Machine Learning Algorithm to Predict Solar Radiation

Sreeja Vadakke Veettil

08/03/2022

Executive Summary

This report is related to the second Capstone project of the HarvardX-PH125.9x:Capstone course. The aim of this project is to apply machine learning algorithms that go beyond standard linear regression techniques on a publicly available data set and to clearly communicate the process and insights gained from the analysis. In this context, the data set chosen for this project is the meteorological data collected from the Hawaii Space Exploration Analog and Simulation (HI-SEAS) weather station during four months (September to December 2016) between Mission IV and Mission V of the NASA HI-SEAS mission. This data set was a part of the NASA Space Apps Challenge hackathon 2017, where participants were asked to predict solar radiation given a set of measurable meteorological conditions.

The report starts with a general introduction to the project followed by a description of the data set. A preliminary data analysis is carried out to better understand the parameters to be included in the machine learning algorithm. This is followed by a description and results of the different algorithms, before deciding on the final algorithm. The testing on the final validation data set revealed comparable performance for three algorithms, namely Random Forest(RF), Extreme Gradient Boosting (XGBoost) and an Ensemble algorithm (built by ensembling the Support Vector Regression (SVR), RF, Stochastic Gradient Boosting (SGB), XGBoost algorithms), in predicting the solar radiation.

Contents

Introduction

The Hawaii Space Exploration Analog and Simulation (HI-SEAS) was a planetary surface exploration analog habitat for human spaceflight to Mars located at approximately 8,500 feet above sea level on the Mauna Loa side of the saddle area on the island of Hawaii. HI-SEAS acted as a test and training bed for humans as the capability to explore Mars was being developed. Six HI-SEAS missions of extended duration from four months to a year were funded during 2013 to 2017 by research grants from the NASA's Human Research Program and the University of Hawaii. One of the NASA Space Apps Challenge hackathon in 2017 focused on the use of the data collected from the HI-SEAS site to predict the level of solar radiation given a set of measurable meteorological conditions. The original data set is available at https://www.kaggle.com/dronio/SolarEnergy.

A large solar array and battery bank installed at the HI-SEAS site were the only available power source. On sunny days, the array collected enough energy to power the entire site and recharge the battery bank. On overcast days and at night, the battery bank was only sparingly used as there was a strict power budget for operations each day to ensure that vital equipment stays online. Information on the conditions most favorable for the incident solar radiation is crucial for deciding when to deploy solar energy harvesting equipment, especially for colonists or astronauts on the surface of Mars.

The aim of this Capstone project is to develop a machine learning algorithm using the meteorological data collected during September to December 2016 from the HI-SEAS site and made available by NASA. To facilitate the algorithm development, the original data set is initially split into a training set and a final hold-out test set (validation data set). The algorithm is developed using the training data set and then used to predict the level of solar radiation in the validation data set. The R-Squared (R^2) and Root Mean Squared Error (RMSE) are used to evaluate how close the final algorithm predictions are to the actual values in the validation data set.

Data description and preliminary analysis

The original data set consists of 32,686 rows and 11 columns. Each row in the data set represents the time stamp within each day, when the values for the meteorological variables are available. The fields in the data set are: UNIX time (seconds since Jan 1, 1970); Solar radiation (watts/meter^2), Temperature (degrees Fahrenheit); Humidity (%); Barometric pressure (Hg); Wind direction (degrees); Wind speed(miles/hour); Sunrise/sunset time (Hawaii timezone). The original data set is first edited, checked and conditioned before applying the machine learning algorithm. An initial check of the data set indicated that there are no missing values, and therefore there is no need to deal with missing data.

As part of the preliminary analysis to format the data, the UNIX time is converted into datetime in the correct timezone, i.e. Hawaii Standard Time (HST). It is to be noted that while converting UNIX time to datetime, the UTC timezone is assumed, and as this data was collected under the HST timezone, this needs to be updated. The sunrise and sunset times are formatted to datetime in the correct timezone and the length of the daylight hours for each day given by the difference between the sunrise and sunset times is calculated. As it is well known that the level of solar radiation changes according to the position of the sun in the sky, the relative time of the day when the data was collected is also estimated. Date and time of the day are independent variables, as their values are unaffected by the other variables. The meteorological variables such as temperature, pressure and humidity do not directly affect one another significantly, but since they are all properties of the local atmosphere, they do not vary independently from one another. Furthermore, it is expected that these variables will have a relationship with time of day.

The diurnal variation of all the parameters on 01 October 2016 is shown in Figure 1. The sunrise and sunset times on this day are shown respectively by red and blue dotted lines. It can be observed that as expected the value of solar radiation is approximately zero before sunrise and after sunset, with a high variability during the daylight hours. A clear diurnal variation in the temperature is observed, with values increasing after sunrise and decreasing after sunset. The pressure shows a sinusoidal variation over the day and no clear diurnal variation is observed for the other parameters such as humidity, wind direction and speed. The

variation in the wind direction and speed over a day is extremely irregular. It is also clear from the figure that temperature, pressure, humidity and wind speed assumes only discrete values. This could be related to the type of sensors used in the data collection.

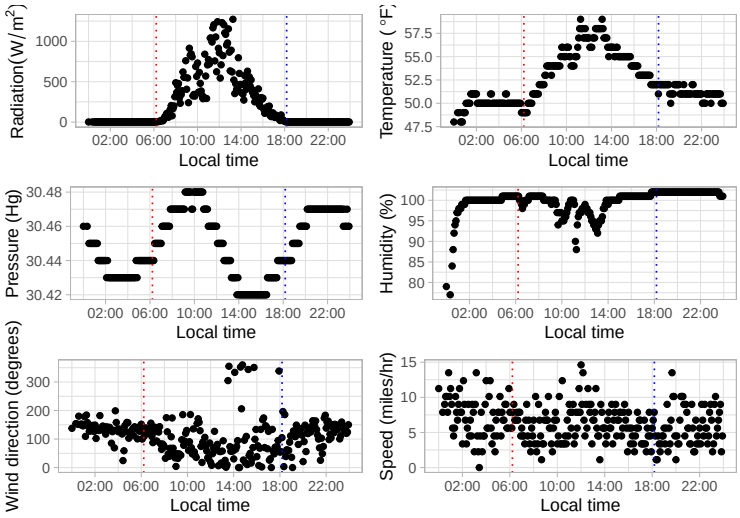

Figure 1: Diurnal variation of parameters on October 01 2016

The next step in the preliminary analysis is to plot the distribution of all the parameters to better understand the range of values for the parameters and to validate if they are reasonable.

Solar radiation

On analyzing the data, the solar radiation assumes only positive values, with the minimum and maximum values of 1.11 W/m^2 and 1601.26 W/m^2 respectively. The overall average value of the solar radiation is 207.125 W/m^2. Figure 2 shows the distribution of the solar radiation and it is clear that the distribution is right skewed with roughly 68% of values located between 0 and 200 W/m^2. This is expected, as the values of solar radiation before sunrise and after sunset are small.

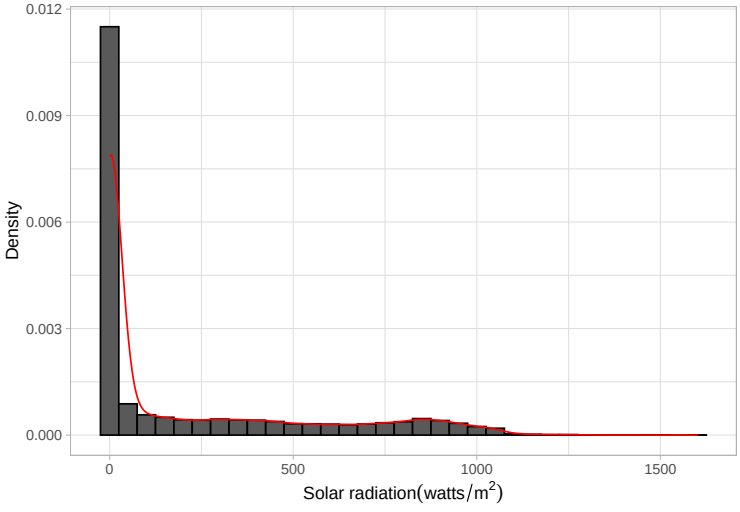

Figure 2: Distribution of solar radiation

Temperature

The values of the temperature ranges between 34°F and 71°F. Figure 3 shows a right skewed distribution for temperature, with ~ 66% of values located between 45 and 55°F.

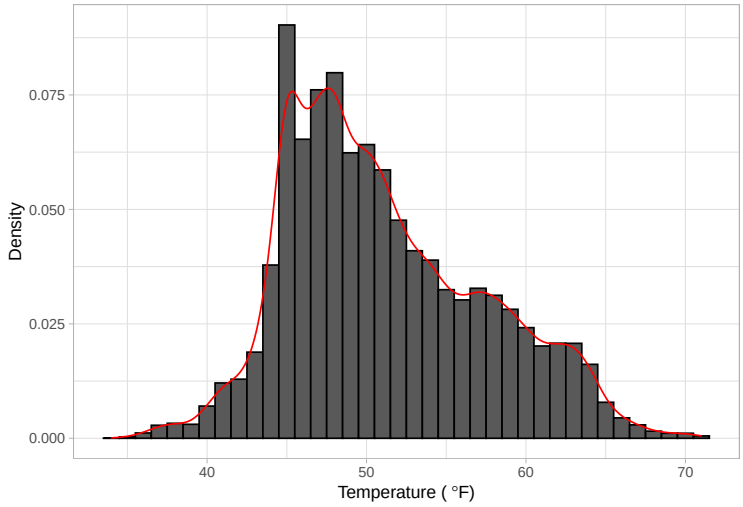

Figure 3: Distribution of temperature

Pressure

The distribution of pressure is shown in Figure 4, which is slightly left skewed. The variation in the pressure is very little (30.19-30.56 Hg), with values mostly around 1 bar.

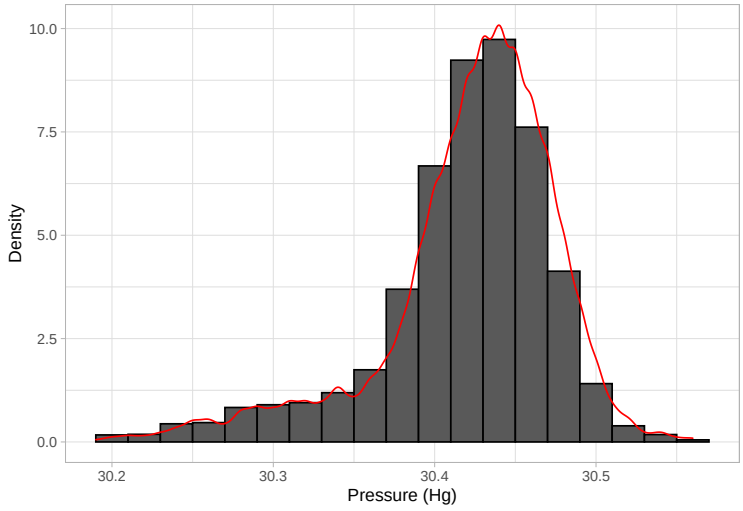

Figure 4: Distribution of pressure

Humidity

Figure 5 shows the distribution of humidity, the minimum and maximum value is 8 and 103% respectively. The distribution is left skewed and ~ 57% of the humidity values are above 80%.

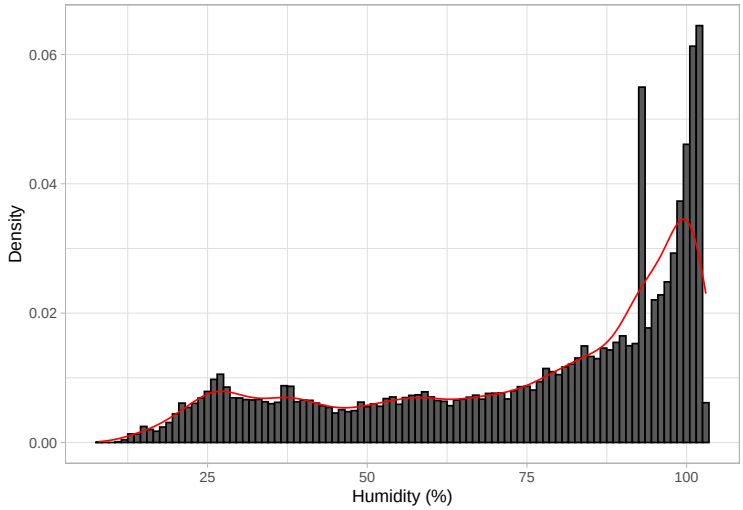

Figure 5: Distribution of humidity

Wind direction

The values of wind direction varies between 0.09°-359.95°. Figure 6 shows the distribution of wind direction, which is characterized by three peaks.

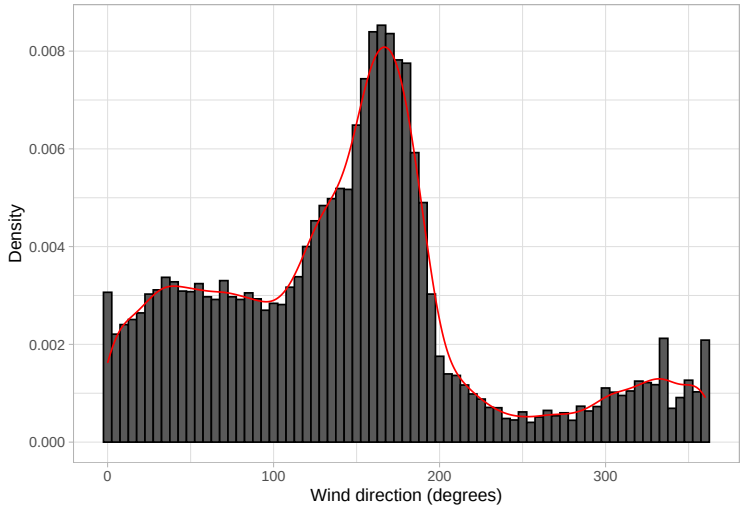

Figure 6: Distribution of wind direction

Wind speed

The distribution of wind speed is shown in Figure 7, the the values of wind speed ranges between 0 and 40.5 miles/hour. The distribution is right skewed and ~ 99% of values are below 15 miles/h.

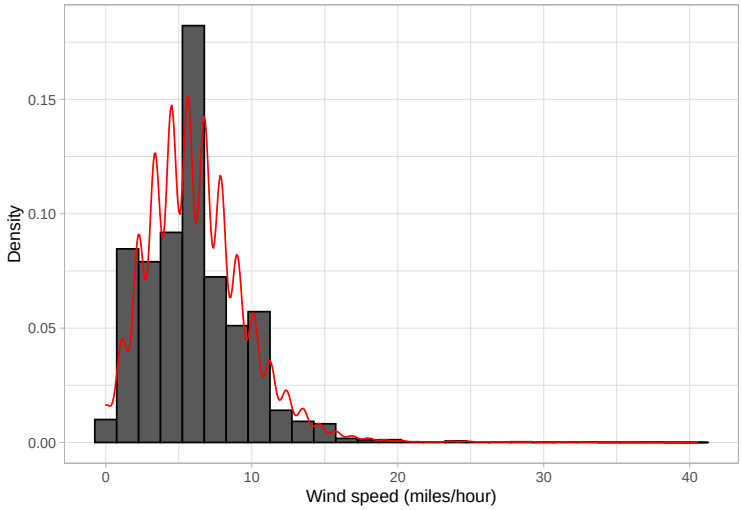

Figure 7: Distribution of wind speed

Correlation matrix

A correlation analysis is performed to investigate if there is a relationship between the various parameters in the data set. A correlation matrix using the Pearson correlation coefficient is generated to identify which parameters have a linear relationship with each other. A value for the correlation coefficient close to 1 indicates a strong linear positive correlation, while a value close to -1 indicates a strong negative linear correlation. Values around zero indicate that there is no linear correlation, but do not exclude the possibility of other kind of correlations (i.e. quadratic, exponential, higher order polynomials, etc..).

The pairwise correlation matrix for the parameters is shown in Figure 8. It is clear that a positive linear correlation exists between the ambient temperature and solar radiation, as confirmed by the correlation coefficient of 0.73. Humidity has a lesser, but potentially significant, impact on radiation with a correlation coefficient of -0.23. Pressure does not seem to have a significant impact on the radiation (coefficient = 0.12), but does correlate with temperature (coefficient = 0.31) and humidity (coefficient = -0.22). This is quite expected, as temperature, pressure and humidity are all characteristics of the atmosphere. Wind speed and wind direction are again characteristics of the local atmosphere, with the wind direction having a moderate correlation (coefficient = -0.23) with radiation. Wind direction also has a moderate correlation with temperature (coefficient = -0.26) and pressure (coefficient = -0.23). No clear linear correlation is observed for the other parameters.

To identify any potential non-linear relationship between solar radiation and other parameters, a scatter plot of solar radiation as a function of the various parameters is shown in Figure 9. A strong linear correlation between solar radiation and ambient temperature can be confirmed from this figure. It can also be noted that the highest values of solar radiation are observed when the ambient pressure values are the highest. The maximum values of solar radiation tends to decrease for high wind speeds. As expected, solar radiation seems to have a strong correlation with the relative time of day, which implicitly accounts for the solar incidence angle.

Algorithm development methodology and results

Splitting the dataset into train and validation sets

The data set analysed here is a time series data and as the values of solar radiation have a strong correlation with the time of the day, splitting this data using the simple train/test technique (assigning 80% of the data

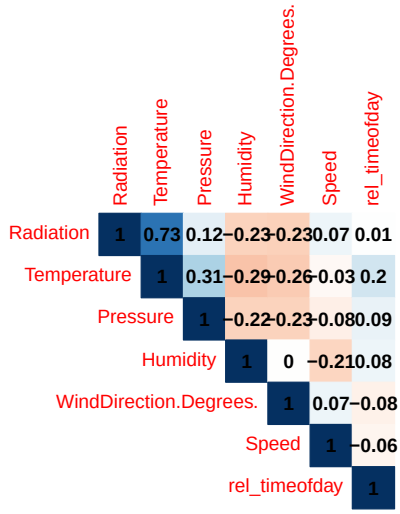

Figure 8: Pairwise correlation matrix

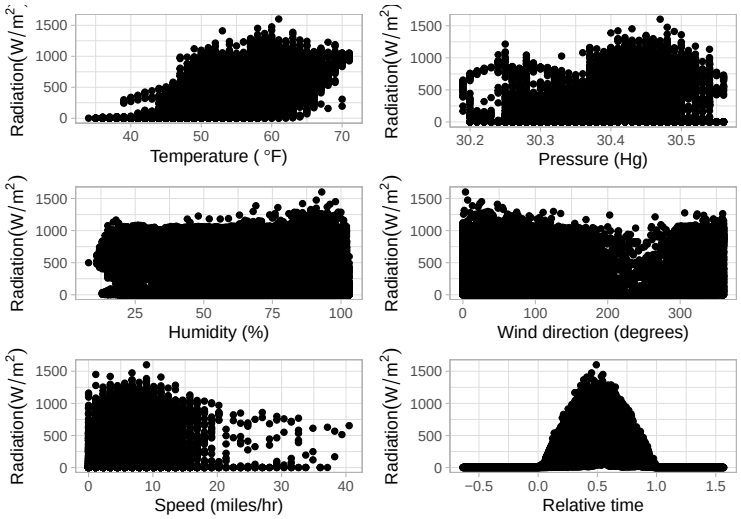

Figure 9: Scatter plot of solar radiation with other parameters

to train and the remaining 20% to test) cannot be employed here as this technique splits the data randomly assuming the observations to be independent. Therefore the data set is split by using a date (20 December 2016), where the values at the end of the data set are used for the final testing/validation (12 days) and everything else for training (110 days).

The training data set is used to both train and test the different machine learning algorithms following the k-fold cross validation method to choose the algorithm with the best performance. The k-fold cross-validation is a very effective method to estimate the prediction error and the accuracy of an algorithm. The general procedure is as follows: randomly split the training data set into k-subsets (k-folds); reserve one subset and train the algorithm on all the other subsets; test the algorithm on the reserved subset; repeat this process until each of the k-subsets have served as the test set; generate the overall prediction error known as the cross-validation accuracy by taking the average of prediction errors in every case. This error serves as the performance metric for the algorithm. It is very important to use the correct value for k, as a low value leads to a biased model and a high value can lead to variability in the performance metrics of the algorithm. A k-fold cross-validation is usually performed using k=5 or k=10, as these values have been shown empirically to yield error rate estimates that suffer neither from excessively high bias nor from very high variance. Therefore in this analysis, k=5 is chosen.

Algorithm accuracy estimation

The goodness of the different machine learning algorithms is evaluated using the R-squared (R^2) and the Root Mean Squared Error (RMSE). R^2 represents the squared correlation between the observed values and the values predicted by the algorithm. The higher the R^2, the better the algorithm. RMSE measures the average prediction error made by the algorithm in predicting the outcome for an observation. That is, the average difference between the observed values and the values predicted by the algorithm. The lower the RMSE, the better the model. R^2 is a measure of how well the model estimates the variability in the observations, whereas the RMSE is a measure of how accurately the model estimates the magnitude of the observations themselves. A higher value for R^2 indicates that the model represents the trend of the observations well while not giving any indication of possible underlying systematic errors, whereas a low value of the RMSE indicates that the model contains no systematic errors. Low RMSE values will indicate more accurate model estimations than high values of R^2.

Testing different machine learning algorithms on the training data set

The objective of this project is to choose the best optimal machine learning algorithm to predict solar radiation as close as possible to the actual values in the test data set, i.e. achieve an R^2 as high as possible and an RMSE as small as possible. The 'caret' (Classification And REgression Training) package in R that contains functions to streamline the model training process is used in this analysis to compare the performance of the following machine learning algorithms and to evaluate the best algorithm to predict solar radiation:

- Linear Regression
- Support Vector Regression
- Random Forest Regression
- Stochastic Gradient Boosting
- Extreme Gradient Boosting
- Ensemble model

The 'caret' package also makes available the variable importance estimates with the use of 'varImp()' for any algorithm. In this analysis, the variable importance is estimated for the Random Forest, Stochastic Gradient Boosting and Extreme Gradient Boosting algorithms. There are two main uses of estimating the variable importance from various algorithms, 1) variables that are important for the majority of algorithms represents genuinely important variables and 2) for building the ensemble model, predictions from algorithms that have significantly different variable importance should be used as their predictions are also expected to be different.

Linear Regression

The preliminary data analysis revealed that the solar radiation is related to temperature in a linear way and to other parameters, at least partially, in a non-linear way. Therefore, the linear regression algorithm, probably the simplest fit, is first tested to check its performance and to use as a benchmark for other algorithms.

Method	R2	RMSE
Linear regression	0.58333	205.9848

The results indicate a relatively poor fitting (R^2 = 0.58333 and RMSE = 205.98477), thus suggesting that more complex class of algorithms, like the Support Vector Regression and tree based, are required to accurately predict the solar radiation. Given that these set of algorithms are more complex than the simple linear regression, some parameters (known as hyper-parameters) have to be user-specified prior to the training of the algorithms. A proper selection of the parameters, known as "hyper-parameter tuning" is very important to achieve the best performance of the algorithm. In this analysis, hyper-parameter tuning is achieved by means of a grid-search. In the grid-search method, a list of potential values for the different hyper-parameters are defined and an exhaustive search over this manually specified subset of the hyper-parameter space is performed to find the best performing set of hyper-parameters.

Support Vector Regression

Support Vector Machine (SVM) is a kernel based machine learning algorithm used for regression and classification problems. SVM when used in regression problems is known as Support Vector Regression (SVR). In SVR, the parameters are normalized to make their scale comparable, i.e. zero mean and unit variance. This is achieved by setting the option 'preProcess = c("center","scale")'.

Method	R2	RMSE
Linear regression	0.58333	205.98477
Support Vector Regression	0.89096	105.45737

The algorithm in this analysis is build using the non-linear Radial Basis Function kernel. In Radial Basis kernel, the optimal values for the hyper-parameters of the SVR algorithm, namely cost and sigma, are estimated using a range of values for cost "C" (1, 1.5, 2, 2.5, 3) and "sigma" (0.5, 0.75, 1, 1.25, 1.5). The parameter cost specifies the tolerance of misclassification; when the cost argument is large, then the margins will be narrow and there will be few support vectors on the margin or violating the margin. The parameter sigma defines how far a single training example can influence the calculation of the separation line. A low value of sigma means that far away points can be also considered. The best hyper-parameter combination obtained via the grid search process uses values of 2 and 1 respectively for cost and sigma. The R^2 and RMSE achieved with the SVR algorithm are 0.89096 and 105.45737 respectively with a substantial improvement of 48.80% as compared to the linear regression algorithm.

In the subsequent sections, 3 ensemble methods based on decision trees are considered to predict solar radiation. Decision tree based algorithms are considered non-parametric and have two main advantages 1)

they are capable of capturing non-linear relationships between the dependent and independent variables; 2) they do not require feature scaling (i.e. regularization) and hence less data cleaning. All of the three ensemble methods take a decision tree and then apply either bootstrap aggregating (bagging) or boosting as a way to reduce variance and bias. Tree-based models also allow for extracting the variable importance which helps to understand whether the models allocate the same importance to the various parameters.

Random Forest Regression

Random forest is an ensemble learning method for classification and regression that operates by building multiple decision trees, known as forest, during training and establishing the outcome based on the predictions of the decision trees. For regression problems, it predicts by taking the average of the output from various trees. The random forest approach is similar to the ensemble technique called as bagging. In this approach, multiple trees are generated by bootstrap samples from training data and then the correlation between the trees is reduced. Performing this approach increases the performance of decision trees and helps in avoiding over fitting.

Method	R2	RMSE
Linear regression	0.58333	205.98477
Support Vector Regression	0.89096	105.45737
Random Forest regression	0.91549	92.83523

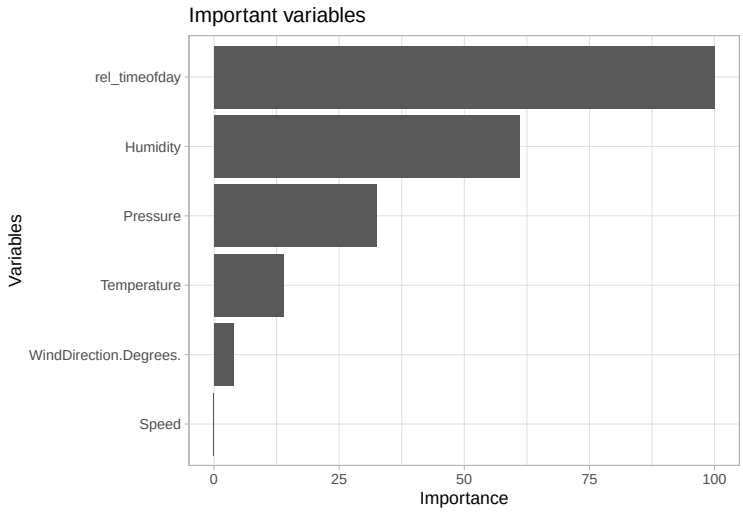

Figure 10: Variable importance

In this analysis, only the two hyper-parameters, namely mtry and ntree, have been tuned as these have the biggest effect on the final accuracy. Here ntree is the number of trees to grow and mtry is the number of variables randomly sampled as candidates at each split. The best hyper-parameter combination obtained via the grid search process is 500 for ntree and 5 for mtry. The default values in the package are used for the other hyper-parameters such as nodesize (minimum number of samples within the terminal nodes) and maxnodes (maximum number of terminal nodes). The random forest regression resulted in $R^2 = 0.91549$ and RMSE = 92.83523 with an improvement of 11.97% as compared to the SVR and 54.93% as compared to linear regression.

To assess the variable importance, the option "importance = TRUE" is set. The variable importance for RF algorithm is measured by recording the decrease in the mean squared error (MSE) each time a variable is used as a node split in a tree. The remaining error left in predictive accuracy after a node split is known as node

impurity and a variable that reduces this impurity is considered more important than those variables that do not. Consequently, the reduction in MSE for each variable across all the trees is accumulated and the variable with the greatest accumulated impact is considered the most important. It can be observed from Figure 10 that for the RF algorithm, the most important variables are the relative time of the day and humidity. This is not quite as expected from the correlation plot shown in Figure 8, where humidity is observed to have only a lesser impact on radiation (correlation coefficient = -0.23). The parameter temperature which has the highest positive linear correlation with solar radiation is ranked only fourth in the variable importance.

Stochastic Gradient Boosting

Stochastic Gradient boosting (SGB) refers to a class of ensemble machine learning algorithms used for classification and regression that builds up on the concept of the RF algorithm. It is a generalized method that boosts the accuracy of the algorithm by considering a series of models and then producing a weighted sum of the said models. In other words, the boosting works in a similar way to the RF algorithm, except that the trees are grown sequentially, i.e. each tree is grown using information from previously grown trees. One of the main differences between boosting and random forests is that in boosting, because the growth of a particular tree takes into account the other trees that have already been grown, smaller trees are typically sufficient, i.e. less splits and depth.

Method	R2	RMSE
Linear regression	0.58333	205.98477
Support Vector Regression	0.89096	105.45737
Random Forest regression	0.91549	92.83523
SGB regression	0.9024	99.75857

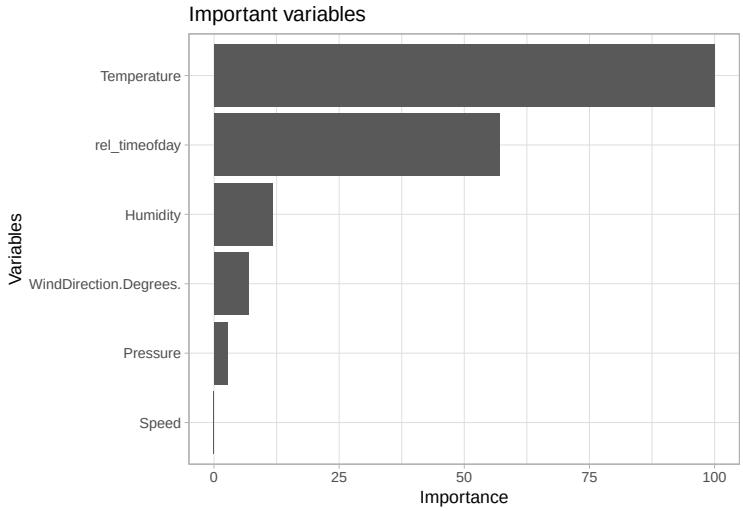

Figure 11: Variable importance

The main tuning parameters for SGB are 1) number of iterations (n.trees), 2) complexity of the tree (interaction.depth), 3) learning rate (shrinkage) to determine the contribution of each tree on the final outcome and control how quickly the algorithm proceeds down the gradient descent and 4) the minimum number of training set samples in a node to commence splitting (n.minobsinnode). The best hyper-parameter combination obtained via the grid search process is 12 for interaction.depth, 1500 for n.trees, 0.1 for shrinkage and 10 for n.minobsinnode. The R^2 and RMSE achieved with SGB are respectively 0.9024 and 99.75857 with a substantial improvement of 51.57% as compared to the linear regression algorithm.

Figure 11 shows the variable importance plot for the SGB algorithm which indicates that the most important features are temperature and relative time of the day. This is expected since a high correlation of solar radiation with temperature and relative time of the day was observed in the correlation and scatter plots shown respectively in Figure 8 and Figure 9. However, the variable importance observed for SGB is different from what is observed for the RF regression (Figure 10). This can be explained as due to the fact that for SGB algorithm, whenever a node is split based on a feature, the reduction in MSE attributed to the split (squared error in one "mixed" node before - sum of squared error in two "purer" nodes afterwards) is counted towards the absolute variable importance. Also, the importance of the variables are summed over each boosting iteration.

Extreme Gradient Boosting

The eXtreme Gradient Boosting (XGBoost) is an ensemble machine learning algorithm and is a more regularized form of Gradient Boosting. XGBoost uses advanced regularization to control over-fitting, which improves the model generalization capabilities. XGBoost delivers high performance as compared to Gradient Boosting and its training is very fast and can be parallelized across clusters. In this analysis, the booster type used is "xgbTree". The seven hyper-tuning parameters for XGBoost are 1) nrounds: it controls the maximum number of iterations, 2) eta: it controls the learning rate, i.e., the rate at which the algorithm learns patterns in data. After every round, it shrinks the feature weights to reach the best optimum. Lower eta leads to slower computation and therefore must be supported by increase in nrounds, 3) gamma: it controls regularization (or prevents overfitting). Higher the value of gamma, higher the regularization, 4) max_depth: it controls the depth of the tree. Larger the depth, more complex the algorithm and therefore higher chances of overfitting, 5) min_child_weight: it refers to the minimum number of instances required in a child node, i.e. it blocks the potential feature interactions to prevent overfitting, 6) subsample: it controls the number of samples (observations) supplied to a tree, 7) colsample_bytree: it controls the number of variables supplied to a tree. These tuning parameters were changed in steps so that the search grid was not too big.

Method	R2	RMSE
Linear regression	0.58333	205.98477
Support Vector Regression	0.89096	105.45737
Random Forest regression	0.91549	92.83523
SGB regression	0.9024	99.75857
XGBoost regression	0.91177	94.811

The best hyper-parameter combination obtained is 1000 for nrounds, 11 for max_depth, 0.01 for eta, 0 for gamma, 1 for colsample_bytree, subsample and min_child_weight. The XGBoost resulted in $R^2 = 0.91177$ and RMSE = 94.811 with an improvement of 53.97% as compared to the linear regression algorithm. The variable importance plot for the XGB algorithm is shown in Figure 12. It is clear that the XGBoost and SGB selected the same order of importance for the available features, and that the most important features are the ambient temperature and the relative time of the day.

The results obtained so far indicates that the RF, SGB and XGBoost algorithms provides similar results when comparing the R^2. However the obtained RMSE values indicate that the RF and XGBoost algorithms provide the best improvement. This possibly indicates that a single algorithm may not able to capture all the variability to make the perfect prediction, thus suggesting the need to develop an Ensemble model.

Ensemble Algorithm

Ensemble model is a machine learning approach to combine multiple other algorithms in the prediction process. Machine learning algorithms have their limitations and therefore producing a single algorithm with high accuracy is challenging. If multiple algorithms can be built and combined, the overall accuracy

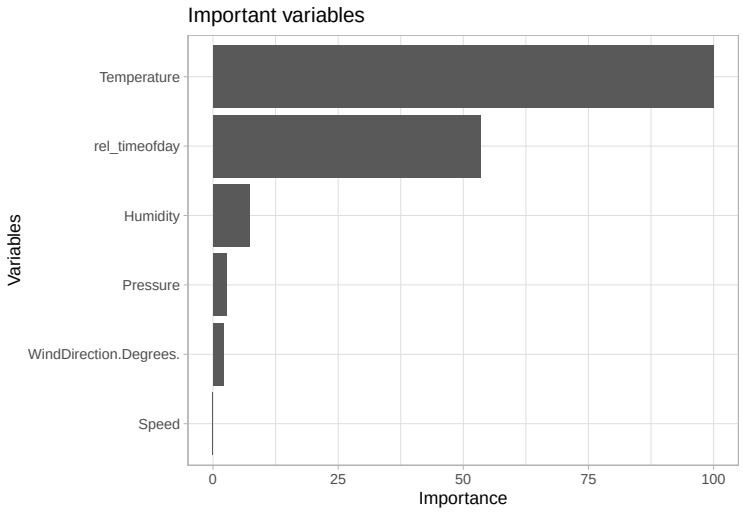

Figure 12: Variable importance

of the final algorithm could get boosted. The motivation for using ensemble algorithms is to reduce the generalization error of the prediction. In this analysis, the 'caretEnsemble' package in R is used to built an ensemble algorithm. The 'caretList' function is used to build lists of caret models on the training data and the 'caretEnsemble' function is used to create a simple linear blend of models. An algorithm is built by ensembling the SVR, RF, SGB and XGBoost regression models. As in the development of the individual algorithms, 5 fold cross-validation for each algorithm and the same set of values for the algorithm hyper-parameters are used.

Table 1: Model correlation

	SGB	RF	SVR	XGBoost
SGB	1.0000000	0.1077570	0.4646625	0.2449001
RF	0.1077570	1.0000000	0.5956528	0.8040191
SVR	0.4646625	0.5956528	1.0000000	0.8329500
XGBoost	0.2449001	0.8040191	0.8329500	1.0000000

The correlation between the different algorithms used in the ensemble algorithm development is shown in Table 1, which indicates some high correlation between SVR and XGBoost, RF and XGBoost, and RF and SVR. Ideally, algorithms that are low correlated with each other, i.e. diverse and independent, are used in developing an ensemble algorithm as this helps in reducing the prediction error when the ensemble approach is used.

Method	R2	RMSE
Linear regression	0.58333	205.98477
Support Vector Regression	0.89096	105.45737
Random Forest regression	0.91549	92.83523
SGB regression	0.9024	99.75857
XGBoost regression	0.91177	94.811
Ensemble algorithm	0.91488	93.14551

The R^2 and RMSE achieved with the Ensemble algorithm are respectively 0.91488 and 93.14551, with an improvement of 54.78% compared to the linear regression algorithm. The results are quite comparable to

what is achieved with RF regression. The 'caretStack' which uses a caret model to combine the outputs from several component caret models was also used to create an ensemble algorithm. However, the performance could not be improved further and therefore is not included in this analysis.

Final test using the validation dataset

It is clear from the analysis presented above that using the RF, XGBoost and Ensemble algorithms achieves the best accuracy with the highest R^2 and lowest RMSE on the training data set. Therefore these algorithms are used to predict the solar radiation in the final validation data set.

Method	R2	RMSE
RF regression	0.94776	94.73694
XGBoost regression	0.94487	97.54311
Ensemble algorithm	0.94924	93.07142

It is clear that the all the three algorithms yield comparable performance in terms of R^2 and RMSE, with a slightly improved performance by the Ensemble model with a $R^2 = 0.94924$ and RMSE = 93.07142. A plot of the actual and predicted solar radiation by the three algorithms over the validation data set is shown in Figure 13, which helps to better visualize the performance of the algorithms. The black lines show the actual values in the validation data set, where as red, blue and green lines show the values predicted respectively by RF, XGBoost and Ensemble algorithms. The graphical inspection also indicates that all the three algorithms leads to similar predictions. It is also clear from this Figure that all the three algorithms sometimes fail to accurately predict high values of solar radiation on some days, such as on 25 and 28 December. This could be improved by refining the hyper-parameter tuning.

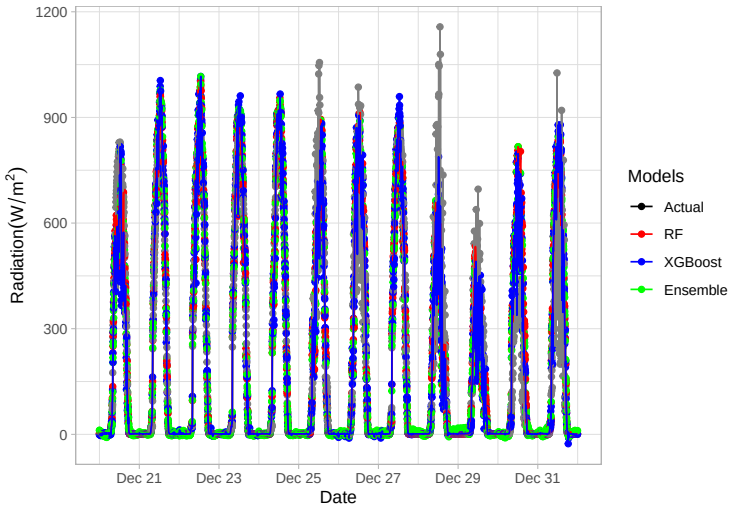

Figure 13: Comparison of Model performance

Conclusions

A data set containing the meteorological data collected from the HI-SEAS weather station during four months (September to December 2016) was analyzed and machine learning algorithms were built with the objective of predicting the solar radiation given a set of measurable meteorological conditions. A preliminary data analysis was carried out to better understand the parameters to be included in the machine learning

algorithm. The data set is initially split into a training and a final validation data set. The training data set is further used to both train and test the different machine learning algorithms following the k-fold cross validation method to choose the algorithm with best performance. The goodness of the different machine learning algorithms was evaluated using the R^2 and RMSE. The best performing algorithms on the training data set were RF (R^2=0.94776, RMSE = 94.73694), XGBoost (R^2=0.94487, RMSE = 97.54311) and an Ensemble algorithm built by ensembling the SVR, RF, SGB and XGBoost algorithms (R^2=0.91488, RMSE = 93.14551). The testing of these three algorithms on the final validation data set achieved comparable performance in terms of R^2 and RMSE.

A further improvement in the predictions could be obtained by applying a more extensive hyper-parameter tuning procedure. In this analysis, the approach followed for hyper-parameter tuning was by applying a grid search technique, i.e. iterating through a list of arbitrary values and choosing the one that yielded the best results. However by including perhaps an analytical method to determine optimal values would be more beneficial. Furthermore, focus could also be placed on building an algorithm that only accounts for the hours of the day when the solar radiation is present, i.e. an algorithm that does not have to account for the night hours. The algorithm could be further improved if the relationships between the different parameters, such as temperature, pressure and humidity, which are not completely independent, and influences on one another can be factored into the algorithm.