Ashley Scurlock (ams5zx), Kevin Hoffman (keh4nb),
Nikki Aaron(na5zn), Siddharth Surapaneni (sss2ea)
Dr. Tashman
DS 5559: Big Data Analytics
5/9/2021

Solar Spark

Abstract

Solar energy can be both an eco-friendly and cost effective way to power your home. To encourage people to make the switch we built a model that could predict the yearly dollars saved on energy if the user installs 30 solar panels on their property. This project was inspired by the National Renewable Energy Laboratory's (NREL) NPVWatts® Calculator which accepts the user's home address, the residential electricity rate, and the system information of the solar panels such as the tilt of the panel, the size, and the percent system loss. Then in return it produces a report of the solar radiation, energy produced, and dollars saved on energy by month. The downside of this tool is that it requires the user to be knowledgeable about the solar panels they have installed or are planning to install which is not always the case. Our model is intended for someone who is curious about the financial benefits of solar energy and is taking the first steps to determine whether it would be a smart investment. By taking into account variables that are location based such as monthly weather patterns, elevation, and cost of energy the user of our model will be able to do less research yet receive a similar result.

Data

To predict solar output across the US, we used data from six sources. First, SimpleMaps provided us with latitude, longitude, county name, fips code, and demographic information about each zip code. This was filtered for continental US states only, then converted to a Geopandas GeoDataFrame for subsequent matching. NREL data from OpenEI was our source for solar radiation. There are three common measures of solar radiation, and we chose to use only Direct Normal Irradiation (DNI), which measures solar rays directly from the sun with no consideration of atmospheric conditions. USGS data was used to get elevations for each county, and this data

was joined by FIPS code. CEDA was our source for weather data consisting of monthly and annual average temperature and number of cloudy days measured at a fine grid of latitude and longitude points across the country. We gathered this data for two years 2018-2019, and averaged the two years together. To join this data to zipcodes, we used the BallTree method from SciKit Learn to find the nearest measured point to each zip code. This method also allowed us to impute any missing values with those from the nearest location. These data were all used to calculate the solar generation potential at each location so that we could run supervised model training. We also added an additional predictor for average annual household energy consumption by state sourced from EIA. This was joined by state name to the rest of the datasets.

Our final response variable was modified from solar generation alone to instead tell how much money could be saved using solar generation as opposed to buying electricity from the grid. To do this, we took the solar power generated and multiplied it by the cost to buy electricity in each state. This state power cost data was obtained from OpenEI, and includes prices as measured in 2018. To calculate power generated at each location from our predictors, we did not directly calculate using various equations available through research, as these were complex and difficult to gather. Instead, we fed the data into a Python package called PVLib, which uses well-researched, precise models and methods developed at Sandia National Laboratories to calculate power generation potential. To compare all locations under equal conditions, we assumed that each household had 30 solar panels of the same model, the Canadian Solar CS5P-220M (220W), angled at the same tilt. This produced the distribution of dollars saved shown below, with a mean of $597 and standard deviation of $146.

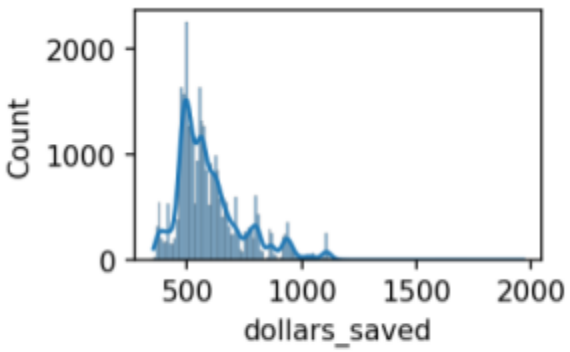

Figure 1: Response Variable Distribution

During exploratory data analysis, we were surprised to find that of all our predictors, solar radiation was the one with the smallest variation with a standard deviation of 5% of the mean. Factors that affect solar generation that varied the most were elevation and temperature (especially winter temperatures). Producing a Pearson correlation matrix revealed that multicollinearity would be an issue for us if we included cost of electricity as a predictor, as this variable was used to derive our response and had a high correlation (0.98). Solar radiation generated was highly correlated with elevation (0.81), average temperature (-0.54). Other high correlations (> 0.5) were seen between annual electricity consumption, percent cloudy days, and DNI. According to these correlations and choropleth mapped predictors shown in Figure 2, areas in the Mountain timezone should have great solar potential with high elevations, high radiation, and low average temperatures. A choropleth of the response variable in Figure 3 confirms this, but we also see high savings in California due to high DNI and electricity prices, as well as in northern Michigan due to their high electricity prices.

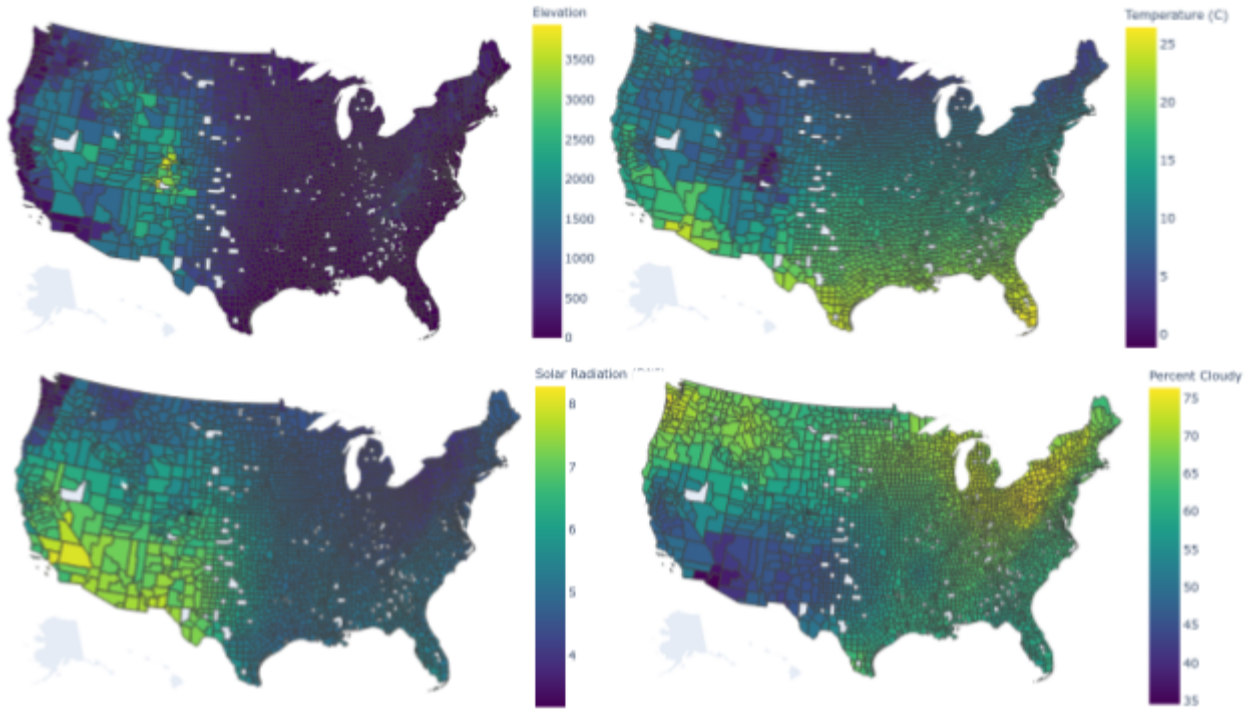

Figure 2: Predictor Choropleths - Elevation (top left), Temperature (top right), Solar Radiation (bottom left), Percent Cloudy Days (bottom right)

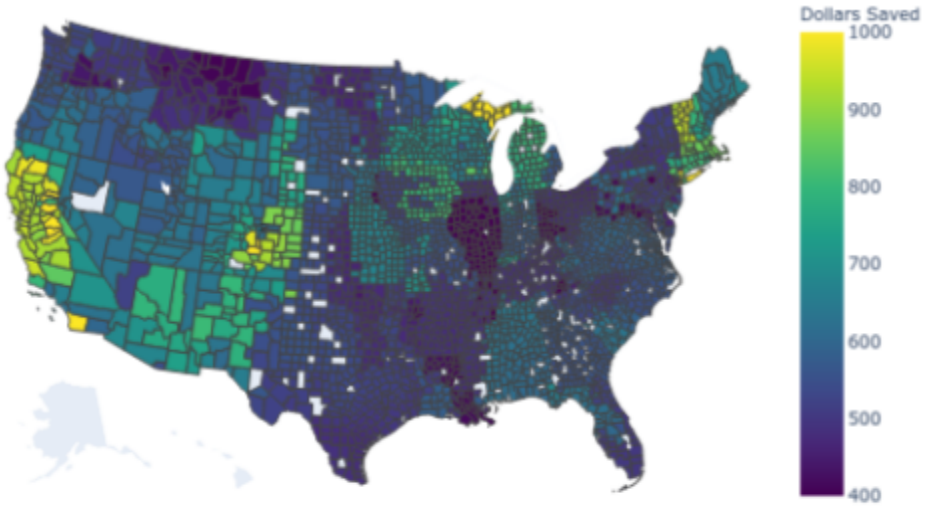

Figure 3: Response Choropleth - Dollars Saved

Methods

We fit three different models to our data in order to predict the response variable, dollars saved. The first model we constructed was a linear regression model to use as our baseline model to compare our other models to. Linear regression in Spark implements elastic net regularization so we tried tuning the regParam (or α) and elasticNetParam (or λ). The regularization parameters did not help the linear regression model as setting it to 0 resulted in the best performing model. The second model built was a random forest regression model which is an ensemble method of decision trees. Because the trees are built independently of each other, Spark can build these trees in parallel and then average the predictions from each tree resulting in a fast fitting and tuning process. We also chose to build this model because it is great for predicting non-linear relationships which we saw in our data from exploratory data analysis. The parameters tuned were maxIterations (number of trees), maxDepth, and minInstancesPerNode (or minimum samples per leaf). The third model we developed was a gradient boosted tree model. Like random forest regression, it uses decision trees but it builds them one at a time, improving upon the previous iteration. We also tuned it using similar parameters as the random forest regression. maxIteration in this case does not refer to the number of trees, but how many times to iterate on the previous tree.

Results

After fitting and tuning our three different approaches, we evaluated the fit using the Root Mean Square Error (RMSE). We chose to use RMSE as our evaluation metric because it penalizes outliers more and we want our model to be as close as possible to the true dollars saved. Below is a summary of our results.

Model Name	RMSE	Hyperparameters
Linear Regression	$109.23	maxIterations: 100 regParam: 0 elasticNetParam: 0
Random Forest Regression	$54.95	maxDepth: 8 numTrees: 250 minInstancesPerNode: 1
Gradient Boosted Trees	$39.59	maxIterations: 225 maxDepth: 7 minInstancesPerNode: 10

Figure 4: Our three different approaches along with their respective RMSE and hyperparameters.

Our best performing model was the gradient boosted tree model with an RMSE of 39.59 and the worst model was the linear regression with a RMSE of 109.23. Some of our features exhibited a non-linear relationship to the response variable of dollars saved so it makes sense it does not perform well. The main parameters we focused on tuning for the two tree based models were maxDepth and minInstancesPerNode using a manual grid search. maxDepth controls how deep the trees grow as a deeper tree has the capacity to learn a more complex relationship, at the risk of overfitting. We can regularize these trees with the minInstancesPerNode as it controls the minimum amount of observations in a leaf before splitting again. Increasing this number will reduce the number of splits at higher depths and prevent overfitting. One drawback to our best model, gradient boosted tree, is the fitting time as it took over 2 hours on a 16 core machine which made it harder to tune.

We visualized our results by plotting our prediction error on the United States map to see if the model did not do as well in certain locations.

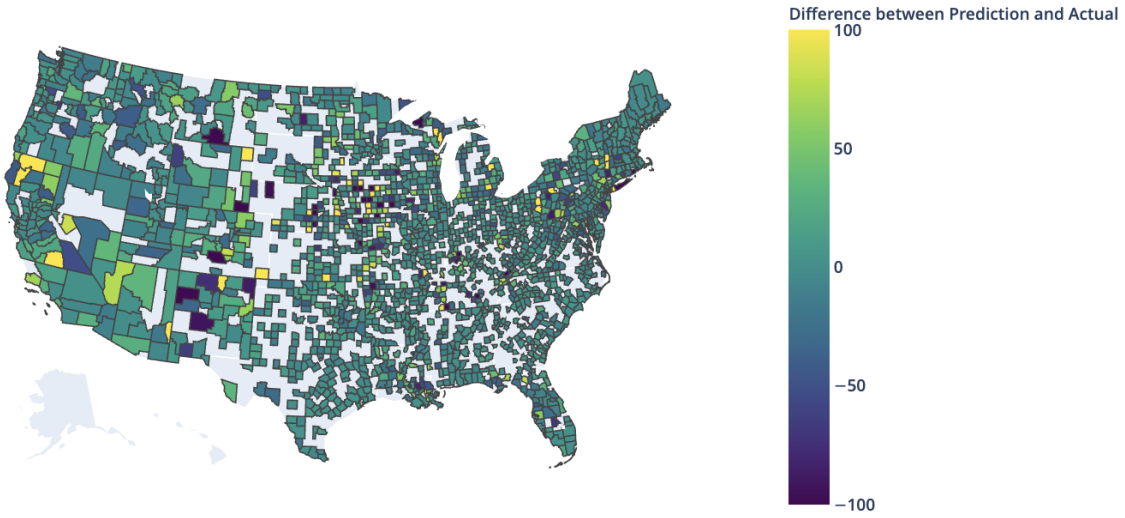

Figure 5: Choropleth map of the continental US showing the difference between our predicted dollars saved using our Gradient Boosted Trees model and the actual dollars saved.

When assessing the prediction error, we noticed that there were a few areas where the error was greater than $300. These areas were places like Isle La Motte, Vermont (error: $435.67), Fishers Island, New York (error: -$636.27), Fawnskin New York (error: -$707.89), Block Island, Rhode Island (error: -$1178.23). We noticed that through the map that the RMSE was giving a high weight to large errors. Due to the map showing generally decent predictions and the large error, we also assessed the Mean Absolute Error (MAE) and this was $18.11.

When assessing the feature importances from the Gradient Boosted Trees model, features related to consumption (the annual kWh used) and location (longitude and latitude) had the highest feature importances. Longitude and annual kWh used also had the highest feature importances for the Random Forest Regressor model.

]:

	feature	score
6	annual_kwh_used	0.195357
1	lng	0.127142
0	lat	0.051254
17	temp_Oct	0.048217
26	pct_cloudy_days_Mar	0.048105
21	pct_cloudy_days_Dec	0.039097
4	elevation	0.036069
22	pct_cloudy_days_Feb	0.032837
25	pct_cloudy_days_Jun	0.028367
15	temp_May	0.027375

Figure 6: The top 10 most significant features extracted from the Gradient Boosted Trees model.

Conclusion

The Gradient Boosted Trees model did an effective job at predicting the dollars saved from using solars panels. We were able to determine that factors related to location and consumption were crucial to predicting the dollars saved. In terms of future work, considering that the current solar panel data came from Canadian solar panel model that has been discontinued, working with data from more current solar panels seems to be a good next path to pursue. It would also be interesting to be able to use maintenance cost data for solar panels. In terms of expanding the scope of the project, it would be interesting to take into consideration data from outside the United States of America.

Bibliography

National Renewable Energy Laboratory. “NREL GIS Data: Continental United States Direct Normal Solar Resource 10km Resolution (1998 - 2005).” *OpenEI*, https://openei.org/datasets/dataset/nrel-gis-data-continental-united-states-high-resolution-direct-normal-solar-resource. Accessed 8 March 2021.

National Renewable Energy Laboratory. “NREL's PVWatts® Calculator.” *PVWatts® Calculator*, National Renewable Energy Laboratory, https://pvwatts.nrel.gov/index.php. Accessed 8 March 2021.

SimpleMaps. “US Zip Codes Database.” *SimpleMaps*, https://simplemaps.com/data/us-zips. Accessed 8 March 2021.

University of East Anglia Climatic Research Unit, et al. “CRU TS4.00: Climatic Research Unit (CRU) Time-Series (TS) version 4.00 of high-resolution gridded data of month-by-month variation in climate (Jan. 1901- Dec. 2015).” Centre for Environmental Data Analysis, 25 August 2017, http://dx.doi.org/10.5285/edf8febfdaad48abb2cbaf7d7e846a86. Accessed 8 March 2021.

U.S. Board on Geographic Names. “Geographic Names Information System (GNIS) Data.” *United States Geological Survey*, https://www.usgs.gov/core-science-systems/ngp/board-on-geographic-names/download-gnis-data. Accessed 8 March 2021.

US Department Of Energy. “U.S. Electric Utility Companies and Rates: Look-up by Zipcode (2018).” *OpenEI*, https://openei.org/doe-opendata/dataset/u-s-electric-utility-companies-and-rates-look-up-by-zipcode-2018. Accessed 8 March 2021.

US Energy Information Administration. “Table C14. Total Energy Consumption Estimates per Capita by End-Use Sector, Ranked by State, 2018.” *eia*, https://www.eia.gov/state/seds/data.php?incfile=/state/seds/sep_sum/html/rank_use_capita.html&sid=US. Accessed 8 March 2021.

William F. Holmgren, Clifford W. Hansen, and Mark A. Mikofski. “pvlib python: a python package for modeling solar energy systems.” Journal of Open Source Software, 3(29), 884, (2018). https://doi.org/10.21105/joss.00884