Solar Radiation Prediction

NIDHI AGARWAL, University of Pittsburgh
POOJA GHATGE, University of Pittsburgh
SHASHANK BR, University of Pittsburgh

XX

Thirty-two thousand six hundred eighty-seven recordings of eleven parameters have been provided by HI-SEAS weather station, NASA for the duration of September-December 2016. These meteorological data will be analyzed and with enough data, will be able to predict the solar-radiation level.

KEYWORDS
Machine Learning, Regression, xgboost

1 INTRODUCTION

HI-SEAS or Hawaii Space Exploration Analog and Simulation wish to make solar batteries. They are setting up base camps to trace solar energy for use in Mars space project. Our dataset is part of the data collected during missions HI-SEAS IV and HI-SEAS V. We aim to utilize machine learning algorithms on this dataset to predict solar radiation levels. So that they can evaluate the best time and day of month to trace solar energy .Along with this, they can evaluate how long the battery will run.

2 DATASET

It consists of 11 columns and 32687 rows. Columns are represented by measurement criteria and rows contain measurement values recorded. Figure 1 shows a small subset of the database containing 15 rows. The features are:

1. UNIXTime: Number of seconds elapsed since Unix epoch (January 1st, 1970) and time when the measurement was recorded.
2. Data: Date and time when the measurement was recorded.
3. Time: Time of day in HH: MM: SS, when the measurement was recorded.
4. Radiation: Solar radiation measured in watts per meter square when the measurement was recorded.
5. Temperature: Temperature, measured in degrees Fahrenheit when the measurement was recorded.
6. Pressure: Barometric pressure, measured in Mercury level (Hg) when the measurement was recorded.
7. Humidity: Percentage of humidity present when the measurement was recorded.
8. WindDirection: Specified in degrees when the measurement was recorded.
9. WindSpeed: Speed of the wind, measured in miles per hour when the measurement was recorded.
10. TimeSunRise: Sunrise time in HH:MM:SS, at Hawaii on the day when the measurement was recorded.
11. TimeSunSet: Sunset time in HH:MM:SS, at Hawaii on the day when the measurement was recorded.

	UNIXTime	Data	Time	Radiation	Temperature	Pressure	Humidity	WindDirection(Degrees)	Speed	TimeSunRise	TimeSunSet
0	1475229326	9/29/2016 12:00:00 AM	23:55:26	1.21	48	30.46	59	177.39	5.62	06:13:00	18:13:00
1	1475229023	9/29/2016 12:00:00 AM	23:50:23	1.21	48	30.46	58	176.78	3.37	06:13:00	18:13:00
2	1475228726	9/29/2016 12:00:00 AM	23:45:26	1.23	48	30.46	57	158.75	3.37	06:13:00	18:13:00
3	1475228421	9/29/2016 12:00:00 AM	23:40:21	1.21	48	30.46	60	137.71	3.37	06:13:00	18:13:00
4	1475228124	9/29/2016 12:00:00 AM	23:35:24	1.17	48	30.46	62	104.95	5.62	06:13:00	18:13:00
5	1475227824	9/29/2016 12:00:00 AM	23:30:24	1.21	48	30.46	64	120.20	5.62	06:13:00	18:13:00
6	1475227519	9/29/2016 12:00:00 AM	23:25:19	1.20	49	30.46	72	112.45	6.75	06:13:00	18:13:00
7	1475227222	9/29/2016 12:00:00 AM	23:20:22	1.24	49	30.46	71	122.97	5.62	06:13:00	18:13:00
8	1475226922	9/29/2016 12:00:00 AM	23:15:22	1.23	49	30.46	80	101.18	4.50	06:13:00	18:13:00
9	1475226622	9/29/2016 12:00:00 AM	23:10:22	1.21	49	30.46	85	141.87	4.50	06:13:00	18:13:00
10	1475226323	9/29/2016 12:00:00 AM	23:05:23	1.23	49	30.47	93	120.55	2.25	06:13:00	18:13:00
11	1475226025	9/29/2016 12:00:00 AM	23:00:25	1.21	49	30.47	98	144.19	3.37	06:13:00	18:13:00
12	1475225720	9/29/2016 12:00:00 AM	22:55:20	1.22	49	30.47	99	139.80	6.75	06:13:00	18:13:00
13	1475225419	9/29/2016 12:00:00 AM	22:50:19	1.21	50	30.47	99	140.92	2.25	06:13:00	18:13:00
14	1475225131	9/29/2016 12:00:00 AM	22:45:31	1.23	50	30.47	99	147.61	5.62	06:13:00	18:13:00
15	1475224823	9/29/2016 12:00:00 AM	22:40:23	1.22	50	30.47	99	113.78	4.50	06:13:00	18:13:00

Fig. 1. First 15 rows of the dataset.

3 SOLUTION

3.1 Data Pre-processing

As shown in Figure 2, dataset provided by HI-SEAS, NASA did not have any missing values. We convert the HH:MM:SS format into seconds passed since 00:00:00 to make it easier for computation.

```
1 # checking for missing values
2 df1.isnull().sum()
```

```
Unnamed: 0                0
UNIXTime                  0
Time                      0
Temperature               0
Pressure                  0
Humidity                  0
WindDirection(Degrees)    0
Speed                     0
TimeSunRise               0
TimeSunSet                0
day_length                0
target_radiation          0
dtype: int64
```

Fig. 2. List of missing values in each column.

Columns UNIXTime and Time both tells us about the time when the measurements were recorded meaning, having two columns representing same value is redundant so, we drop UNIXTime column. We create a new column named day_length which is difference between TimeSunRise and TimeSunSet and drop TimeSunRise and TimeSunSet. Since time when measurement recorded is given in Time column, we don't need to extract that information from Data column. We make a new column by extracting the month and date from Data column and we drop Data column. We combine the WindDirection(Degrees) and Speed to create a new column called Wind_speed containing effective wind speed. We also move the Radiation column to the last position. Figure 3 shows the first 15 rows of the pre-processed dataset.

	Time	Temperature	Pressure	Humidity	Day_length	Month	Day_of_month	Wind_speed	Radiation
0	8288	48	30.46	59	43200	3	21	0.616870	1.21
1	8257	48	30.46	58	43200	3	21	2.222088	1.21
2	8228	48	30.46	57	43200	3	21	-0.335000	1.23
3	8194	48	30.46	60	43200	3	21	2.924422	1.21
4	8168	48	30.46	62	43200	3	21	-1.625104	1.17
5	8141	48	30.46	64	43200	3	21	3.836223	1.21
6	8108	49	30.46	72	43200	3	21	5.384431	1.20
7	8085	49	30.46	71	43200	3	21	-5.065668	1.24
8	8056	49	30.46	80	43200	3	21	3.585003	1.23
9	8027	49	30.46	85	43200	3	21	-3.952718	1.21
10	8001	49	30.47	93	43200	3	21	0.878920	1.23
11	7975	49	30.47	98	43200	3	21	3.195448	1.21
12	7938	49	30.47	99	43200	3	21	0.005893	1.22
13	7907	50	30.47	99	43200	3	21	-2.024369	1.21
14	7885	50	30.47	99	43200	3	21	-5.614347	1.23
15	7853	50	30.47	99	43200	3	21	3.491526	1.22

Fig. 3. First 15 rows of the pre-processed dataset.

3.2 Data Visualization

Figure 4 shows the heatmap of correlation between all the features. Figures 5.1 – Figure 5.7 shows the scatter plot of all features versus our target feature, Radiation.

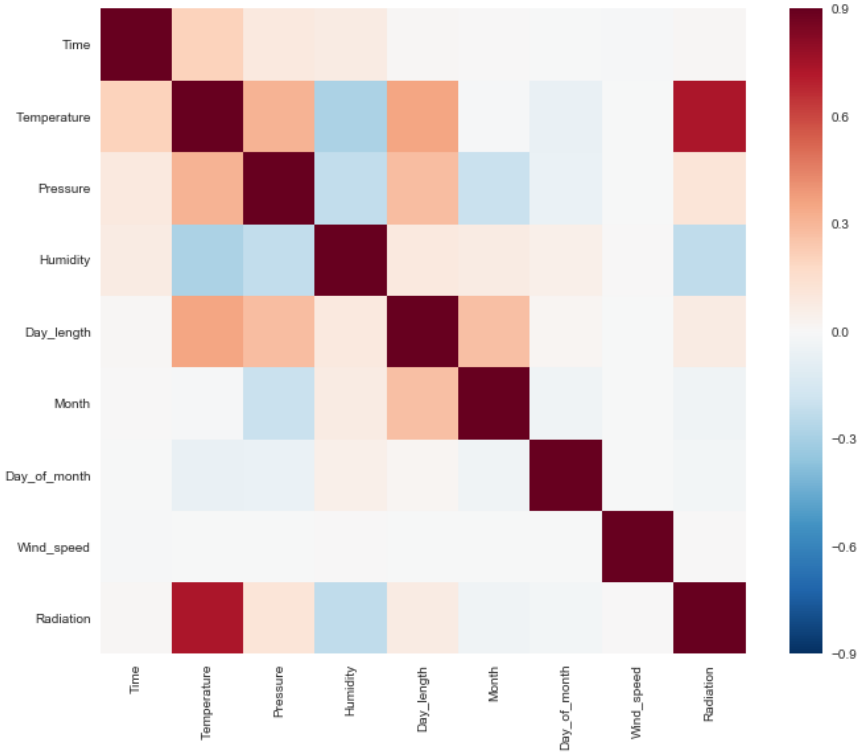

Fig. 4. Heatmap of correlation matrix.

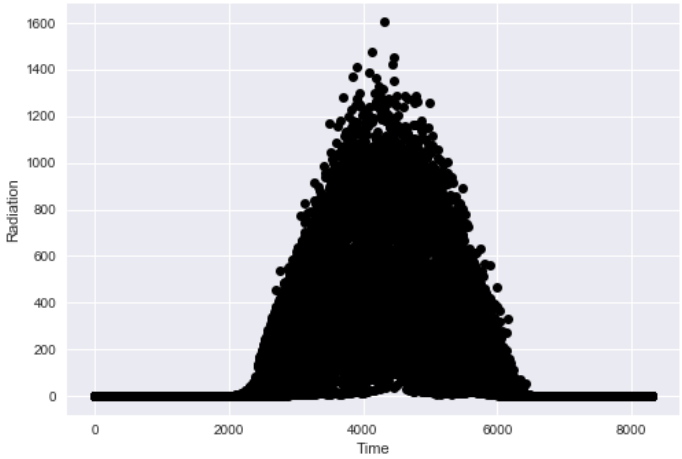

Fig. 5.1. Radiation v/s Time.

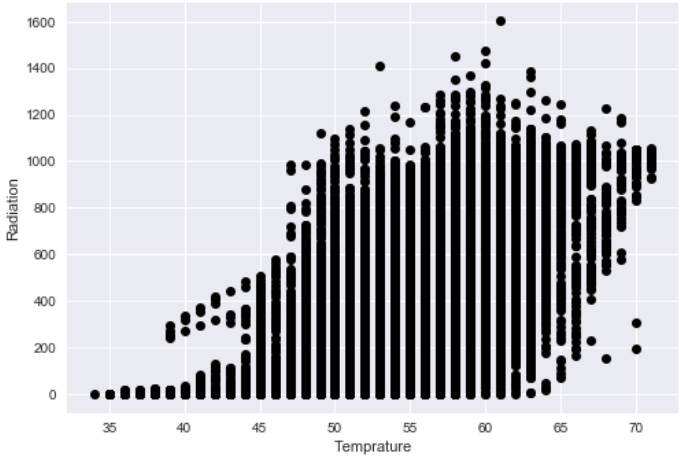

Fig. 5.2. Radiation v/s Temperature.

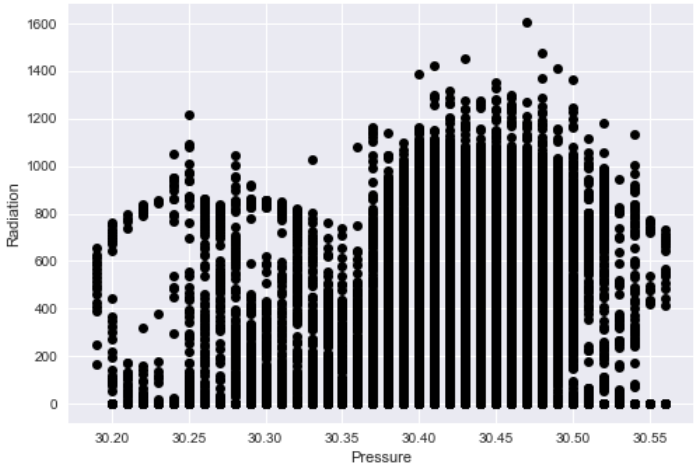

Fig. 5.3. Radiation v/s Pressure.

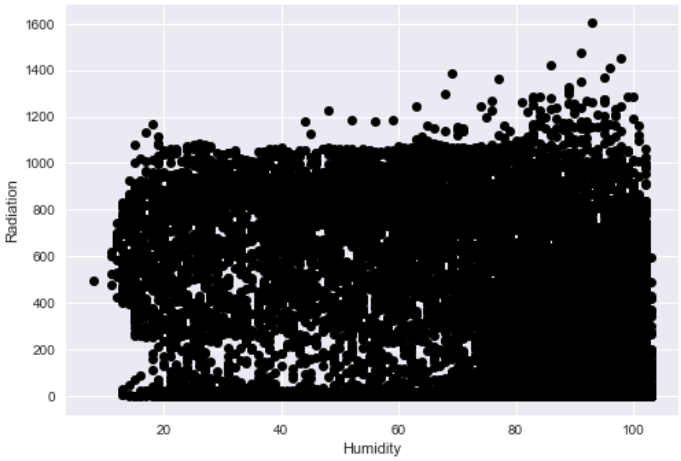

Fig. 5.4. Radiation v/s Humidity.

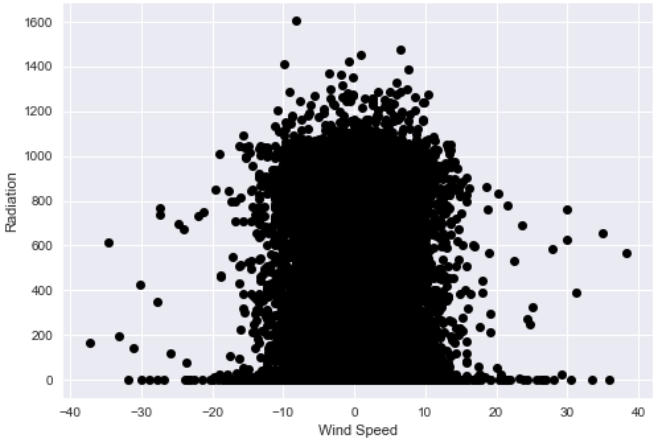

Fig. 5.5. Radiation v/s Wind speed.

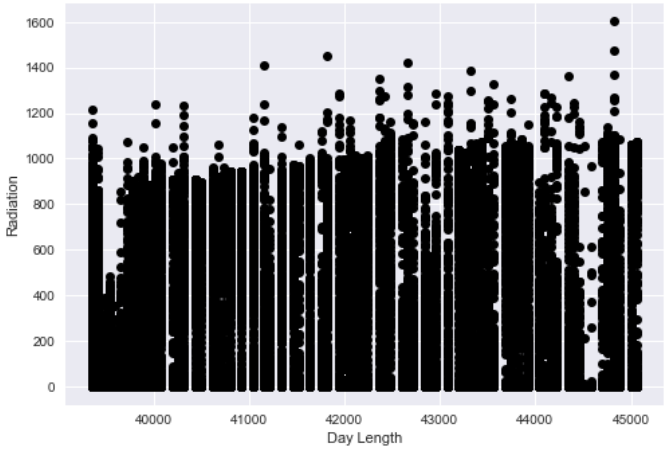

Fig. 5.6. Radiation v/s day_length.

From the above scatter plots, we can observe Time vs Radiation is a perfectly skewed graph, there is a fairly strong correlation between Radiation-Temperature and Radiation-Wind_speed.

3.3 Model Fitting

We use 'train_test_split' package from 'sklearn.model_selection' to split our data into train set and test set in the ratio of 4:1. We scale our data using 'MinMaxScaler()' from 'sklearn.preprocessing' package.

We start with a Linear model. We fit considering all 8 features and the score obtained was 0.623 which was not satisfactory. We then try to fit with a polynomial model with degree 3 and the score improved quite a bit to 0.738 but it was still not good enough. We try to fit various models and compare their scores. Table 1 shows the models we have fitted with their respective scores and MSE. Models which resulted in significant milestones in our project have been briefly described below.

	Score	Mean Squared Error
Linear Model	0.623	37383
Polynomial Model	0.738	26010
Lasso Model	0.627	36505
ENET Model	0.627	36504
Support Vector Regressor	-0.413	138158
ADA Boost	0.928	1.389
GBoost	0.983	0.327
Decision Tree Regressor	0.887	19211
Neural Network (MLP)	0.804	19211
Random Forest Regressor	0.938	6181
Gradient Boosting Regressor	0.983	0.327
XGBoost	0.989	0.012

Table. 1. Models with respective scores and MSE.

1. **Linear Model:**

   Linear regression is a linear model, e.g. a model that assumes a linear relationship between the input variables (x) and the single output variable (y).

   More specifically, that y can be calculated from a linear combination of the input variables (x). Because we had 8 distinct features, we need to use Linear Regression Model with multiple features.

   Multiple linear regression attempts to model the relationship between two or more explanatory variables and a response variable by fitting a linear equation to observed data. Every value of the independent variable (x) is associated with a value of the dependent variable (y_1, y_2… y_n).

   We fitted our data into a linear model and obtained score of 0.623 which was bad but not surprising considering we did not see any linear relation between features and the target during data visualization phase. To capture any polynomial relation between features, we also fit our data using polynomial model up to degrees = 3.

   New linear model score was 0.738. It was an improvement but still not good enough.

2. **Neural Network:**

   We have implemented neural network using Multilayer Perceptron model. A Multilayer Perceptron (MLP) is a class of feedforward artificial neural network.

   An MLP consists of at least three layers of nodes. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training.

   It's multiple layers and non-linear activation distinguishes MLP from a linear perceptron. It can distinguish data that is not linearly separable. This would be perfect since the data is not linearly separable.

   Unfortunately, we got a score of 0.804. Because we have only 13 features, neural network is not going to be effective.

3. **Random Forest:**

   Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

   Random decision forests correct for decision trees' habit of overfitting to their training set. Scatter plots show the distribution of the data has some margins which means decision trees will work well which means Random Forest model would be a very good fit for our data.

   Using random forest model, we could obtain score of 0.938.

4. **XGBoost:**

   XGBoost (Extreme gradient boosting) is a gradient boosting algorithm. The principle used in XGboost is similar to gradient boosted trees. It uses additive training approach to boost performance of the algorithm. The model uses combination of Regression and classification trees (CART). In such models the leaf nodes are associated with additional weights of positive classification known as leaf scores.

The ultimate respective leaf score of the ensemble tree is the respective sum of the leaf scores in set of trees under consideration. Standard XGBoost worked very well on our dataset. It has various additional features like tree pruning, built-in cross validation and regularized boosting. Since we are implementing regularized boosting we can now easily control and avoid overfitting.

We got a score of 0.989 using XGBoost model.

REFERENCES

[1] https://machinelearningmastery.com/4-steps-to-get-started-in-machine-learning
[2] https://www.kaggle.com/hguimaraes/pysolar-dataset-exploration-and-train-test
[3] https://www.toptal.com/machine-learning/machine-learning-theory-an-introductory-primer
[4] https://towardsdatascience.com/understand-these-5-basic-concepts-to-sound-like-a-machine-learning-expert-6221ec0fe960
[5] https://machinelearningmastery.com/basic-concepts-in-machine-learning/

Received December 2017