**Summary of the methodology used for the**

**AMS 2013-2014 Solar Energy Prediction Contest**

**1. Summary**

As is often the case in predictive analytics, data preparation was the most important step in this project. Since the localization of the *mesonet* stations does not coincide with the position of the GEFS nodes, some transformations were necessary in the training and testing datasets. The best accuracy was achieved with boosted regression trees. An implementation of this technique is directly available in R (gbm package) with the mean absolute error (MAE). No feature selection was implemented and in order to reduce the overfitting risks, ensembling of several boosted trees was performed. Following the rules of this competition, no external data sources were used.

**2. Features Selection / Extraction**

**2.1 Features selection**

All the 75 weather features were used without any prior selection. They correspond to the 15 meteorological variables forecasted each day for 5 different hours.

We did not use the detailed 11 ensemble members with perturbed initial conditions and simply averaged them to get a single value for each variable.

**2.2 From GEFS forecasts to mesonet forecasts**

GEFS positions do not coincide perfectly with *mesonet* position. For each of the 98 *mesonet,* we made a linear interpolation of the four nearest GEFS points (weighted by the distance). The following figure illustrates this:

GEFS1

GEFS2

d1

d2

d4

d3

GEFS3

GEFS4

We used the following formula:

Vmesonet = (w1 x VGEFS1 + w2 x VGEFS2 + w3 x VGEFS3 + w4 x VGEFS4) / (w1 + w2 + w3 + w4)

With wi = Max(0, 1 – di) where d is the Euclidian distance from the *mesonet* to the near GEFS node (assuming that the we have a square

A specific program has been developed to directly get these *mesonet* forecasts form the NetCDF files (see §4)

**2.3 Additional features**

The following values were provided and have therefore been used:

* The elevation, the latitude and the longitude of the *mesonet* stations (i.e. the position of the *mesonet* stations)
* The month of the observation expressed as a real value (12 x day of the year / 365)

Following the first modelling, the following variables clearly appeared as the most important: dswrf\_H2, dswrf\_H3, dswrf\_H4, pwat\_H1, pwat\_H2 and pwat\_H3 (with the following convention: H0: 12.00UTC, H1: 15.00UTC, H2: 18.00UTC, H3: 21.00UTC, H4: 24.00UTC). For these 6 variables we therefore created the following derived features:

|  |  |
| --- | --- |
| Variable | Derived variables |
| dswrf\_H2 | dswrf\_H2\_SW, dswrf\_H2\_W,dswrf\_H2\_NW, dswrf\_H2\_S, dswrf\_H2\_N, dswrf\_H2\_SE, dswrf\_H2\_E, dswrf\_H2\_NE dswrf\_H2\_Ext |
| dswrf\_H3 | dswrf\_H3\_W, dswrf\_H3\_S, dswrf\_H3\_N, dswrf\_H3\_E, dswrf\_H2\_Ext |
| dswrf\_H4 | dswrf\_H4\_SW, dswrf\_H4\_W, dswrf\_H4\_NW, dswrf\_H4\_S, dswrf\_H4\_N, dswrf\_H4\_SE, dswrf\_H4\_E, dswrf\_H4\_NE, dswrf\_H2\_Ext |
| All dswrf | dswrf\_D12, dswrf\_D23, dswrf\_D34 |
| pwat\_H1 | pwat\_H1\_W, pwat\_H1\_S, pwat\_H1\_N, pwat\_H1\_E, pwat\_H1\_Ext |
| pwat\_H2 | pwat\_H2\_W, pwat\_H2\_S, pwat\_H2\_N, pwat\_H2\_E, pwat\_H2\_Ext |
| pwat\_H3 | pwat\_H3\_W, pwat\_H3\_S, pwat\_H3\_N, pwat\_H3\_E, pwat\_H3\_Ext |
| All pwat | pwat\_D01, pwat\_D12, pwat\_D23, pwat\_D34 |

The new derived variables with extension \_Dxy correspond to the time variation between 2 forecasts separated by 3 hours. For example: dswrf\_D12 = dswrf\_H2 - dswrf\_H1.

All the other derived variables are linked to the spatial variations. To illustrate their meaning, let’s consider the following figure:

VSW

VNW

VNE

VSE

VN

VS

VE

VW

V

Lon + 1°

Lat + 1°

Let’s denote:

* V a weather variable forecasted for a *mesonet*
* VSW, VW, VNW, VN, VNE, VE, VSE and VS the forecasted values of V for (imaginary) points separated by 1° in longitude and/or latitude.

Then we define the following derived variables:

V\_SW = V – VSW V\_W = V – VW V\_NW = V – VNW

V\_N = V – VN V\_NE = V – VNE V\_E = V – VE

V\_SE = V – VSE V\_S = V – VS V\_Ext = V/2 + (VSW+VW+VNW+VN+VNE+VE+VSE+VS)/8

For each *mesonet* station, we also added the overall average of energy production and the difference between the actual elevation of the station and the elevation of the nearest GEFS point. At the end, we therefore have 75 + 49 + 4 + 2 = 128 explanatory variables.

**3. Modeling Techniques and Training**

**3.1 Datasets**

All the data from the 98 *mesonet* stations were gathered. It means that there is one training set, one testing set, and finally one single model for all stations. Some trials were made with separated datasets for each station but they never gave more accurate predictions.

Consequently, the training dataset has 501074 rows and the testing dataset has 176068 rows.

We did not perform any data cleansing (only the transformations described in the previous paragraph).

**3.2 Modeling techniques**

We selected gradient boosting techniques and used the R gbm package. The combination of several boosted regression trees enabled to improve the accuracy (by reducing the overfit of this technique).

For each of the boosted trees, we use the following training settings:

* Mean Absolute Error 🡺 distribution="laplace"
* Number of expansions between 2000 and 3000 🡺 n.trees=2000 or 3000
* Depth of the trees between 6 or 8 🡺 interaction.depth = 6, 7 or 8
* Learning rate of 0.05 🡺 shrinkage=0.05
* Out Of the Bag proportion: 30% 🡺 bag.fraction=0.7

NB: Random forests were also tested but not retained because less accurate.

**4. Code Description**

A specific C# program has been developed to directly get the forecasts at the *mesonet* form the NetCDF files with all the transformations described in §2.2. It uses the *Microsoft.Research.Science.Data* library.

The steps are the following:

1. Create csv files from the NetCDF files with the C# program
2. Import the csv files *Microsoft Excel* where basic formulas are used to calculate the derived features described in §2.3.
3. Once the training and testing datasets are built in *Microsoft Excel*, export the datasheets in csv files (DataTrain.csv and DataTest.csv) in order to be directly readable in R environment.

In R, the following instructions were used directly in the command line:

library(gbm) ‘ Load the gbm package

DataTrain = read.table(“DataTrain.csv", header = T, sep = ",", dec = ".") ‘ Load training set

DataTest = read.table(“DataTest.csv", header = T, sep = ",", dec = ".") ‘ Load testing set

Model.gbm <- gbm(Power ~ .,distribution="laplace", data=DataTrain, n.trees=3000,

interaction.depth =6, shrinkage=0.05, bag.fraction) ‘ Train the model

Prediction <- predict.gbm(Model.gbm, DataTest, n.trees=3000) ‘ Predict the test set

**5. Results**

The predictions of 12 different boosted trees are averaged to increase the accuracy. It was also noted that the accuracy of the prediction was slightly improved by increasing the global output by 1% (an interpretation could be that the efficiency of the mesonet has been improved between 1994-2007 and 2008-2009).