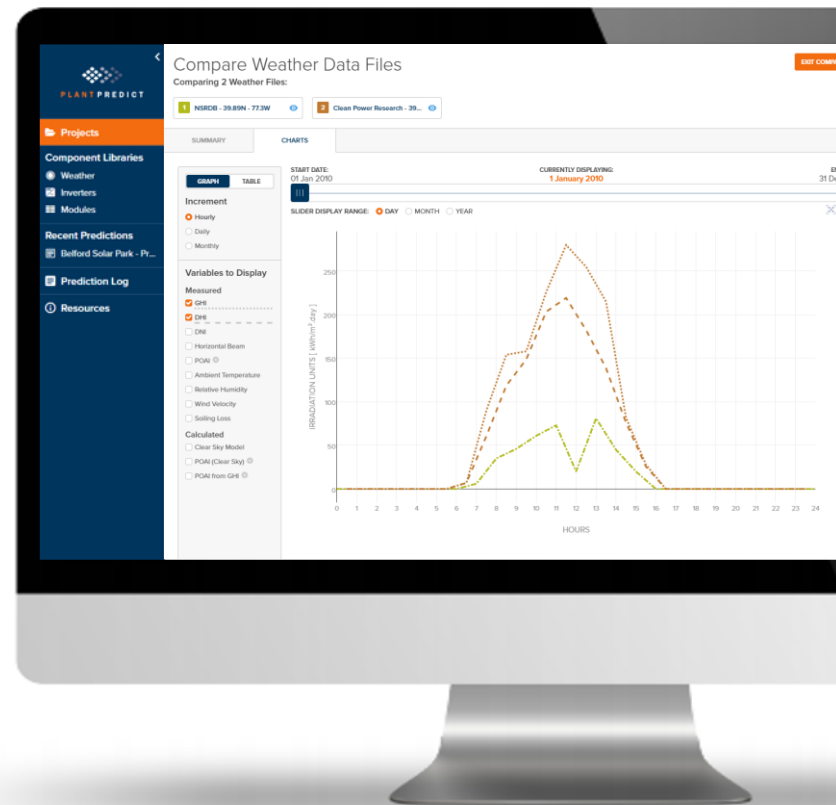
- Designed with the **utility-scale PV industry** in mind
- **Free**
- **Web-based tool**
- **Reduce prediction time by up to 75%**
- **Used in over 350 MWAC of contracted utility-scale PV projects**
- **All algorithms documented and published on www.plantpredict.com**
- **Independently reviewed and benchmarked against over 1 GW of operating facilities**



Reviewed by:    

PLANTPREDICT FEATURES

- *End-to-end utility-scale modeling*
 - Sub-hourly and multi-year predictions
 - Built-in spectral correction
 - Built-in MV and HV transformers
 - Built-in Availability and LGIA losses
 - No need for pre- or post-processing
- *Everything in one tool*
 - One-click weather download from leading providers
 - Pre-loaded with industry-standard weather, module, and inverter files
- *Advanced features, simple interface*
 - Compare weather and predictions side-by-side
 - Optimize your design with Batch, Clone, and Quick Edit
 - Cloud-hosted for ease of sharing and data security
 - API available for automation

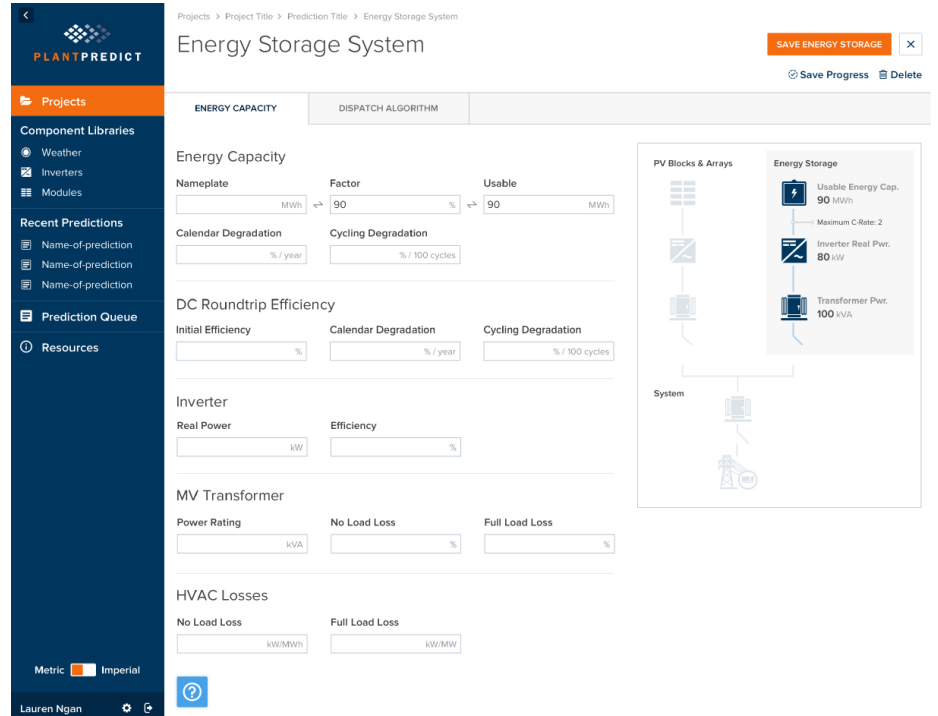
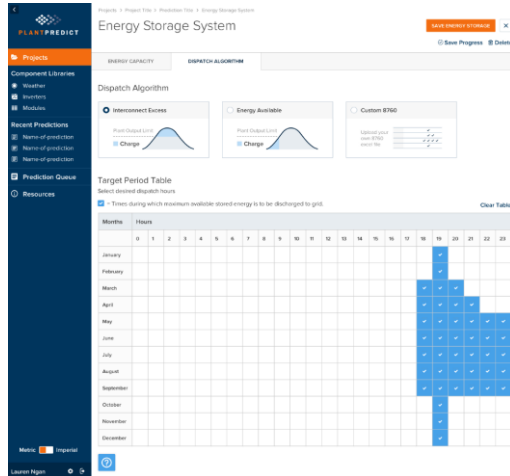




UPCOMING FEATURES

3 modeling options

1. **LGIA Excess** energy used to charge battery. Energy is discharged during user-defined target period.
2. **Energy Available** is used to charge the battery until it is full. Energy is discharged during user-defined target period.
3. **Custom** option allows user to define both charging and discharging.



- Characterize modules as “Monofacial” or “Bifacial”.
- Model structure shading and backside mismatch and post height for a DC field.
- Model transmission and bifaciality for a PV module.
- Uses the NREL 2D View Factor model (Marion)

The screenshot displays the 'Power Plant Builder' software interface. On the left is a dark blue sidebar with navigation options: Projects, Component Libraries (Weather, Inverters, Modules), Recent Predictions, Prediction Queue, and Resources. The main area shows a project configuration for 'Power Plant Builder' with a progress bar indicating steps: BLOCK (27.35 MWdc), ARRAY (27.35 MWdc), INVERTER (83.35 kWdc), and DC FIELD (83.35 MWdc, 300 W, 10 m). The 'DC Field' configuration is selected, showing details for 'Aleo S 19 300' modules: Manufacturer (Aleo Solar), Model (Aleo S 19 / 300), Rated Power (300 W), Temp. Coeff. (-0.4 %), Cell Technology (n-type mono c-Si), and Facility (Bifacial).

Below the module details are several configuration sections:

- Electrical:** Module Quality (0 %), Light Induced Deg. (2.5 %), DC Health (1 %).
- Mounting Structure:** Mismatch (1 %), DC Wiring at STC (1.5 %).
- Losses:** Heat Bal. Conductive Coeff. (29 U_c), Sandia Conductive Coeff. (-3.56 A), Cell to Module Temp. Diff. (3 °C).
- Heat Bal. Convective Coeff.:** 0 U_v and -0.075 B.
- Sandia Convective Coeff.:** -0.075 B.
- Tracker Actuator Load:** 0 MWh/MWp/Year.
- Bifacial Losses:** Structure Shading (0.5 %) and Backside Mismatch (2.0 %).

On the right, a 3D visualization shows a perspective view of a DC field with blue solar panels on a grid. A smaller diagram below shows a panel with a 15.10m height, 6.00m width, and 20.00m spacing, tilted at a 45-degree angle. A control panel for the visualization includes 'View Angles' (X, Y, Z, Z+), 'Zoom' (+, -), 'Viewer Azimuth' (21°), and 'Observer Height' (10'). A summary box indicates 'Modules High 9', 'GCR 33.4%', and 'Orientation Landscape' with an 'Azimuth 90°'.

Calculate single-diode parameters from datasheet parameters and adjust to match a target effective irradiance response.

Input data via IEC61853-type grid or full I-V curves.

Module Generator
Review and adjust calculations

General Characteristics

Cell Technology: pType mono Si BSF | Number of Cells in Series: 60

Module Type: 1-Diode Recommendation

Electrical Data at STC

Maximum Power	110.0 W
Voc	86.4 V
Isc	1.82 A
Vmp	66.9 V
Imp	1.64 A

Temperature Coeffs.

Power Temp. Coeff.	-0.31 %/°C
Voc Temp. Coeff.	0.34 %/°C
Isc Temp. Coeff.	0.04 %/°C

Calculated Performance

Module Temp: 30 °C | Irradiance: 1000 w/m²

Max Power	110.0 W	Isc	1.82 A	Imp	1.64 A	Par. Temp. Coeff.	-0.29 %/°C	Isr Temp. Coeff.	0.04 %/°C
Voc	86.4 V	Vmp	66.9 V	Voc Temp. Coeff.	0.34 %/°C				

1-Diode Parameters

Series Resistance at STC: 2.5 mΩ | Recombination Parameter: 14 | Shunt Resistance at STC: | Exp. Dep. of Shunt Resist.: | Dark Shunt Resistance: |

Series Resistance MAX: 200 mΩ | Recomb. Parameter MAX: | Saturation Current At STC: | Diode Ideality at STC: | Lin. Temp. Dep. of Gamma: |

Temperature: 25 °C | Displaying Curve: 25 °C

Irradiance	Rel. Efficiency
1000 w/m²	100 %
800 w/m²	97 %
200 w/m²	92 %

Variables to Display: Relative Efficiency Difference/Error

Efficiency Relative to 1000 W/m²

— 25 °C | 1000 w/m²
— 30 °C | 1000 w/m²

Module Generator
Enter Key I-V Points

25 °C | 50 °C | 60 °C | Add Temp. °C

Irradiance	Isc	Voc	Imp	Vmp	Pmp	Rel. Efficiency
1000 w/m²	1.86 A	82.7 V	1.42 A	64.9 V	92.5 W	100 %
100 w/m²	1.74 A	83.2 V	1.57 A	65.0 V	102.0 W	97 %
800 w/m²	1.26 A	81.8 V	1.16 A	65.0 V	74.4 W	92 %

Calculated Performance

Electrical Data at STC

Max Power	110.0 W	Isc	1.82 A	Imp	1.64 A
Voc	86.4 V	Vmp	66.9 V		

Temperature Coefficients

Par. Temp. Coeff.	-0.29 %/°C	Isr Temp. Coeff.	0.04 %/°C
Voc Temp. Coeff.	0.34 %/°C		

Download Excel Template | Upload Excel File

Module Generator
Enter Full I-V Curves

25 °C | 50 °C | 60 °C | Add Temp. °C

Displaying: 50 °C, 1000 w/m²

Irradiance	Current	Voltage	Power
1000 w/m²	0.488 A	200 V	97.6 W
1030 w/m²	0.397 A	221 V	87.8 W
1040 w/m²	0.421 A	216 V	90.8 W
1050 w/m²			

Relative Efficiency: 105%

Calculated Performance

Electrical Data at STC

Max Power	110.0 W	Isc	1.82 A	Imp	1.64 A
Voc	86.4 V	Vmp	66.9 V		

Temperature Coefficients

Par. Temp. Coeff.	-0.29 %/°C	Isr Temp. Coeff.	0.04 %/°C
Voc Temp. Coeff.	0.34 %/°C		

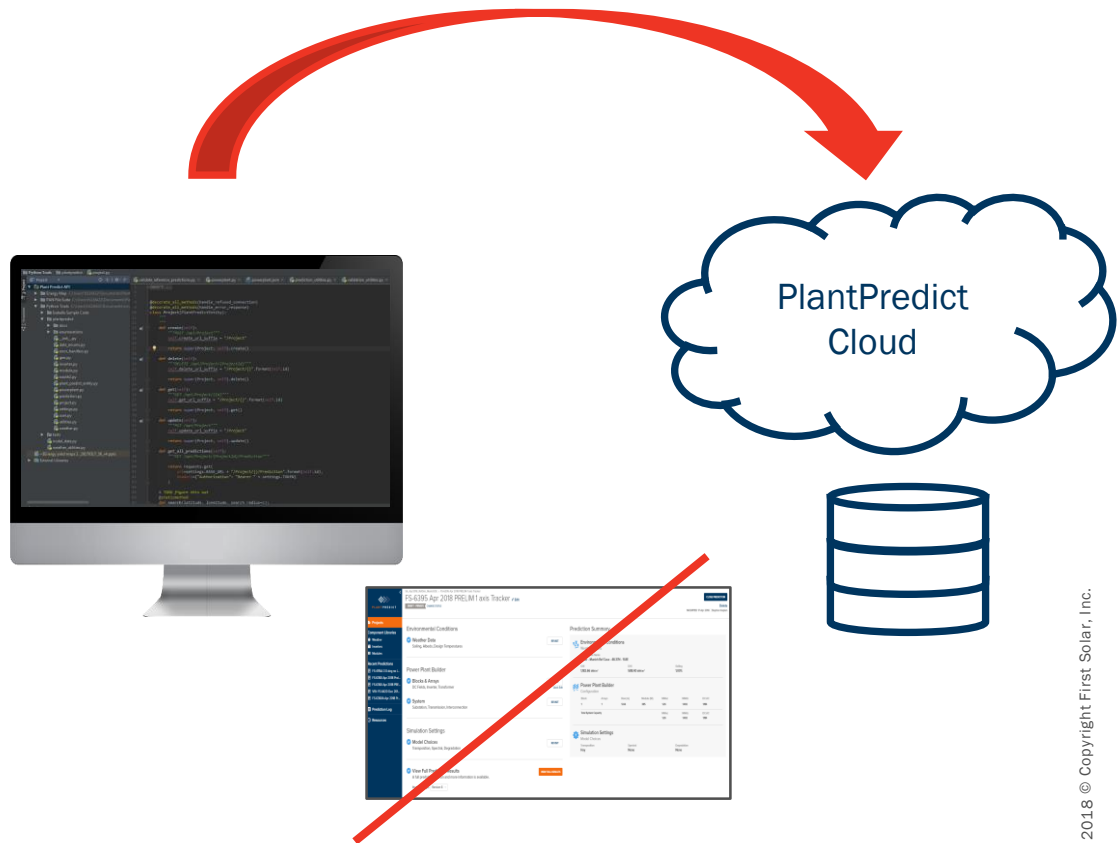
Download Excel Template | Upload Excel File



PLANTPREDICT API & CASE STUDY

PlantPredict API

- **PlantPredict's Applied Programming Interface (API) allows developer access to broad functionality.**
- **Automate everything**
 - Build power plants, run predictions, upload weather data, and more...
- **Endless applications**
 - Business/O&M software integration
 - Run batches of predictions with control over *every parameter*
 - Utilize as a weather/module/inverter database for any external application



Python SDK

The Python Software Development Kit (SDK) makes the PlantPredict API even more accessible with a library of functions.



Python SDK

The Python Software Development Kit (SDK) makes the PlantPredict API even more accessible with a library of functions.



Run all prediction variants in a project.

```
# Run all predictions in a project.
import plantpredict

project = plantpredict.Project(id=1)
for p in project.get_all_predictions():
    prediction = plantpredict.Prediction(id=p['id'], project_id=project.id)
    prediction.run()
```

Python SDK

The Python Software Development Kit (SDK) makes the PlantPredict API even more accessible with a library of functions.



Run all prediction variants in a project.

```
1 # Run all predictions in a project.
2 import plantpredict
3
4 project = plantpredict.Project(id=1)
5 for p in project.get_all_predictions():
6     prediction = plantpredict.Prediction(id=p['id'], project_id=project.id)
7     prediction.run()
```

Download and retrieve a weather file at a specified location..

```
1 # Download and retrieve a weather file from Meteorm.
2 import plantpredict
3
4
5 weather = plantpredict.Weather()
6 weather.download(latitude=30, longitude=-110, provider=plantpredict.WEATHER_DATA_PROVIDER['METEONORM'])
7 weather.get()
8
```

Python SDK

The Python Software Development Kit (SDK) makes the PlantPredict API even more accessible with a library of functions.



Run all prediction variants in a project.

```
1 # Run all predictions in a project.
2 import plantpredict
3
4
5 project = plantpredict.Project(id=1)
6 for p in project.get_all_predictions():
7     prediction = plantpredict.Prediction(id=p['id'], project_id=project.id)
8     prediction.run()
```

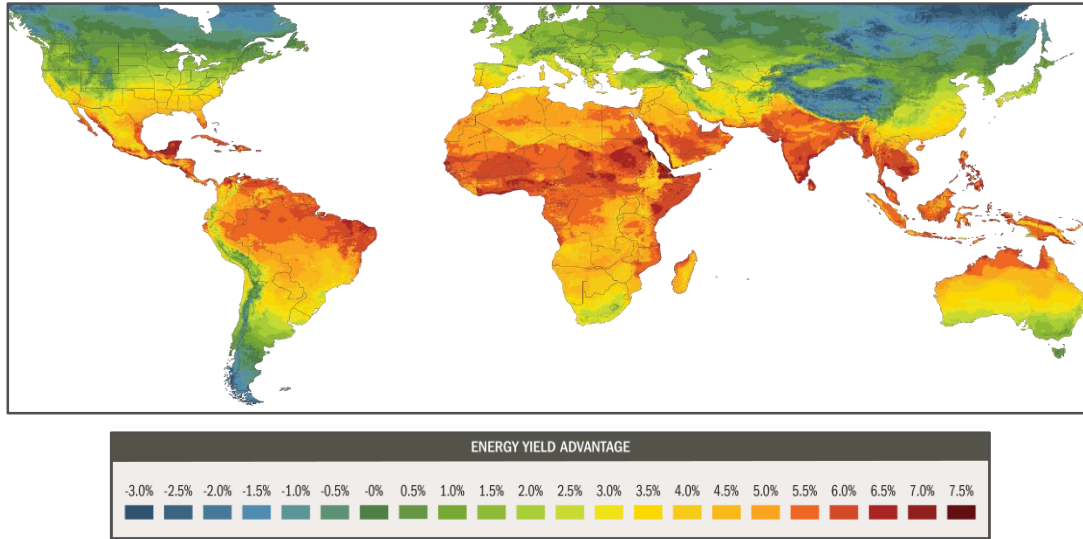
Update module used in energy prediction.

```
2 # Update module used in simulation.
3 import plantpredict
4
5
6 power_plant = plantpredict.PowerPlant(project_id=1, prediction_id=2)
7 power_plant.get()
8 power_plant.blocks[0]['arrays'][0]['inverters'][0]['dcFields'][0]['moduleId'] = 4
9 power_plant.update()
```

Download and retrieve a weather file at a specified location..

```
2 # Download and retrieve a weather file from Meteornorm.
3 import plantpredict
4
5
6 weather = plantpredict.Weather()
7 weather.download(latitude=30, longitude=-110, provider=plantpredict.WEATHER_DATA_PROVIDER['METEONORM'])
8 weather.get()
```


Case Study – Global Module Energy Yield Comparison



Integrated the PlantPredict Python SDK with Python statistics and plotting libraries to generate a global heat map, representing the relative performance of 2 modules.

STEPS

1. Automatically configure and run predictions at 100 representative locations across the globe for 2 modules.
2. Run a regression to correlate weather data with relative energy production.
3. Use a global grid of annual weather data at 0.25° (latitude) x 0.25° (longitude) resolution to calculate relative energy production across the globe.
4. Use graphing library to generate a heat map from the resulting data.

Need an account?
support@plantpredict.com





First Solar®