Solar Sight

Green! Solar! Sustainable!

Description: A crop and fertilizer recommendation.

APK file: github.com/Agnij-Moitra/solar-sight/raw/app/solarsight-release.apk

Source Code: github.com/Agnij-Moitra/solar-sight/tree/main

Vision

It was rightly said by James Cameron, “The nation that leads in renewable energy will lead the world.”. Our vision is to make solar energy accessible to the common people and add value to the lives of millions of people who rely on clean energy.

Problem

Due to the various uncertainties in weather conditions and solar radiation, Solar Power Plants cannot optimize their systems to incorporate the variance of atmospheric conditions. Similarly, for individual users they cannot estimate the amount of electricity produced.

Solution

Solar Sight is able to estimate the amount of electrical power generated in an area per 100 ft^2. It uses machine learning using python in the backend and Flutter for the app.

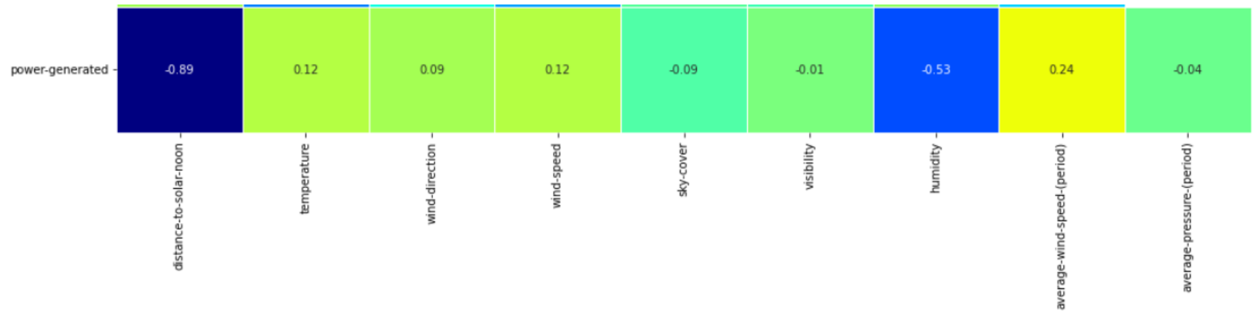
Target Clients and Commercial Viability

Solar Sight aims the help the existing individuals and corporations who use green energy calculate the amount of electricity generated. This is especially helpful for the solar power plant companies. It helps them improvise and optimize their power management systems.

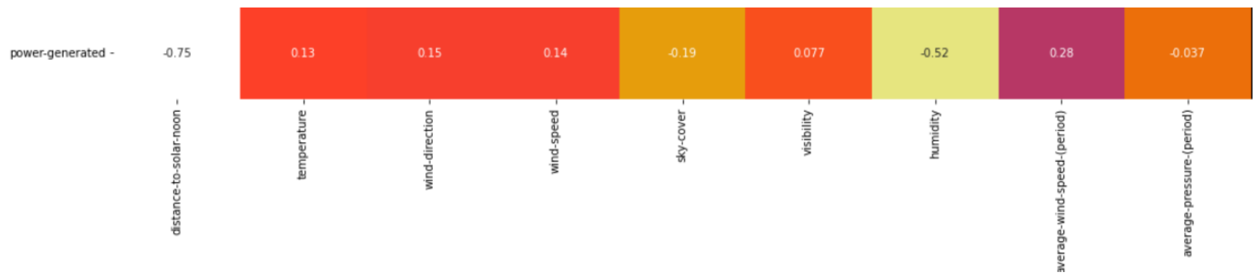
Implementation

Backend

- Using Spearman and Pearson correlation operators on our datasets we concluded that pressure, humidity, wind direction, wind speed and the radial distance to solar noon are correlated to the amount of electrical power generated.



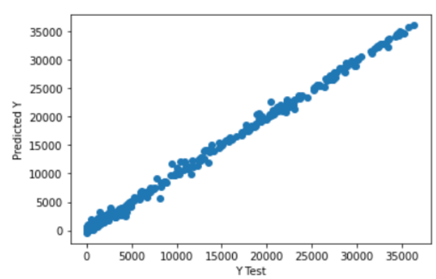
Spearman Correlation



Pearson Correlation

- Then we made a SciKit Learn pipeline for the above featureset which included preprocessing using Mean Imputers, to fill the missing values, and OneHotEncoder for numeric transformations. After that we used XGradient Boosting Ensemble as our model. We had an accuracy of 96%, Mean Absolute Score of 240 and Mean Squared Error of 426. Then we used pickle to convert our model to a binary file in order to decrease the time complexity. Moreover, we used Open Weather Maps and IP Geolocations API to fetch the above features for any location, as defined in supplementary.py.

```
MAE: 240.5830164670638
MSE: 181832.9705986999
RMSE: 426.41877374090825
```



- Further we used Flask to create our api on Heroku which returns the estimated electricity generated as a json file. It can be used in the form, https://solar-sight.herokuapp.com/api/?place=<location here> .

Frontend

For the front end we used Flutter, first prompting the user for the location. Eventually, we used the location to make an api call using flutter's built-in functions and displayed the output.