Predicting Solar Power Generation from Weather Forecasts Using Machine Learning Models

Kushal Juneja
kushal19057@iiitd.ac.in

Naval Kumar Shukla
naval19065@iiitd.ac.in

Rishi Singhal
rishi19194@iiitd.ac.in

Udit Narang
udit19120@iiitd.ac.in

Abstract

Predicting future renewable energy generation is important, since the power grid must use fossil fuel based energy as demand varies. We explore prediction models for solar power generation from NSRDB database using machine learning techniques. We compare multiple regression techniques for generating prediction models.

1. Introduction

Energy from the sun is an inexhaustible resource that creates little to no pollution. However, current methods of collecting solar energy and converting it to electricity can capture and store only a small amount of the available energy from the sun at any given time. The highly variable nature of solar energy production puts stress on fossil fuel based power generation. We aim to predict solar intensity for a given area 48 hours into the future using local time-series weather data. Our goal is to provide high-confidence forecasts of solar generation (via solar intensity) using readily-available weather data. This will enable better regulation of fossil fuel based power generation. We have acquired solar intensity and weather data from NSRDB[1]. This project aligns with our sustainable development goals and deploying machine learning techniques to solve real world problems.

2. Literature Survey

Navin Sharma et al. compare multiple regression techniques for generating prediction models, including linear least squares regression and support vector machines using multiple kernel functions [3].

Abuella and Choudhary propose a solar energy prediction model using Artificial Neural Network's (ANN's)[2].

3. Dataset

3.1. General Information

The NSRDB dataset includes both observed weather data (temperature, pressure, cloud cover, solar zenith angle ,etc.) and solar intensity data measured in watts per square meter. The dataset includes several solar radiation measures such as Diffused Normal Irradiance (DNI), Diffused Horizontal Irradiance (DHI) and Global Horizontal Irradiance (GHI). We choose to include GHI measurements since it incorporates DHI, DNI and ambient solar radiation reflected from nearby surfaces. This makes it a good indicator for solar panel readings.

The NSRDB data is measured once every 10 minutes. We investigate a single location - Las Vegas, Nevada, USA. We limit our analysis to the year 2019 only.
Our dataset contains more than 50,000 distinct observations, each with 16 features (including Time values such as Month, Day, Hour and Minute) shown in table 1 with corresponding measure of GHI.

Month	Day	Hour
Minute	Temperature	Cloud Type
Fill Flag	Surface Albedo	Ozone
Pressure	Dew Point	Precipitable Water
Wind Direction	Wind Speed	Relative Humidity
	Solar Zenith Angle	

3.2. Exploratory Data Analysis

We plot distribution of each feature using histograms, pie-charts and box-plots. We also study pairwise correlation between features using correlation-matrix & skew index table.

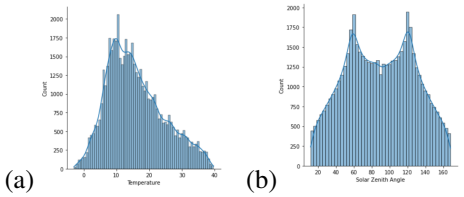


Figure 1: Plots: (a) Temperature (b) Solar Zenith

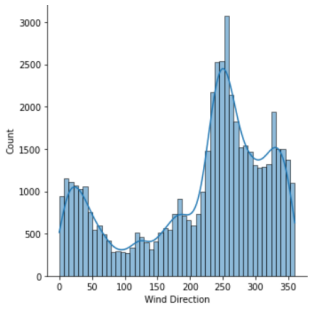


Figure 2: Wind Direction

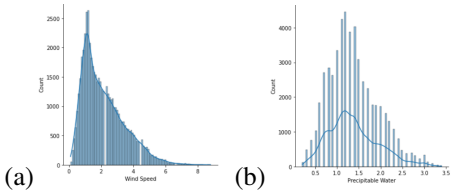


Figure 3: Plots: (a) Wind Speed (b) Precipitable Water

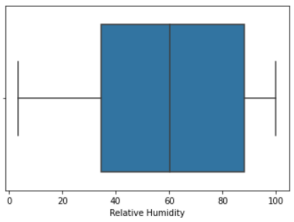


Figure 4: Relative Humidity

Feature	Skew Index
Month	-0.01
Day	0.07
Hour	0
Minute	0
Temperature	0.55
Cloud Type	1.30
Dew Point	-1.34
Fill Flag	3.96
Ozone	0.78
Relative Humidity	-0.091
Solar Zenith Angle	-0.00013
Surface Albedo	-0.71
Pressure	0.046
Precip. Water	0.62
Wind Direction	-0.73
Wind Speed	0.99

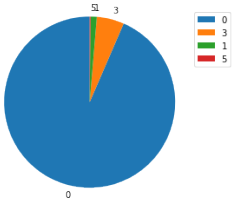


Figure 5: Fill Flag

3.3. Data Preprocessing

We spotted outliers in the distribution curves. However, we assumed that they are the natural part of the weather observations we are studying, hence we didn't remove them. Also we didn't find any missing/NaN values in our dataset, which would result in imbalance observations.

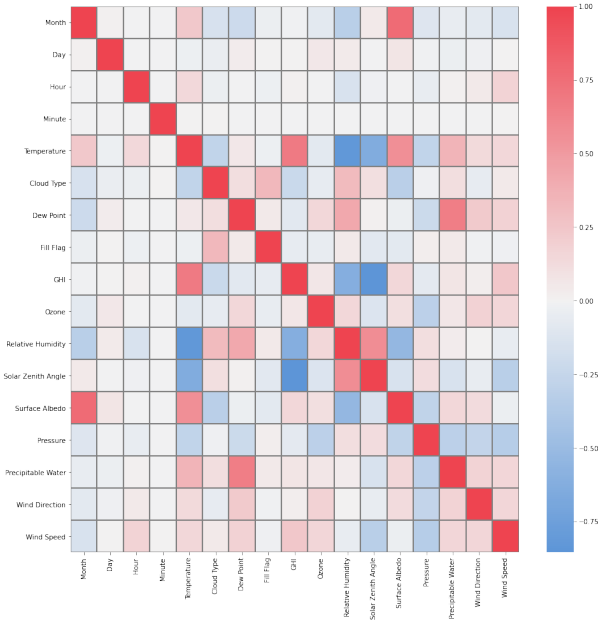


Figure 6: Correlation Matrix

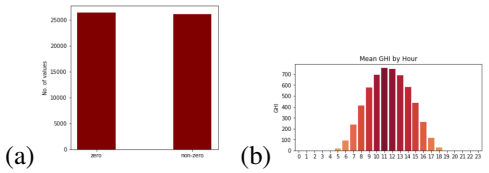
The Correlation Matrix shows that features 'Dew Point' and 'Precipitable Water' are highly correlated. Since highly correlated features have almost the same effect on the dependent variable, we drop one of them. We test the performance of our baseline model by dropping both of the features one by one. Based on the results, we choose to drop 'Dew Point'.

The Pie-Chart for 'Fill Flag' feature shows that more than 90% observations correspond to '0'. It indicates that the value is not available. Therefore 'Fill Flag' adds negligible information to the dataset. Hence, we drop 'Fill Flag'.

3.4. Dataset Preparation

Our task is to predict solar intensity values 48 hours into the future. Thus, we establish a one-to-one mapping between current weather observations and GHI values 48 hours into the future.

4. Methadology



(a) (b)

Figure 7: Plots: (a) GHI Sparsity (b) Mean GHI by Hour

NOTE : We have a sparse dataset with around 50% GHI values equal to 0. The zero GHI values are corresponding to the time before sunrise and after sunset. Thus, we trained our model using the dataset corresponding to the non-zero GHI values only.

Once we have trained our model, we are predicting the GHI values for the weather conditions of any given time. While prediction, we are giving the dataset including the zero and non-zero GHI values to the model obtained. To resolve the zero GHI values, we have checked the Hour feature, if the Hour is less than 7 a.m.(before sunrise) or greater than 5 p.m.(after sunset), we then predicted the value of GHI will be 0 for those data points.

To test the performance of our models we have the used RMSE metrics which is as follows:-

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

To train our prediction model, we explore 2 different methods:-

4.1. Without Feature Expansion

We split the dataset into 80:20 train-test split. Then we trained our model using different techniques like Linear Regression, Ridge Regression(L2), Lasso Regression(L1). We also used the Polynomial Regression with degrees 2,3,4 & 5. (We couldn't perform Polynomial Regression with degree greater than 5 due to unavailability of sufficient RAM on Google Colab). After training our models, we predicted on the train and test set & computed the train and test RMSE values and compared them for different models.

4.2. With Feature Expansion

At first we modified our dataset by using data from previous time points as new features.(feature expansion). After that We split the modified data into 80:20 train-test split. We used feature expansions of values 1,2,5,15,30,45,55,60 and used the following techniques to predict on our test and train set:-

4.2.1 Baseline Linear Regression

We train the model using Linear Regression and found out the the predicted values on our train and test sets and computed the RMSE values.

4.2.2 Principal Component Analysis (PCA)

With increase in number of expansions, the number of features increased.Thus during training the model complexity increased by manyfolds. We apply PCA to reduce the dimensions of the dataset keeping the explained variance sufficiently high. This allowed to train a less complex model.

5. Results and Analysis

5.1. Without Feature Expansion

Model	Train RMSE	Test RMSE
Baseline LR	86.69	87.36
Lasso	86.70	87.36
Ridge	86.71	87.36
SGD	86.70	87.48

In the baseline LR model without feature expansion, we observed low variance. Since the model isn't over-fitting, Lasso and Ridge didn't helped to improve the performance. We also applied SGD Regressor and observed a similar result as that in baseline LR model.

Degree	Train RMSE	Test RMSE
2	82.9	82.99
3	71.92	73.05
4	64.32	70.17
5	54.35	554.42

We then applied polynomial regression with different degrees, and observed that both train and test RMSE values decrease till degree 4. From degree 4 onwards, as the model complexity increases, the performance worsens. We noted that our model does not generalise well from degree 5, hence resulting in over-fitting. We are not able to go beyond degree 5 due to the limitation of resources.

5.2. Feature Expansion without PCA

No. of Expansions	Train RMSE	Test RMSE
1	88.95	90.50
2	89.35	89.13
5	88.51	89.51
15	84.56	85.37
30	85.99	83.41
45	81.36	83.27
55	81.36	81.15
60	81.51	87.09

On applying feature expansion without PCA, we observed that the increase in the number of the expansions results in the decrease of both test and train RMSE values. However, the variance started to increase after 60 features expansions as our model started to overfit.

5.3. Feature Expansion with PCA

#	PCA components	Train RMSE	Test RMSE
1	15	88.95	90.50
2	20	88.63	89.43
5	35	89	89.9
15	100	86.2	86.79
30	180	84.5	84.89
45	300	82.89	83.64
55	350	81.78	83.13
60	350	81.76	88.79

NOTE: # refers to No. of feature expansions.

Now, we tried to decrease the dimensions of our data set keeping the explained variance sufficiently large,i.e., ensuring not much information is lost. After applying PCA for the same number of feature expansions, we see that we could obtain approximately the same RMSE values on the test train set but with less dimensions, thus helping us to decrease the complexity of our model.

5.4. Plot of actual v/s predicted GHI values for our best model

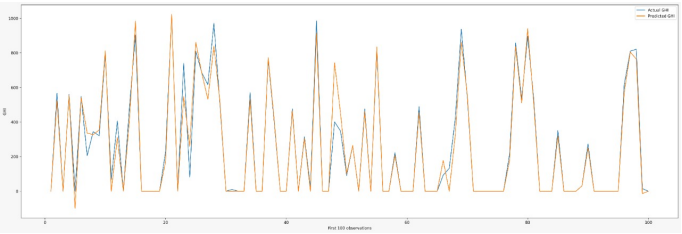


Figure 8: Result Plot

6. Conclusion

6.1. Learning's

- Able to apply Data Preprocessing and Exploratory Data Analysis (EDA) techniques on datasets.
- Able to apply different regression models such as linear, lasso, ridge, and polynomial for prediction
- Able to apply different feature reduction techniques such as Principal Component Analysis (PCA)
- Able to apply different feature engineering and feature expansion techniques
- Able to argue for model performance and bias-variance trade-offs.

6.2. Work Left

- Handling zero GHI values using classification models
- Apply SVM regression
- Apply Regression trees
- K-Means Clustering

6.3. Member Contribution

Each member has contributed to the following areas :

Rishi: EDA, Polynomial Regression, Feature Expansion, SGD Regressor
Udit: Data Pre-processing, Feature Expansion, Polynomial Regression, SGD Regressor
Naval: Data Pre-processing, Baseline LR, Lasso, Ridge, PCA
Kushal: EDA, Baseline LR, Lasso, Ridge, PCA

References

[1] https://nsrdb.nrel.gov.

[2] M. abuella and b. chowdhury, solar power forecasting using artificial neural networks, 2015 north american power symposium (naps), 2015, pp. 1-5, doi: 10.1109/naps.2015.7335176.

[3] N. sharma, p. sharma, d. irwin and p. shenoy, predicting solar generation from weather forecasts using machine learning, 2011 ieee international conference on smart grid communications (smartgridcomm), 2011, pp. 528-533, doi: 10.1109/smartgridcomm.2011.6102379.