Forecasting Solar Generation

Rose Dennis

May 8, 2019

1 Introduction

1.1 Motivation

The way our society functions today has been progressively increasing the need for electricity. The continuous rise of this demand has become a highly intricate problem for data analysts. In light of this, scientists, mathematicians, and statisticians have designed, constructed, and improved a smart grid- a multidimensional network that monitors, measures, and manages the transport of electriciy. The original smart grid only included the exchange of coal-generated electricity between individual homes, corporate buildings, utilities, and power plants. However, solely relying on these primary sources of energy has become unsustainable so our society is gradually moving towards more innovative, renewable energy methods. Wind power, biomass, hydroelectric, and solar power are just a few of these kinds of renewable energy approaches that are naturally replenished on a human timescale. This project focuses on the specific method of solar energy.

1.2 Solar Energy

With the smart grid described above in mind, being able to forecast solar generation can provide for better management and optimization of the grid systems and a better understanding of necessary power production procedures. The total amount of solar energy comes from two groups; solar farms and individual homes. The former is usually government regulated and therefore has substantial data and analysis regarding its generation and role in the current smart grid. The latter group refers to the solar generation from individual homes. This is called ”behind-the-meter” solar production and is less accessible and extremely subjective to individual homes. Therefore, we have less data and fewer analyses pertaining to this group, even though it has become an integral part in how it's incorporated into the grid.

1.3 Goal

The future goal for this project's results is to assist in modeling the interplay between a home that can produce solar energy and its respective electricity consumption and to further integrate it into the smart grid. Our immediate goal of this project is to use weather related factors to create a model for predicting solar generation for a single home utilizing different statistical methods.

2 Data

2.1 Data Sources and Description

The data were obtained from a collection of datasets called the Smart* Data Set for Sustainability which is part of the Umass Trace Repository. An arbitrary single home's solar generation dataset was chosen which included the usage and weather data to complement the solar data. The comma separated value dataset file was imported into R and underwent cleaning and formatting. The single home's data was from year 2014, with a total of 3,285 observations. Each row represents the figures for a single hour of the day comprising of 32 features.

2.2 Cleaning and Formatting

The project uses the complete case dataset, eliminating all of the observations with incomplete rows. Secondly, we dropped the variables associated with power consumption, including use because these were not related to our model. Since we are interested in solar generation, reliant on the sun, we further filter only the peak hours in the day-hours 13-17. We were also interested in how the seasons differ so we split the dataset into winter, fall, summer, and spring. Preliminary descriptives were explored on each season but further, in depth, analysis was performed only on the two extreme seasons: winter and summer. The winter season has a total of 586 observations with 14 predictors and the summer season contains 743 observations with 14 predictors.

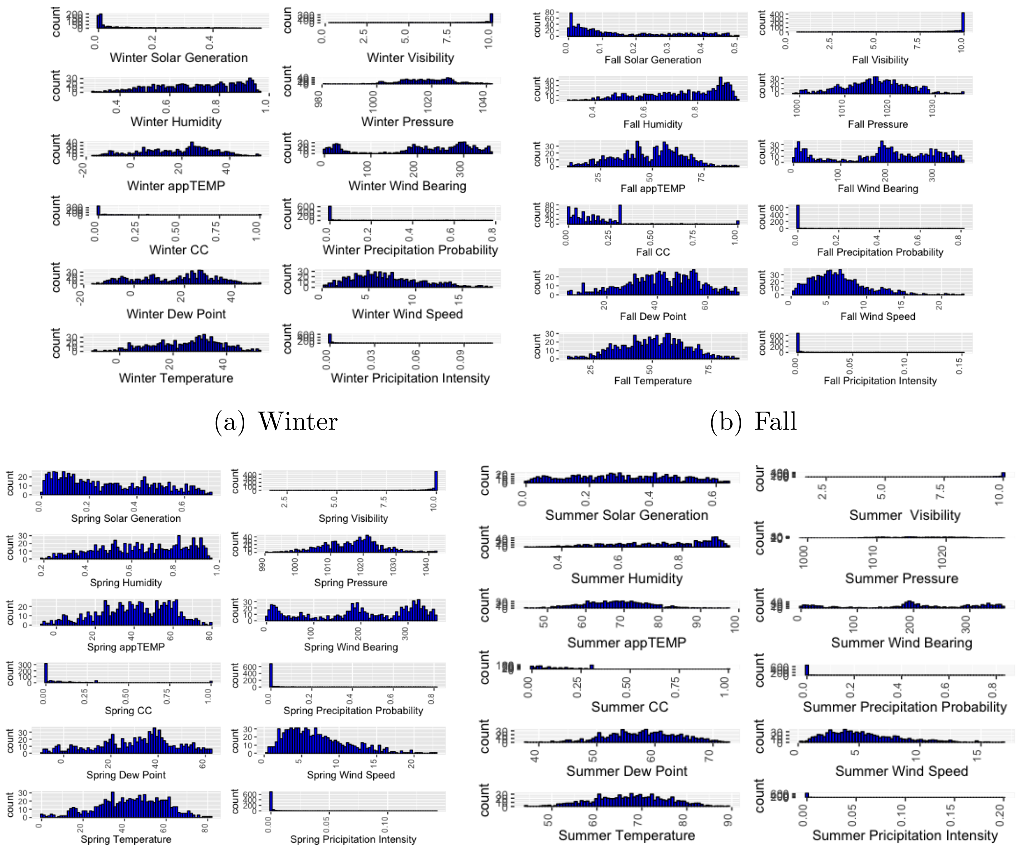
3 Exploratory Data Analysis

3.1 Predictors and Dataset

	Variable Description	
Predictor	Variable Type	Description
Gen	numeric	solar generation
Temp	numeric	avg temperature
Icon	character	overall description
Hum	numeric	humidity measure
Vis	numeric	visibility
Sum	character	atm desc.
AppTemp	numeric	temp. in shade
Press	numeric	pressure
WindSpeed	numeric	wind speed
Cc	numeric	cloud coverage
WindBearing	integer	wind direction
PricipIntensity	numeric	intensity of rain
DewPoint	numeric	dewpoint temp.
PrecipProb	numeric	precipitation prob

Table 1: Data types and Variable Description

To get a sense of the data, we look at each variable type and what it describes. Table 1 displays each variable, what class its coded in R, and a short description of the variable. Solar generation is measured in British thermal units per square foot of collector space.



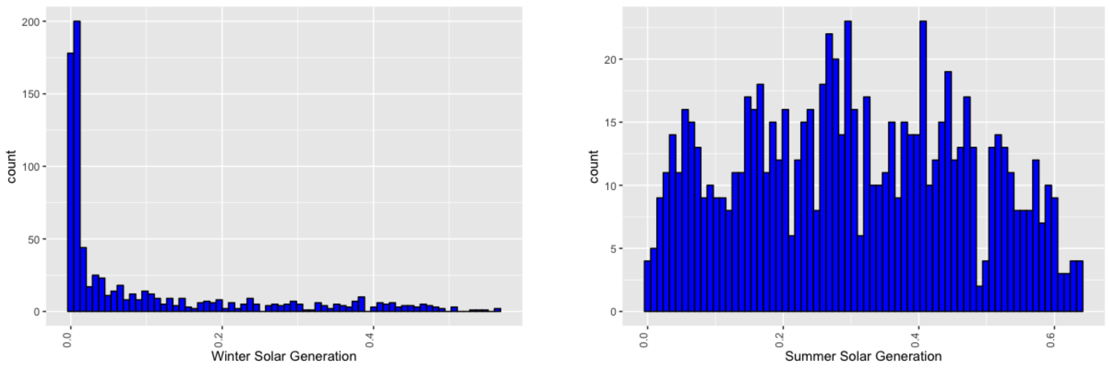
(a) Winter

(b) Fall

(c) Spring

(d) Summer

Figure 1: A set of variable descriptives for each season: (a) distribution plots for each predictor in the Winter; (b) distribution plots for each predictor in the Fall; (c) distribution plots for each predictor in the Spring; and, (d) distribution plots for each predictor in the Summer.

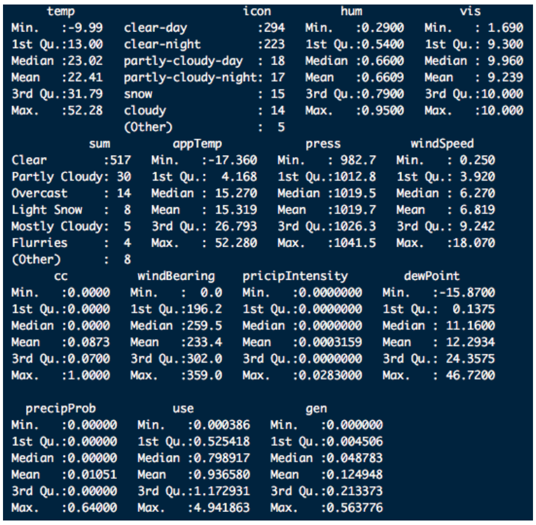


(a) Winter

(b) Summer

Figure 2: Solar Generation

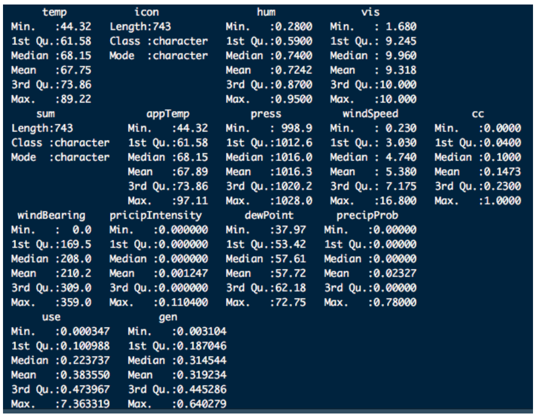
To begin the analysis, we look at the distributions for each continuous variable for each season. Figure 1 allows us to compare these descriptives. As mentioned before,



```
      temp                       icon          hum              vis
Min.   :-9.99   clear-day          :294   Min.   :0.2900   Min.   : 1.690
1st Qu.:13.00   clear-night        :223   1st Qu.:0.5400   1st Qu.: 9.300
Median :23.02   partly-cloudy-day  : 18   Median :0.6600   Median : 9.960
Mean   :22.41   partly-cloudy-night: 17   Mean   :0.6609   Mean   : 9.239
3rd Qu.:31.79   snow               : 15   3rd Qu.:0.7900   3rd Qu.:10.000
Max.   :52.28   cloudy             : 14   Max.   :0.9500   Max.   :10.000
                (Other)            :  5
           sum          appTemp            press          windSpeed
Clear        :517   Min.   :-17.360   Min.   : 982.7   Min.   : 0.250
Partly Cloudy: 30   1st Qu.:  4.168   1st Qu.:1012.8   1st Qu.: 3.920
Overcast     : 14   Median : 15.270   Median :1019.5   Median : 6.270
Light Snow   :  8   Mean   : 15.319   Mean   :1019.7   Mean   : 6.819
Mostly Cloudy:  5   3rd Qu.: 26.793   3rd Qu.:1026.3   3rd Qu.: 9.242
Flurries     :  4   Max.   : 52.280   Max.   :1041.5   Max.   :18.070
(Other)      :  8
       cc          windBearing    pricipIntensity        dewPoint
Min.   :0.0000   Min.   :  0.0   Min.   :0.0000000   Min.   :-15.8700
1st Qu.:0.0000   1st Qu.:196.2   1st Qu.:0.0000000   1st Qu.:  0.1375
Median :0.0000   Median :259.5   Median :0.0000000   Median : 11.1600
Mean   :0.0873   Mean   :233.4   Mean   :0.0003159   Mean   : 12.2934
3rd Qu.:0.0700   3rd Qu.:302.0   3rd Qu.:0.0000000   3rd Qu.: 24.3575
Max.   :1.0000   Max.   :359.0   Max.   :0.0283000   Max.   : 46.7200

  precipProb          use                  gen
Min.   :0.00000   Min.   :0.000386   Min.   :0.000000
1st Qu.:0.00000   1st Qu.:0.525418   1st Qu.:0.004506
Median :0.00000   Median :0.798917   Median :0.048783
Mean   :0.01051   Mean   :0.936580   Mean   :0.124948
3rd Qu.:0.00000   3rd Qu.:1.172931   3rd Qu.:0.213373
Max.   :0.64000   Max.   :4.941863   Max.   :0.563776
```

(a) Winter



```
      temp           icon                 hum              vis
Min.   :44.32   Length:743         Min.   :0.2800   Min.   : 1.680
1st Qu.:61.58   Class :character   1st Qu.:0.5900   1st Qu.: 9.245
Median :68.15   Mode  :character   Median :0.7400   Median : 9.960
Mean   :67.75                      Mean   :0.7242   Mean   : 9.318
3rd Qu.:73.86                      3rd Qu.:0.8700   3rd Qu.:10.000
Max.   :89.22                      Max.   :0.9500   Max.   :10.000
    sum                appTemp          press          windSpeed             cc
Length:743         Min.   :44.32   Min.   : 998.9   Min.   : 0.230   Min.   :0.0000
Class :character   1st Qu.:61.58   1st Qu.:1012.6   1st Qu.: 3.030   1st Qu.:0.0400
Mode  :character   Median :68.15   Median :1016.0   Median : 4.740   Median :0.1000
                   Mean   :67.89   Mean   :1016.3   Mean   : 5.380   Mean   :0.1473
                   3rd Qu.:73.86   3rd Qu.:1020.2   3rd Qu.: 7.175   3rd Qu.:0.2300
                   Max.   :97.11   Max.   :1028.0   Max.   :16.800   Max.   :1.0000
 windBearing    pricipIntensity       dewPoint       precipProb
Min.   :  0.0   Min.   :0.000000   Min.   :37.97   Min.   :0.00000
1st Qu.:169.5   1st Qu.:0.000000   1st Qu.:53.42   1st Qu.:0.00000
Median :208.0   Median :0.000000   Median :57.61   Median :0.00000
Mean   :210.2   Mean   :0.001247   Mean   :57.72   Mean   :0.02327
3rd Qu.:309.0   3rd Qu.:0.000000   3rd Qu.:62.18   3rd Qu.:0.00000
Max.   :359.0   Max.   :0.110400   Max.   :72.75   Max.   :0.78000
     use                gen
Min.   :0.000347   Min.   :0.003104
1st Qu.:0.100988   1st Qu.:0.187046
Median :0.223737   Median :0.314544
Mean   :0.383550   Mean   :0.319234
3rd Qu.:0.473967   3rd Qu.:0.445286
Max.   :7.363319   Max.   :0.640279
```

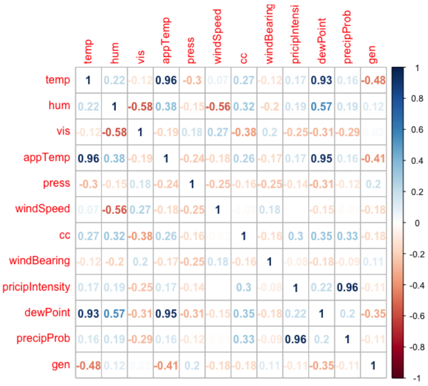
(b) Summer

Figure 3: Summary Statistics

the project will only look at Winter and Summer for modeling because they are the two extremes pertaining to solar generation. In this regard, Figure 2 extracts the distributions from Figure 1 and presents the solar generation in Winter and summer. It is clear that the winter season is heavily right skewed whereas the generation in the summer looks more normally distributed.

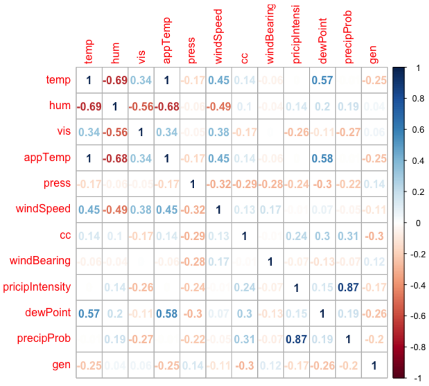
Furthermore, Figure 3 shows a basic summary of all 14 predictors used in both seasons. Note that the mean solar generation of the summer is more than two times that of the winter. Yet, the variance of each season is relatively similar, with the summer generation having a variance of 0.026 and the winter generation having a variance of 0.023.

3.2 Variable Correlation



	temp	hum	vis	appTemp	press	windSpeed	cc	windBearing	pricipIntensi	dewPoint	precipProb	gen
temp	1	0.22	-0.12	0.96	-0.3	0.07	0.27	-0.12	0.17	0.93	0.16	-0.48
hum	0.22	1	-0.58	0.38	-0.15	-0.56	0.32	-0.2	0.19	0.57	0.19	0.12
vis	-0.12	-0.58	1	-0.19	0.18	0.27	-0.38	0.2	-0.25	-0.31	-0.29	0.02
appTemp	0.96	0.38	-0.19	1	-0.24	-0.18	0.26	-0.17	0.17	0.95	0.16	-0.41
press	-0.3	-0.15	0.18	-0.24	1	-0.25	-0.16	-0.25	-0.14	-0.31	-0.12	0.2
windSpeed	0.07	-0.56	0.27	-0.18	-0.25	1	-0.02	0.18	0	-0.15	-0.01	-0.18
cc	0.27	0.32	-0.38	0.26	-0.16	-0.02	1	-0.16	0.3	0.35	0.33	-0.18
windBearing	-0.12	-0.2	0.2	-0.17	-0.25	0.18	-0.16	1	-0.08	-0.18	-0.09	0.11
pricipIntensity	0.17	0.19	-0.25	0.17	-0.14	0	0.3	-0.08	1	0.22	0.96	-0.11
dewPoint	0.93	0.57	-0.31	0.95	-0.31	-0.15	0.35	-0.18	0.22	1	0.2	-0.35
precipProb	0.16	0.19	-0.29	0.16	-0.12	-0.01	0.33	-0.09	0.96	0.2	1	-0.11
gen	-0.48	0.12	0.02	-0.41	0.2	-0.18	-0.18	0.11	-0.11	-0.35	-0.11	1

(a) Winter



	temp	hum	vis	appTemp	press	windSpeed	cc	windBearing	pricipIntensi	dewPoint	precipProb	gen
temp	1	-0.69	0.34	1	-0.17	0.45	0.14	-0.06	0	0.57	-0.01	-0.25
hum	-0.69	1	-0.56	-0.68	-0.06	-0.49	0.1	-0.04	0.14	0.2	0.19	0.04
vis	0.34	-0.56	1	0.34	-0.05	0.38	-0.17	0	-0.26	-0.11	-0.27	0.06
appTemp	1	-0.68	0.34	1	-0.17	0.45	0.14	-0.06	0	0.58	-0.01	-0.25
press	-0.17	-0.06	-0.05	-0.17	1	-0.32	-0.29	-0.28	-0.24	-0.3	-0.22	0.14
windSpeed	0.45	-0.49	0.38	0.45	-0.32	1	0.13	0.17	-0.01	0.07	-0.05	-0.11
cc	0.14	0.1	-0.17	0.14	-0.29	0.13	1	-0.01	0.24	0.3	0.31	-0.3
windBearing	-0.06	-0.04	0	-0.06	-0.28	0.17	-0.01	1	-0.07	-0.13	-0.07	0.12
pricipIntensity	0	0.14	-0.26	0	-0.24	-0.01	0.24	-0.07	1	0.15	0.87	-0.17
dewPoint	0.57	0.2	-0.11	0.58	-0.3	0.07	0.3	-0.13	0.15	1	0.19	-0.26
precipProb	-0.01	0.19	-0.27	-0.01	-0.22	-0.05	0.31	-0.07	0.87	0.19	1	-0.2
gen	-0.25	0.04	0.06	-0.25	0.14	-0.11	-0.3	0.12	-0.17	-0.26	-0.2	1

(b) Summer

Figure 4: Correlation Heat Maps

The last thing to do before starting to model is to check the collinearity between the predictors. We use the *corrplot* package in R to produce Figure 4. We see that in the winter season precipProb and pricipIntensity are strongly correlated with a pearson coefficient of 0.96. The summer season also has a high coefficient between the aformentioned variables of 0.87. We note that the summer season shows a perfect correlation between appTemp and temp. We will address this in later analysis.

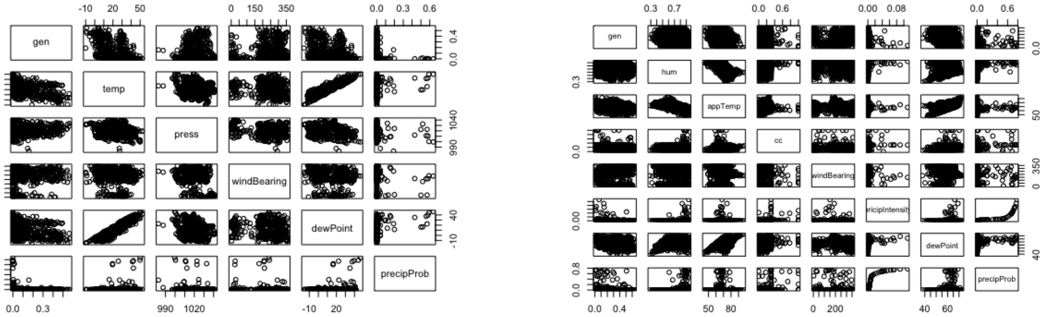
4 Regression Analysis

4.1 Multiple Linear Regression

We begin with Multiple Linear Regression. These models are inflexible but pretty easy to interpret. The full linear regression model was fit to the data for Winter and Summer using generation as the response. Its residual standard error was 0.126 on 564 degrees of freedom, Multiple R-squared statistic was 0.3414, Adjusted R-squared was 0.3169, F-statistic was 13.92 on 21 and 564 DF, and the p-value was < 2.2e-16. We see that this is a relatively low adjusted R-squared value.

Multiple Linear Regression becomes even less flexible because of all the modeling assumptions that it has to meet. We look at these next.

4.1.1 Model Assumptions

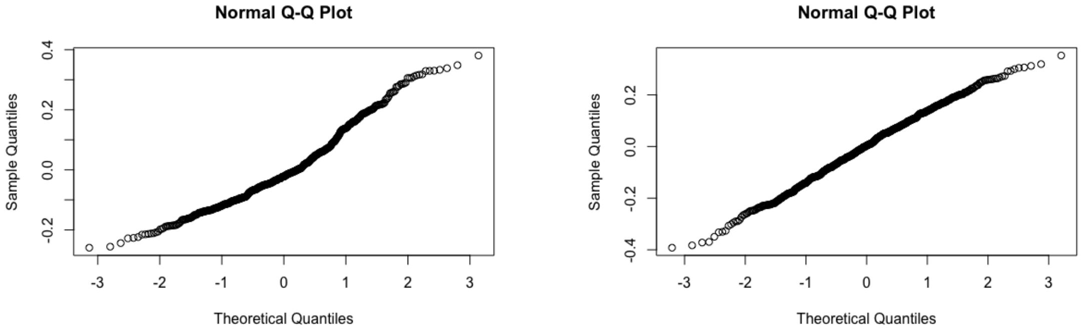


(a) Winter

(b) Summer

Figure 5: Pairs Plots

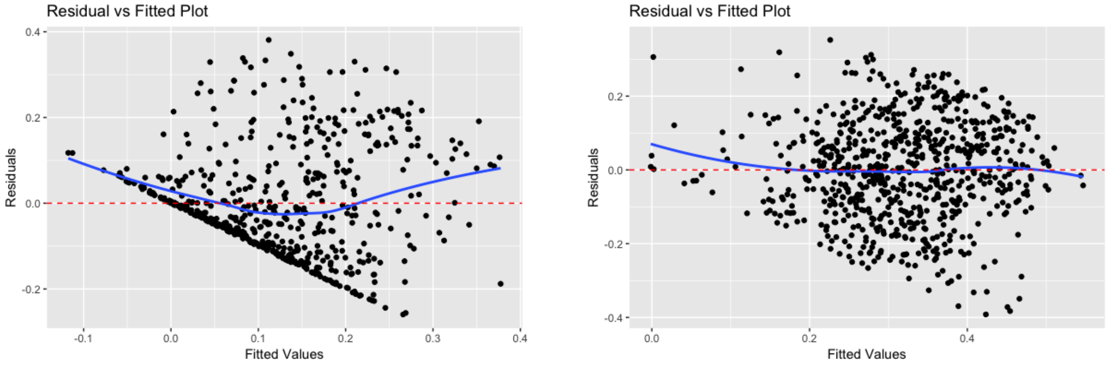
We first require each relationship between a predictor and the response to be linear. Figure 5 shows a scatterplot matrix showing all of the relationships between each other and the response variable, solar generation. We also need to meet the assumption of having normally distributed residuals and constant variance. Here, a QQ plot, as in Figure 6, is useful in that it displays our errors versus what they should look like if they're normally distributed. Another helpful graphic, like Figure 7, plots the residuals against the fitted values. We see that the summer season fulfills this normally distributed errors assumption better than the winter season. Lastly, a linear model assumes that there is no multicollinearity in the data. We've examined this through Figures 5 and 5.



(a) Winter

(b) Summer

Figure 6: QQ Plots



(a) Winter

(b) Summer

Figure 7: Errors Plots

4.1.2 Variable Selection

	Variable Selection	
Method	Adjusted R-squared Value	Selected Variables
Anova	0.3203	temp,icon,hum,vis,press, and windBearing
Backwards	0.3238	temp, iconclear-night, iconcloudy, press, windBearing, dewPoint, precipProb
Forwards	0.3169	iconclear-night, press, windBearing
Forward Stepwise	0.3238	temp, iconclear-night, iconcloudy, press, windBearing, dewPoint, precipProb

Table 2: Winter Variable selection

	Variable Selection	
Method	Adjusted R-squared Value	Selected Variables
Anova	0.2866	temp, icon, sum, appTemp, cc, windBearing, dewPoint, precipProb
Backwards	0.2959	conclear-night,iconpartly-cloudy-night,hum,sumLight Rain, sumMostly Cloudy, appTemp, cc, windBearing, pricipIntensity, dewPoint, precipProb
Forwards	0.2953	iconclear-night,iconpartly-cloudy-night,hum,sumLight Rain, sumMostly Cloudy, cc, windBearing, pricipIntensity dewPoint, precipProb
Forward Stepwise	0.2959	iconclear-night,iconpartly-cloudy-night,hum,sumLight Rain, sumMostly Cloudy, appTemp, cc, windBearing, pricipIntensity dewPoint, precipProb

Table 3: Summer Variable selection

	Variance Inflation Factor		
Variable	GVIF	DF	GVIF$^{(}$1/(2 $*$ Df))
temp	8.782816	1	2.963582
icon	5.982516	6	1.160754
press	1.300800	1	1.140526
windBearing	1.215707	1	1.102591
dewPoint	8.966507	12.994413	
precipProb	4.061834	1	2.015399

Table 4: Variance Inflation Factors

At first, we look at the full model for both seasons which keeps all the predictors to get a sense of the linear models. The adjusted R-squared values were 0.3169 and 0.2953 for winter and summer, respectively. These are relatively low and we want our predictors to explain more of the proportion of the variance in solar generation. In order to increase the adjusted R-squared value, we ran various selection methods. The four subset methods considered were backwards, forwards, forward stepwise, and an anova. We use an Akaike information criterion of 0.05 within each method. Tables 2 and 3 show the adjusted R-squared values for the model that each method produces. We then choose the one with the highest adjusted R-squared. In both cases, backwards selection and forward stepwise result in the highest adjusted R-squared. Our final winter model keeps temp, icon, press, windBearing, dewPoint and precipProb with a test MSE of 0.0162. The functional form of this model is:

$$Gen = -2.102 - 0.01234 \cdot temp + 0.0364 \cdot iconclear - night - 0.06099 \cdot iconcloudy - 0.01628 \cdot iconpartly - cloudy - day + 0.02581 \cdot iconpartly - cloudy - night + 0.1616 \cdot iconrain + 0.02141 \cdot iconsnow + 0.0023 \cdot press + 0.0002 \cdot windBearing + 0.0074 \cdot dewPoint - 0.3050 \cdot precipProb$$

The final summer model retains the icon, hum, sum, appTemp, cc, windBearing, pricipIntensity, dewPoint, and precipProb variables resulting in a test MSE of 0.0199. The functional form of this model is:

$$Gen = 1.283 - 0.1839 \cdot iconclear - night + 0.0261 \cdot iconcloudy - 0.0016 \cdot iconfog - 0.02898 \cdot iconpartly - cloudy - day - 0.1281 \cdot iconpartly - cloudy - night - 0.0346 \cdot iconrain - 0.5494 \cdot hum + 0.2512 \cdot sumDrizzle + 0.2958 \cdot sumLightRain + 0.2462 \cdot sumMostlyCloudy - 0.0192 \cdot appTemp - 0.2711 \cdot cc + 0.0001 \cdot windBearing + 5.779 \cdot pricipIntensity + 0.0135 \cdot dewPoint - 0.864 \cdot precipProb$$

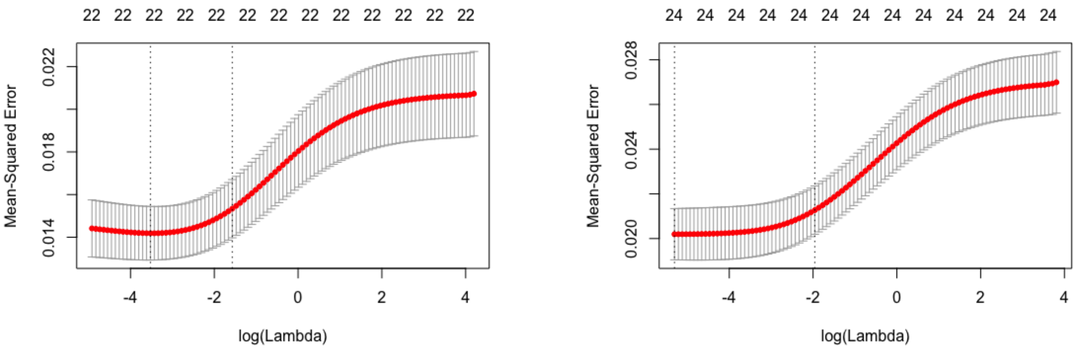
Now that we have the final models for both seasons, we want to check the collinearity between variables. To do this we use two methods: variance inflation factors for the winter season and an alias for the summer season. We chose to run an alias, rather than the VIF, for the summer season because there were some perfectly correlated variables as seen in Figure 4. Table 4 shows that no variable has a dangerously high variance inflation factor and the alias told us that sumFoggy/iconfog, sumOvercast/iconcloudy, sumPartly Cloudy/iconpartly-cloudy-night, sumRain/iconrain, sumPartly Cloudy/sumMostly Cloudy, sumRain/sumDrizzle, sumRain/sumLight Rain are highly correlated. To the layman, these variables being correlated would make sense just from what the variables represent.

4.2 Ridge Regression

The next step in our modeling process was to use ridge regression. We do this because of the bias-variance tradeoff advantage it has over ordinary least squares regression. Ridge regression adds a tuning parameter, λ, to the previous linear regression function that we minimize. We seek coefficients that minimize the RSS but also introduce a shrinkage penalty so that the model shrinks the estimates towards zero. The following equation is minimized for ridge regression:

$$RSS + \lambda \sum_{j=1}^{p} (\beta_j)^2 \tag{1}$$

The equation exhibits that it is clear that the tuning parameter, λ is crucial for fitting our model. As it increases, the shrinkage term dominates making the ridge coefficients go to zero. The way we choose λ is through cross validation. Figure



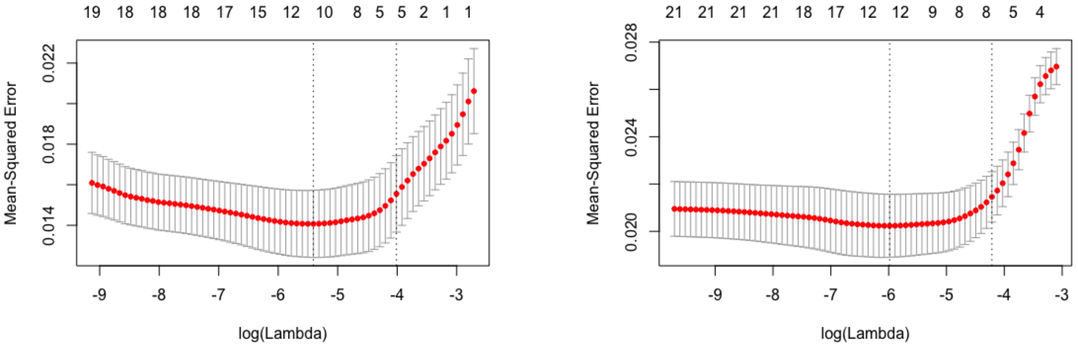
(a) Winter

(b) Summer

Figure 8: Cross Validation Ridge Regression Plots

8 shows the cross validation plot for winter and summer, respectively. R helps us acquire the best lambda to fit, telling us that using $\lambda = 0.0295$ minimizes the cv error and results in a test MSE of 0.0153 for the winter season. Similarly, for the summer season, $\lambda = 0.0050$ minimizes the cv error and results in a test MSE of 0.0186.

4.3 Lasso Regression



(a) Winter

(b) Summer

Figure 9: Cross Validation Lasso Regression Plots

The one drawback of Ridge Regression is that it keeps all of the parameters in the model and therefore does not implement variable selection. Hence, we chose to explore Lasso Regression which does indeed perform variable selection. Similar to ridge, the function of lasso regression also includes a shrinkage penalty. The equation below is minimized:

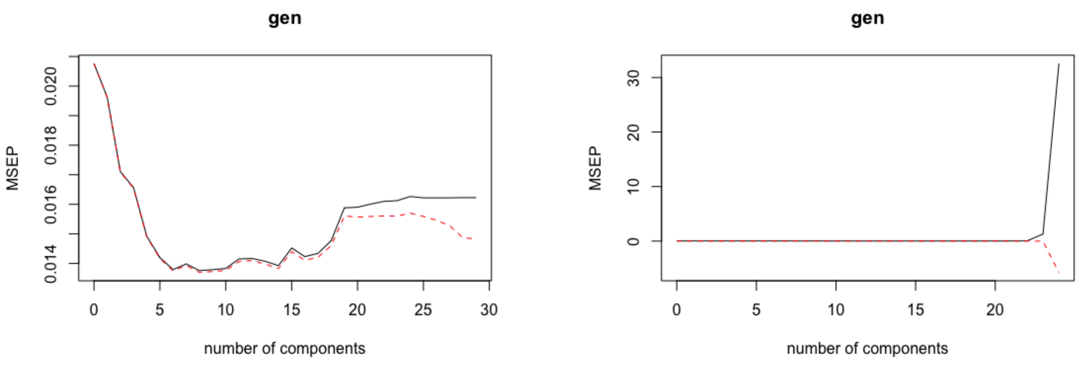
$$RSS + \lambda \sum_{j=1}^{p} |(\beta_j)| \tag{2}$$

With lasso, if λ is large enough, the shrinkage penalty has the effect of shrinking the coefficient estimates to exactly zero. Therefore, similar to least squares regression, lasso executes best subset selection. We run cross validation on both seasons, seen in Figure 9, to find the best λ.

The winter season analysis finds that when = 0.0045, it minimizes the cv error and produces a test MSE of 0.0150. Furthermore, as we stated above, lasso regression performs variable selection and tells us there are 5 non-zero coefficient estimates: iconpartly-cloudy-night, hum, temp, vis, and iconclear-night.

The summer model chose $\lambda = 0.0025$ which produces a test MSE of 0.0186. Here there are 6 non-zero coefficient estimates: iconpartly-cloudy-night, sumPartly Cloudy, sumMostly Cloudy, sumRain, iconclear-night, and appTemp.

4.4 Principle Component Regression



(a) Winter

(b) Summer

Figure 10: Cross Validation Principle Component Regression Plots

Next, we move towards regression methods that instead of shrinking/eliminating the estimate coefficients for the orginal predictors we transform the predictors and then fit a least squares. These approaches are referred to as dimension reduction methods. The first one we use is principal component regression (PCR).

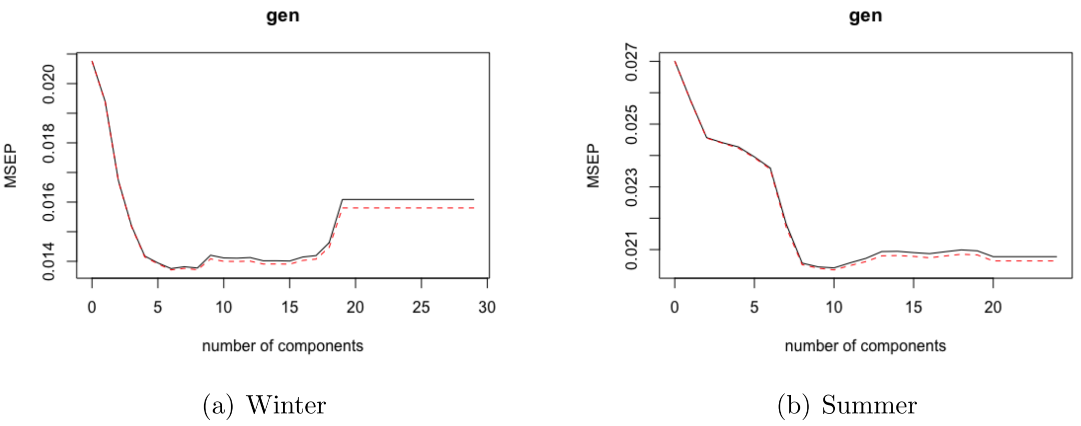
PCR's key idea is that a small number of principle components are sufficient enough to explain most of the variability in the data and the relationship with the response. As more principal components are used, the bias decreases while the variance increases. Like previous methods, this suggests to use cross validation, seen in Figure 10.

The winter model shows we should use 8 components and the summer model shows we should use 10 components. When we test the winter and summer models we get test MSE's of 0.0152 and 0.0214, respectively. A drawback to PCR is that it doesn't perform variable selection, but rather uses all of the transformed predictors.

4.5 Partial Least Squares Regression

Another disadvantage of PCR is that there's no guarantee that the principle components that explain most of the predictors will also be best for predicting the response. This is where partial least squares(PLS) regression is useful. PLS alleviates the prior problem in that it identifies new features that not only approximate the old features but also are related to the response. We again run cross validation to find M, the best number of components to use in the modeling.

Figure 11 shows the cross validation plots for each season. Our PLS winter model uses 8 components resulting in a test MSE of 0.0152 and the PLS summer model uses 10 components resulting in a test MSE of 0.0190.



(a) Winter

(b) Summer

Figure 11: Cross Validation Partial Least Squares Regression Plots

5 Non-parametric Methods

5.1 Decision Trees

Switchin gears, we wanted to try modeling using non-parametric methods. First, we look towards decision trees. Decision trees are useful in that they are more flexible in fitting the model while also keeping the relative ease of interpretability. Unlike normal regression methods, decision trees don't require the need to create dummy variables for categorical predictors or all of the regression assumptions to perform well. Overall, the big advantage of implementing a decision tree is that it grants us the ability to produce more statistically powerful predictions than regression models while also being able to display easily interpreted graphical results.

Our response variable, solar generation, is continuous so we will use regression decision tree analysis. A brief overview of this method starts with dividing the predictor space into high dimensional rectangles. We want to find boxes $R_1, ...R_j$ that minimize the residual sum of squares given by $\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y_{Rj}})^2$ where $\hat{y_{Rj}}$ represents the mean response for the training observations within the jth box.

Using the *tree* package in R, we construct a regression tree for the season we are interested in, use cross validation to select the size of the tree, and finally prune the tree down to determine which predictors to use for the aforementioned best size. We will analyze the winter season first.

The original tree that R constructs for the winter season has 15 terminal nodes, uses the predictors temp, windSpeed, press, hum, icon, appTemp, vis, windBearing, and dewPoint for the internal nodes, and has a deviance of 0.0086. Figure 12 displays this tree. The trees advantage is that it is an easily interpretable graphical image. For example, our original tree tells us that there's a median solar generation of 0.29 units

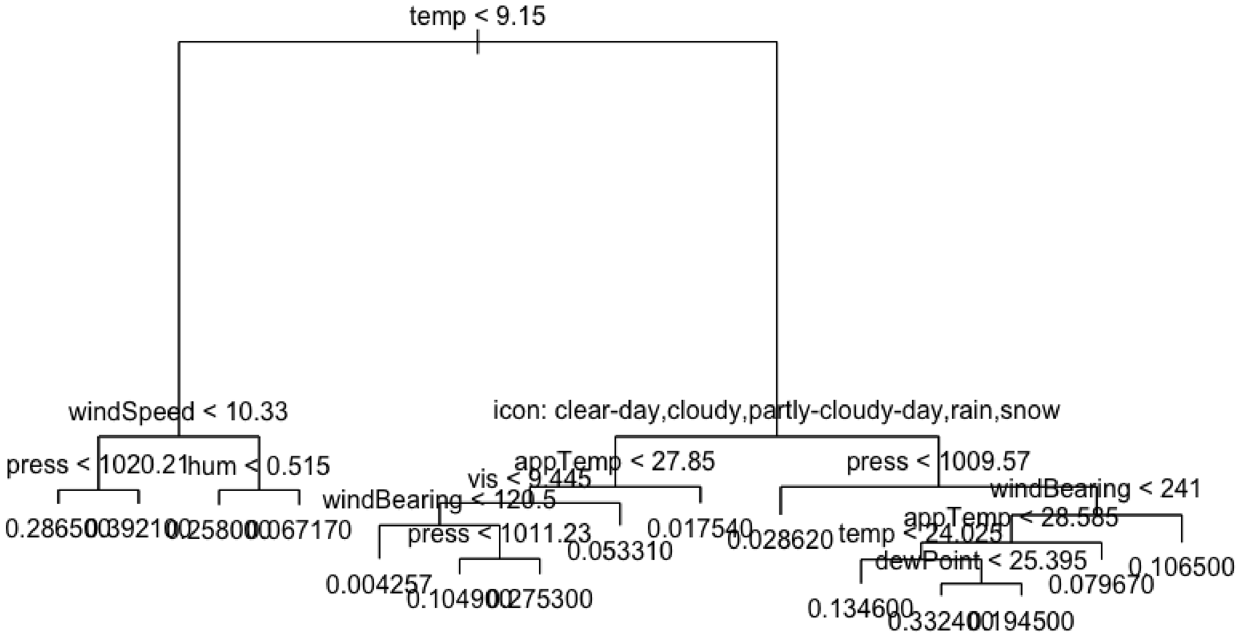


Figure 12: Original Decision Tree

when the temperature is less than 9.15 degrees, windSpeed is less than 10.33mph, and the pressure is less than 1,020. Overall, it is clear that this tree is pretty complex so we want to use cross validation to select the best tree size which will then allow us to apply pruning in finding the best predictors to keep in the tree.

Figure 13 explains that the best size tree that minimizes the cross validation error for this data is 3 which is the size we will use for pruning. The goal of pruning is to have high predictive power but not overfit. Using a size 3 tree, Figure 14 shows the pruned tree which is a lot less busy in regards to the number of variables it needs in the construction of the tree. We see that it only uses temperature and windSpeed and has a deviance of 0.01408.

Lastly, we look at how accurate the predictions are for the original and pruned tree. Figure 15 presents how well our trees do in a plot. Our initial tree, shown in part a), has a test MSE of 0.0194. Taking the square root tells us that the solar predictions are within around 0.1394 of the true median generation value. The pruned tree, part b), has a test MSE of 0.0174 and its solar predictions are within around 0.1320 of the true median generation value. It's clear that the pruned tree performs better.

Following the same progression of steps, we look at the summer season. The original tree has 16 terminal nodes, a deviance of 0.01389, and uses windBearing, cc, press, dewPoint, pricipIntensity, and hum to construct the tree. Figure 16 displays this tree. For example, our original tree tells us that when windBearing is less than 273.5, cc is greater than 0.135, and pricipIntensity is greater than 0.00055, there's a median solar generation of 0.1018 units. Again, this is a pretty involved tree so we want to use cross

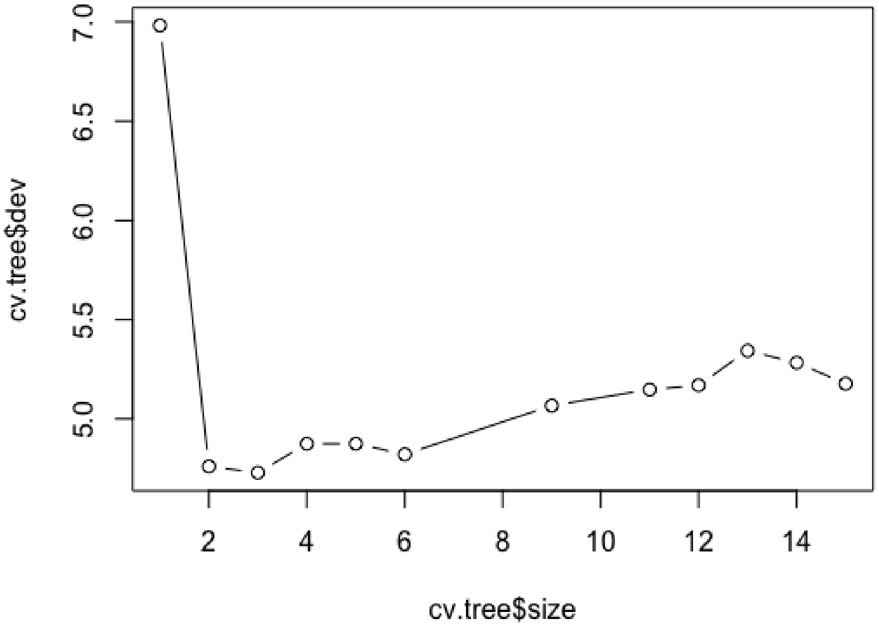


Figure 13: Cross Validation Plot

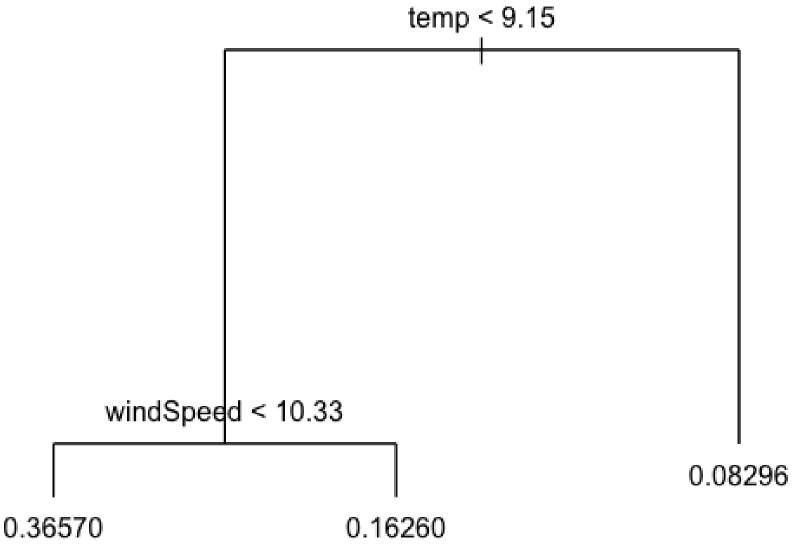
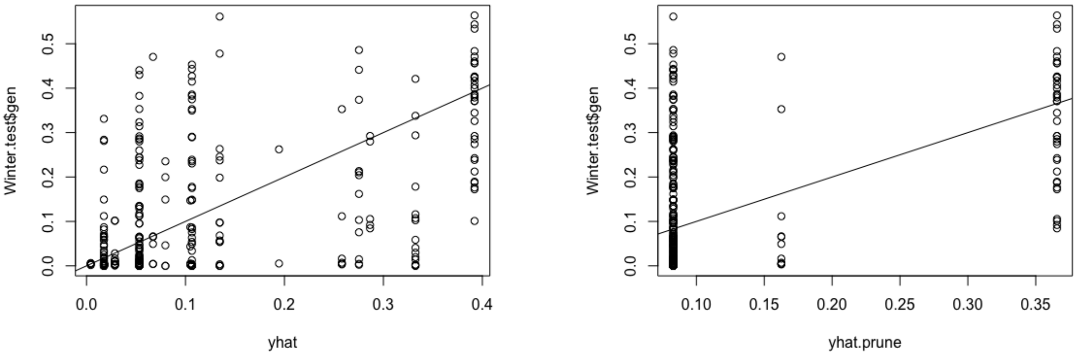


Figure 14: Pruned Tree



(a) Original Tree

(b) Pruned Tree

Figure 15: Tree Prediction Plots

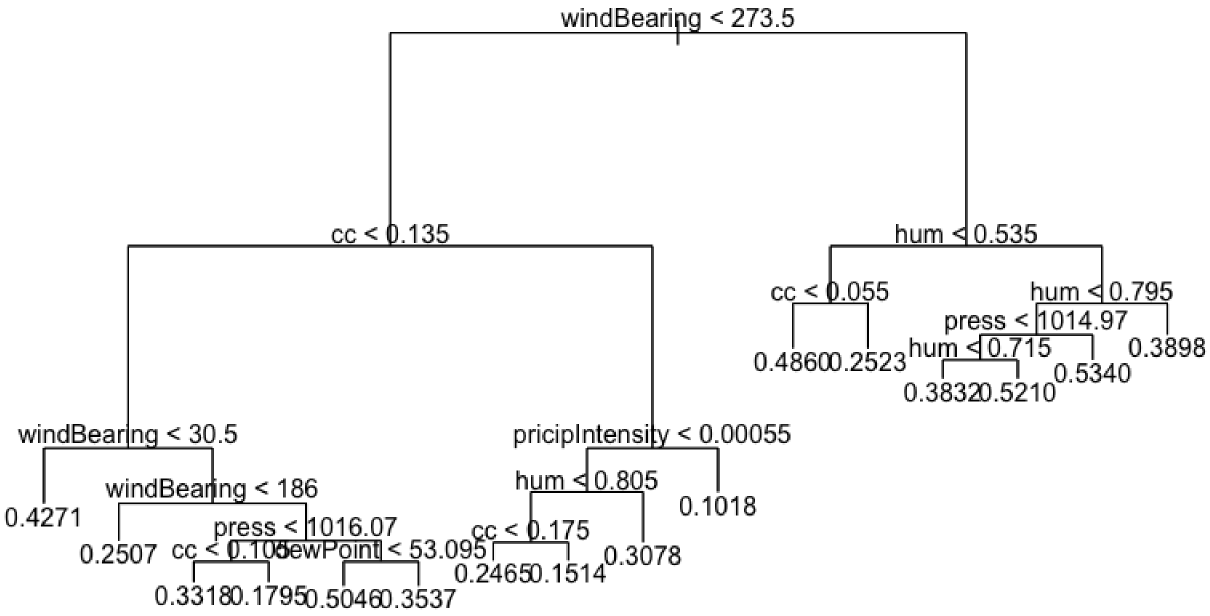


Figure 16: Original Decision Tree

validation to be able to prune it down.

The cross validation plot in Figure 17 displays that the best tree size that minimizes the cv error is size 10. Furthermore, Figure 18 shows which predictors are associated with this respective size. The pruned tree uses windBearing, cc, pricipIntensity, and hum as the internal nodes. This tree has a deviance of 0.01568.

Again, we want to see how well these trees perform in regards to their predictive

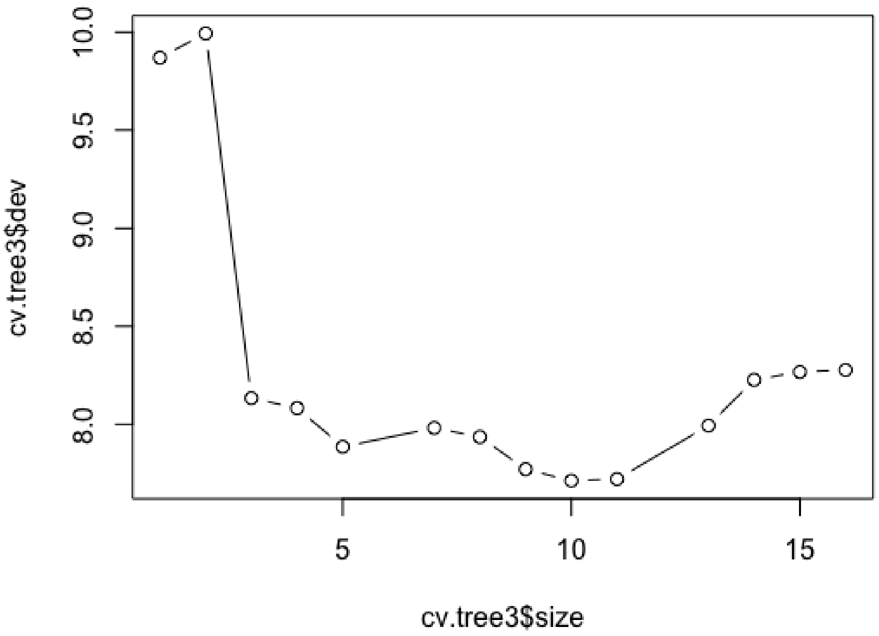


Figure 17: Cross Validation Plot

power. Figure 19 presents how well our trees perform. Our initial tree, part a), has a test MSE of 0.0236. The solar predictions are within around 0.1537 of the true median generation value. The pruned tree, part b), has a test MSE of 0.0250 and its solar predictions are within around 0.1583 of the true median generation value. Unlike the winter model, the original tree performs slightly better.

5.2 Random Forest

Although, decision trees have many advantages over regression modeling, we still did not see the predictive power that we wanted and would like to improve the model. A single decision tree may not have outstanding predictive power and also can be very non-robust in the sense that a small change in the data can cause a large change in the final estimated tree. Therefore, the last, natural progression for modeling our data set was to consider random forests.

Random forest modeling aggregates multiple decision trees which, in turn, substantially improves the predictive performance of the model. There are many ways to utilize multiple trees but we chose random forests over other methods, like bagging, because it eliminates the possibility of having extremely correlated trees. Random forests only allow a subset of predictors to be considered at each split, avoiding the issue of always choosing the strongest predictor and hence giving the other predictors a 'chance'. The usual convention is to use p/3 predictors at each split. Our data has 15 variables so we let 5 predictors be considered at each split. By decorrelating the trees like this, the average of the resulting tree is less variable and more reliable.

Figure 20 tells us the importance of each factor in the winter model. The left plot

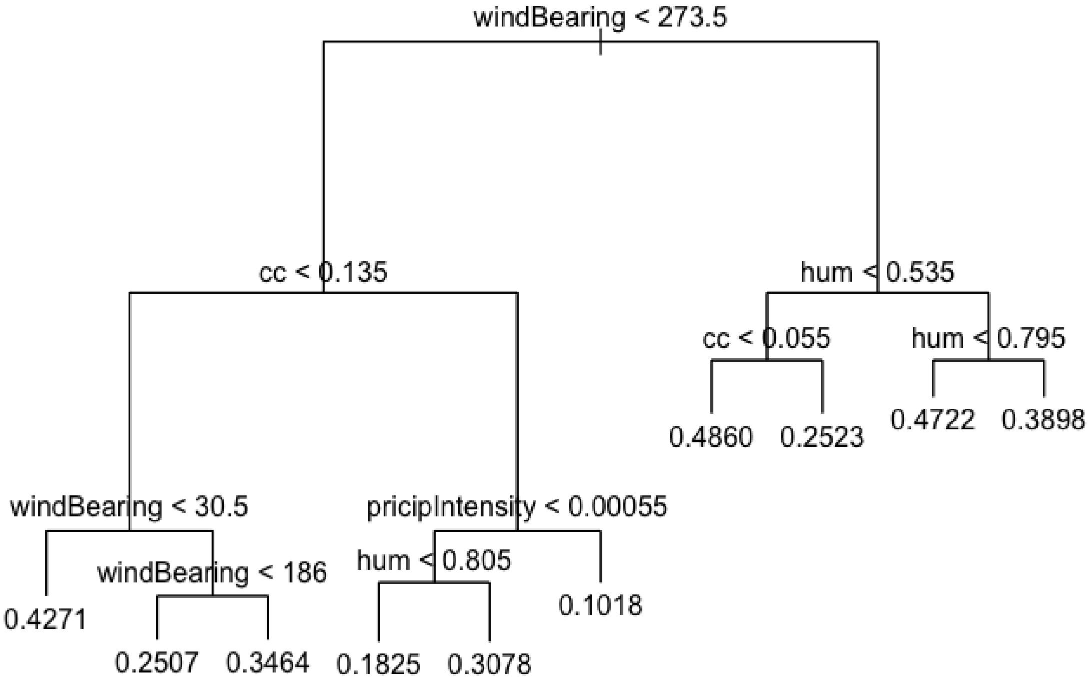
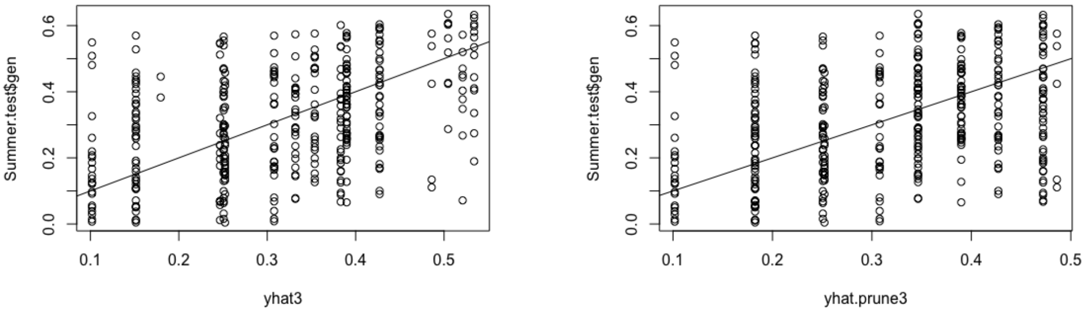


Figure 18: Pruned Tree



(a) Original Tree

(b) Pruned Tree

Figure 19: Tree Prediction Plots

shows the average percent decrease in the MSE that the variable provides and the right plot shows the purity metric of that variable. The test MSE for the the winter season using random forest modeling is 0.0081. Figure 21 shows the percent decrease in the MSE and the impurity for each variable used in the summer model. The model has a test MSE of 0.0144.

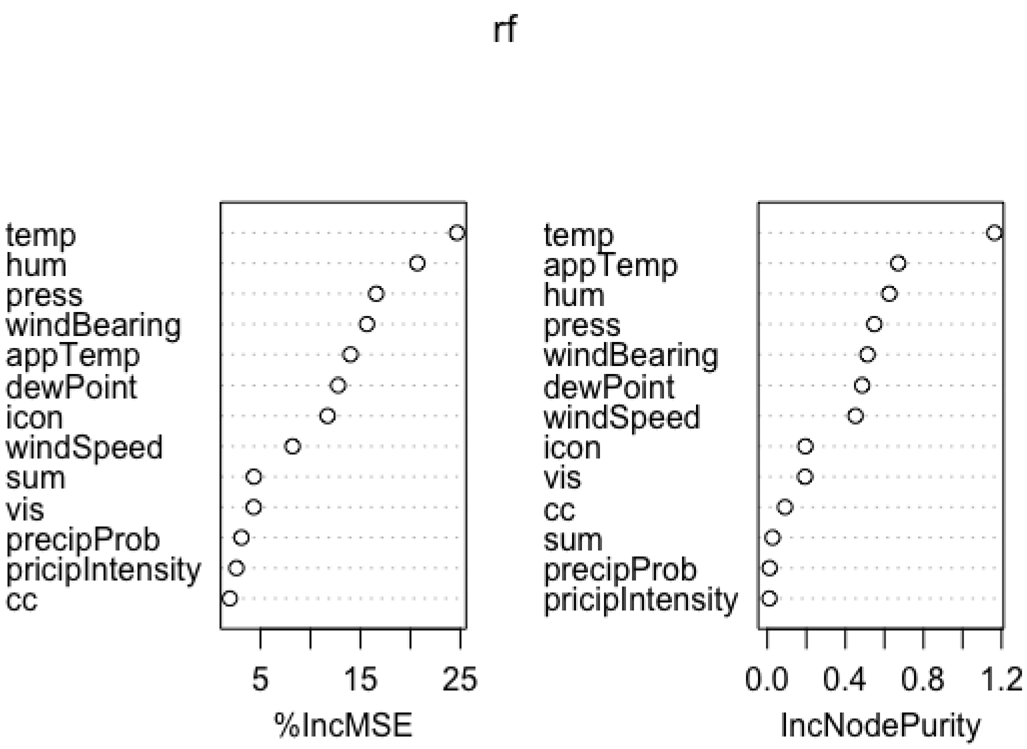


Figure 20: Winter Importance Plot

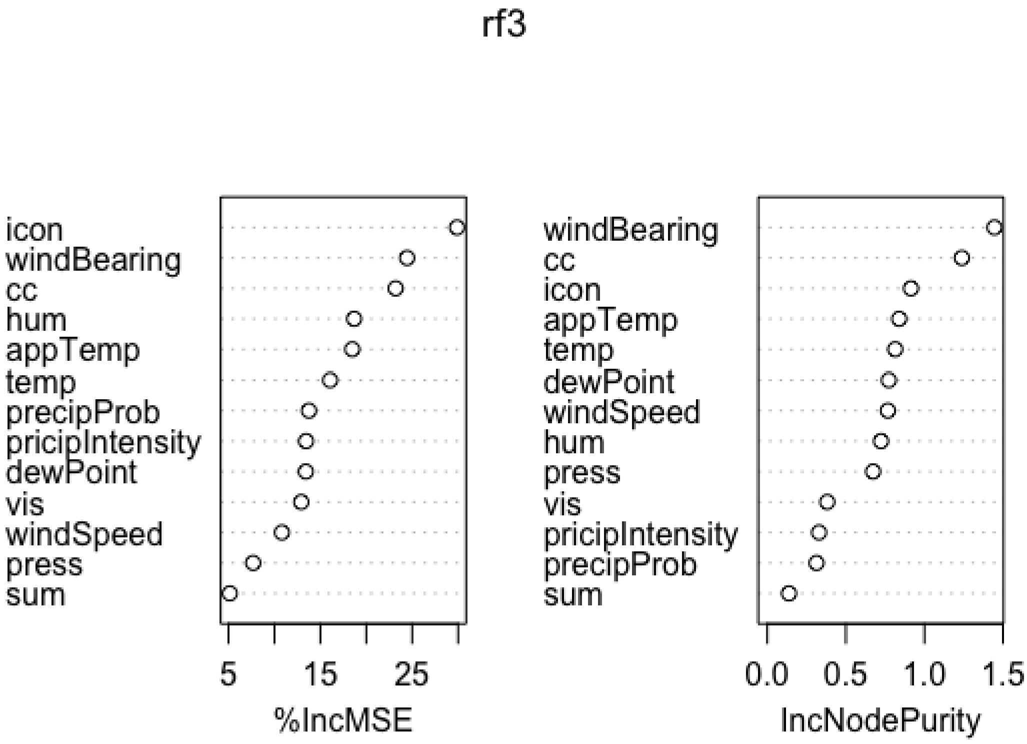


Figure 21: Summer Importance Plot

6 Conclusions

	Winter Results
Modeling Method	Test MSE
Multiple Linear Regression	0.0162
Ridge Regression	0.0153
Lasso Regression	0.0150
Principal Component Regression	0.0152
Partial Least Squares Regression	0.0152
Original Decision Tree	0.0194
Pruned Decision Tree	0.0174
Random Forest	0.0081

Table 5: Winter Method Results

	Summer Results
Modeling Method	Test MSE
Multiple Linear Regression	0.0199
Ridge Regression	0.0186
Lasso Regression	0.0186
Principal Component Regression	0.0214
Partial Least Squares Regression	0.0190
Original Decision Tree	0.0236
Pruned Decison Tree	0.0250
Random Forest	0.0144

Table 6: Summer Method Results

As we can see in Tables 5 and 6 random forest modeling performs the best in regards to predictive power for both the winter and summer seasons. This is a pretty typical result because it is a ”black box” method which means it is very flexible and should perform the best.

When we used the methods that perform variable selection, each method consistently chooses the same variables to keep in the model. One of the most interesting aspects to extract between the two seasons is that the summer model keeps the variable ”cc” (cloud coverage). Intuitively, we can presume that the winter season is almost always cloudy so it would act as a constant. Overall, there does seem to be a difference in regards to variable selection between the two ”extreme” seasons, winter and summer. One last thing to note is for each modeling method, the winter season performs better in predictions than the summer season.

Future work would include analysis by support vector machines, another ”black box” method. We might also want to look into missing data analyses. Utilizing imputation methods for missing data might present interesting results.

7 Citations

- James, Gareth, et al. An Introduction to Statistical Learning: with Applications in R. Springer, 2017.

- Weibel, Thomas. “UMassTraceRepository.” Smart - UMass Trace Repository, traces.cs.umass.edu/index.php/Smart/Smart.