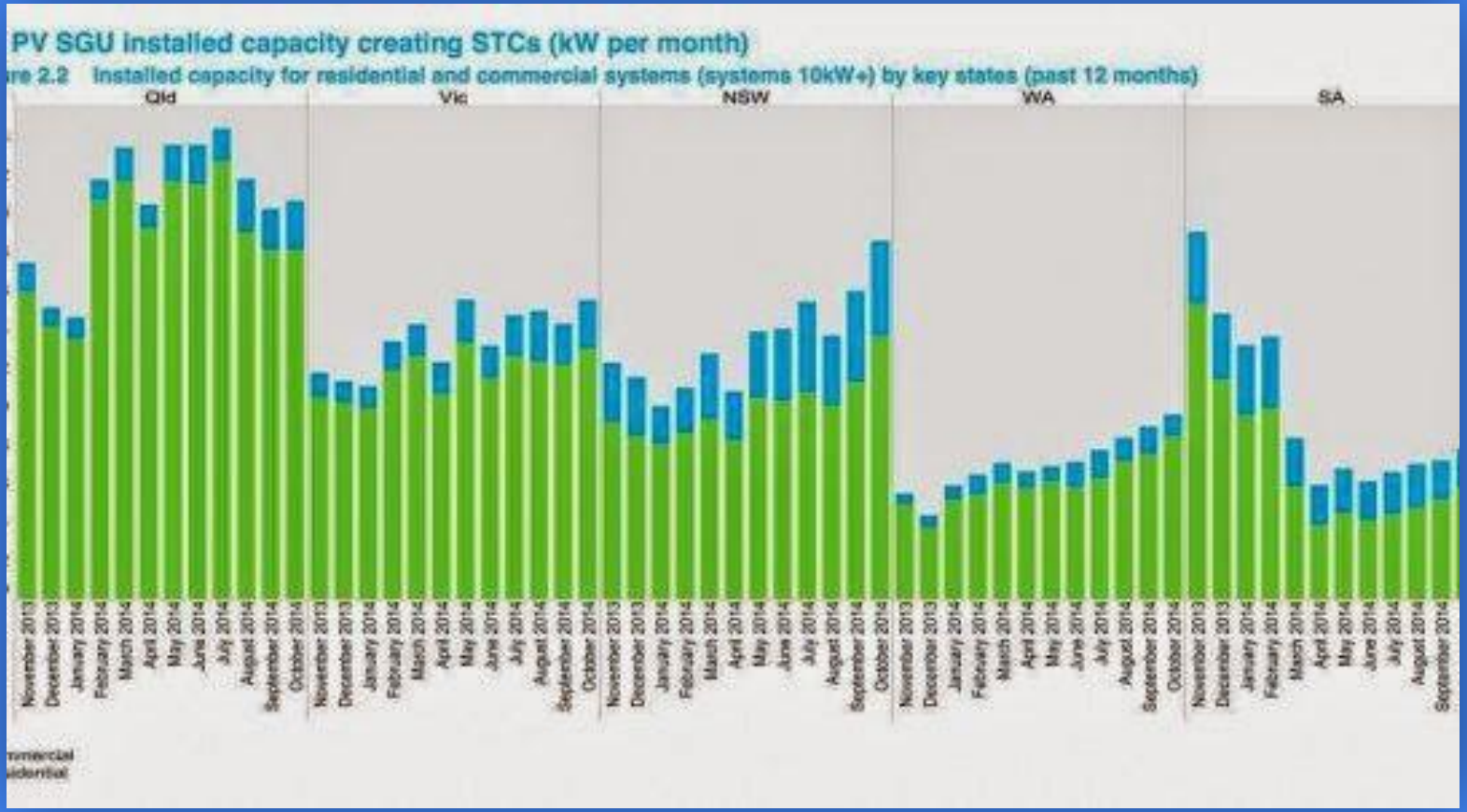
Solar Sight ☀️

Green! Solar! Sustainable!

Our 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.


Market Research

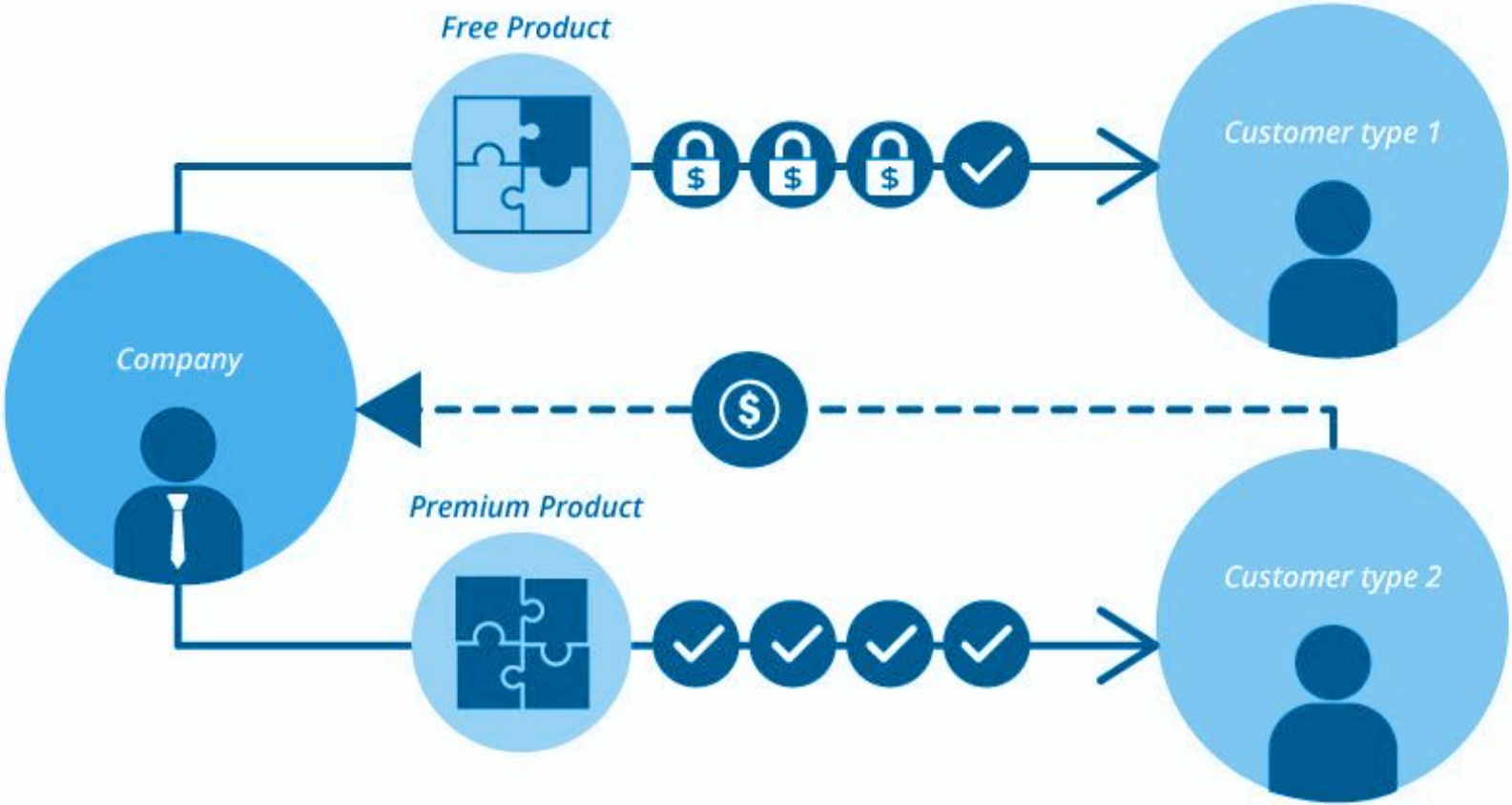

PV SGU installed capacity creating STCs (kW per month)
re 2.2 Installed capacity for residential and commercial systems (systems 10kW+) by key states (past 12 months)
Qld
Vic
NSW
WA
SA
November 2013
December 2013
January 2014
February 2014
March 2014
April 2014
May 2014
June 2014
July 2014
August 2014
September 2014
October 2014

How are we making solar the common man's energy source?

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. This is especially helpful for the solar power plant companies. It helps them improvise and optimize their power management systems.

Commercial Viability

Freemium


Free Product
Customer type 1
Company
Premium Product
Customer type 2

Free

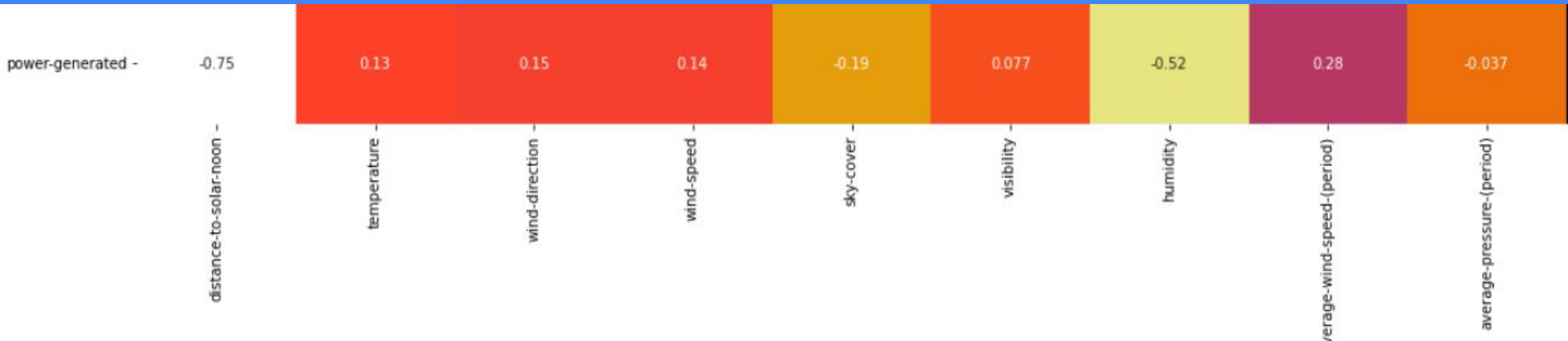
Paid and Sponsorships

adani SOLAR

Loom Solar

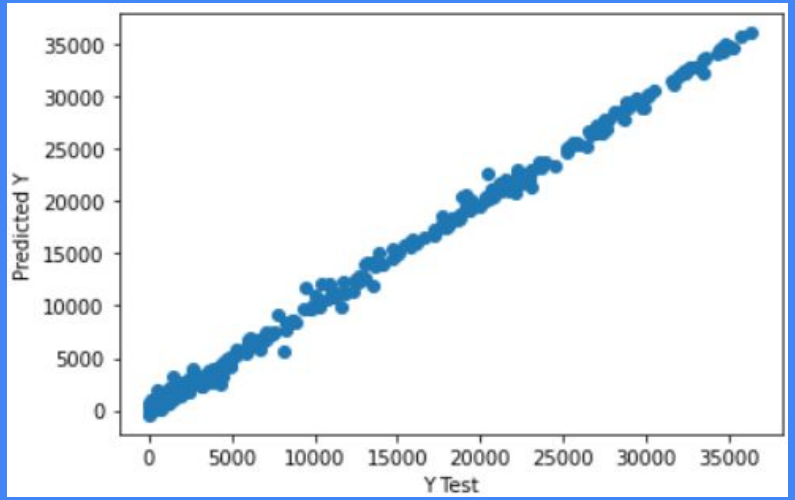
Implementation

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.



We had an accuracy of 96%, Mean Absolute Score of 240 and Mean Squared Error of 426. 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> .

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.

```
String solar(LatLng place) {
  return "${place.latitude},${place.longitude}";
}
```

Thank You!