CS 519 Project Report

Randall Woodall
New Mexico State University
hossrw@nmsu.edu

Emrah Sariboz
New Mexico State University
emrah@nmsu.edu

Aya Elsayed
New Mexico State University
aynasser@nmsu.edu

ACM Reference Format:
Randall Woodall, Emrah Sariboz, and Aya Elsayed. 2020. CS 519 Project Report. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 PROBLEM STATEMENT

Prediction of power production for solar panels from weather information.

1.1 Motivation

Since starting up generators is costly, having an idea of how much power a utility would need to produce can help the utility to reduce costs of generation. There exists an abundance of data in relating to solar production, and an even greater abundance relating to weather and forecasting. Combining weather history, weather forecasting, and information about solar production could lead to a schedule of solar power production that would allow, in the long run, utilities to create more optimal generator schedules and power rates. This type of scheduling would help the consumer to save money by allowing the producer to avoid startup costs and idling costs. Because of this, new and more accurate methods will be valuable to all parties involved in electrical power.

1.2 Direct Problem Definition

Given five second solar production data in the form of power, and also given historical weather data including wind, temperature, humidity and pressure, can we make semi-accurate predictions about the solar data at the time of the weather data? To restate this problem in fewer words, can you use current state of the weather to predict current output of a solar panel? If this is the case, then a day-ahead power forecast should be directly correlated to a day-ahead weather forecast. The goal of this project is to show that, given current data, we can predict what a solar panel produces, and by logical extension, if we can predict what it will produce tomorrow.

1.3 Problem Limitations

We acknowledge that we have a hard limit on the accuracy of any day-ahead power predictions, based in the uncertainty of day-ahead weather predictions. Going back to statistics, this limit on accuracy and, therefore, the introduced error, goes back to the statement, "Garbage in, garbage out". Due to this, we will eliminate the space for introduced error by restraining our semester project to include only the historical data, where we have known values and actual measurements. If we can assume a normal distribution around the predicted weather values wherein the actual weather values will fall, then we can incorporate this into our prediction for power produced to give a normal distribution showing what power will probably be.

2 DATA SOURCES

2.1 Solar Power Data

Our solar data is being collected by direct monitoring of solar panels on the roof of Thomas and Brown hall. This data is being collected on a 5 second interval for use with another experiment, which is a far higher frequency than we can retrieve weather data at. Specifically, our data is the power produced by a set of four panels. Since there are four panels hooked up to a three phase output, we simply added the three phase output power of all the panels together to get the total power out. To get hourly three phase powers to line up with weather data, we took a rolling average over the hourly values to pair up with weather data.

2.2 Weather Station Data

By using our own weather station on Thomas and Brown hall's roof, We are getting a time stamped weather data. Along with the time, we are getting other 26 attribute each five minutes.

2.3 Data Preprocessing

Before we apply any pre-processing technique, the shape of the dataset was (3465, 28). Also, we disregarded the time column for now. At some point in the day, a solar panel couldn't get the power data. Thus, some values of the power attribute had NaN values. The first thing we did was to remove these rows and the shape of the resulted dataset was (3465, 28). And then, we used the Sklearn SimpleImputer library to impute NaN values with the mean of the corresponding columns. The reason we decided to impute rather than dropping is only 10 percent of this column was missing. Dropping these columns could result in serious information losses. The details of the preprocessing steps can be found in the GitHub repository under Stage 3.

3 ALGORITHMS TESTED

3.1 Linear Regression

For linear regression analysis, we first used a correlation matrix to find those features that have a high correlation with the power attribute. In other words, we first conducted a feature section. Upon this process, we have left with two features. By applying linear

Conference'17, July 2017, Washington, DC, USA
© 2020 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

regression, we were able to get an r^2 score of 0.79 on training dataset and 0.80 on testing dataset.

3.2 Polynomial Regression

To use polynomial regression, we first applied PCA to reduce down to 3-dimensions. This is to limit the size of the resulting polynomial coefficient size. Using this technique, we were able to get an r^2 score of 0.819 on testing data, with .821 on training data. Although this is better than standard linear regression, we are hoping to do better than this with a different algorithm.

3.3 Random Forest Regression

Applying Random Forest Regressor from scikit learn library with 100 estimator and a random state of one. By splitting the data by 30 percent for testing data. Achieving a r^2 score for train data of 0.997 and a r2 score for test data of 0.981.

3.4 Algorithm selection

Given that the random forest regression gives the best r^2 on test data, at around .18 above the other methods, we will be using this for our study. The spike in accuracy when compared to the other algorithms is expected, given that ensemble methods usually have this type of behavior compared to single classifiers.