Machine Learning Engineer Nanodegree
Capstone Proposal

Title: Appliance Energy Prediction

Domain Background

Modern economies depend on the reliable and affordable delivery of electricity. Yet many places in the world are still affected by power outages.

One significant cause of power outages is overloading. Overloading occurs when too much power is drawn from an electric circuit at once. This can be avoided if we know when excess electricity is going to be used. Heating and cooling appliances consume the most power in a household.

The goal is to predict the electricity usage of heating and cooling appliances in a household based on internal and external temperatures and other weather conditions. Each observation measures electricity in a 10-minute interval. The temperatures and humidity have been averaged for 10-minute intervals.

Related research and previous work:

http://dx.doi.org/10.1016/j.enbuild.2017.01.083

(Research Paper)

https://github.com/LuisM78/Appliances-energy-prediction-data

(Previous work by the author)

Problem Statement

Predict the electricity usage of heating and cooling appliances in a household based on internal and external temperatures and other weather conditions.

Datasets and Inputs

Dataset link: http://archive.ics.uci.edu/ml/datasets/Appliances+energy+prediction

The author has provided separate training and testing data files in his GitHub repository (link above).

Dataset Information[2]:

The data set is at 10 min for about 4.5 months. The house temperature and humidity conditions were monitored with a ZigBee wireless sensor network. Each wireless node transmitted the temperature and humidity conditions around 3.3 min. Then, the wireless data was averaged for 10 minutes' periods. The energy data was logged every 10 minutes with m-bus energy meters. Weather from the nearest airport weather station (Chievres Airport, Belgium) was downloaded from a public data set from Reliable Prognosis (rp5.ru), and merged together with the experimental data sets using the date and time column. Two random variables have been included in the data set for testing the regression models and to filter out non predictive attributes (parameters).

The dataset has 19375 instances and 29 attributes including predictors and target variable.

The training data provided by author contains 14803 instances and testing data contains 4932 instances.

Attribute Information[2]:

- date time year-month-day hour:minute:second
- **Target Variable: Appliances, energy use in Wh**
- lights, energy use of light fixtures in the house in Wh
- T1, Temperature in kitchen area, in Celsius
- RH_1, Humidity in kitchen area, in %
- T2, Temperature in living room area, in Celsius
- RH_2, Humidity in living room area, in %
- T3, Temperature in laundry room area
- RH_3, Humidity in laundry room area, in %
- T4, Temperature in office room, in Celsius
- RH_4, Humidity in office room, in %
- T5, Temperature in bathroom, in Celsius
- RH_5, Humidity in bathroom, in %
- T6, Temperature outside the building (north side), in Celsius
- RH_6, Humidity outside the building (north side), in %
- T7, Temperature in ironing room, in Celsius
- RH_7, Humidity in ironing room, in %
- T8, Temperature in teenager room 2, in Celsius
- RH_8, Humidity in teenager room 2, in %
- T9, Temperature in parents' room, in Celsius
- RH_9, Humidity in parents' room, in %
- To, Temperature outside (from Chievres weather station), in Celsius
- Pressure (from Chievres weather station), in mm Hg
  RH_out, Humidity outside (from Chievres weather station), in %
- Wind speed (from Chievres weather station), in m/s

- Visibility (from Chievres weather station), in km
- Tdewpoint (from Chievres weather station), Â°C
- rv1, Random variable 1, non-dimensional
- rv2, Random variable 2, non-dimensional

Where indicated, hourly data (then interpolated) from the nearest airport weather station (Chievres Airport, Belgium) was downloaded from a public data set from Reliable Prognosis, rp5.ru. Permission was obtained from Reliable Prognosis for the distribution of the 4.5 months of weather data.

Solution Statement

Generally, regression is used to solve these kinds of problems. Some common regression methods are:

- Linear regression
- Polynomial regression
- Ridge regression
- LASSO regression

Ridge and LASSO regression are regularized methods.

A linear regression equation looks like this:

$$Y = \theta_1 X_1 + \theta_2 X_2 + \ldots \theta_n X_n$$

Where Y is the dependent variable, X's are the independent variables and the thetas are the coefficients. These coefficients are basically weights which determine importance of that particular variable.

In polynomial regression, at least one variable has a degree of more than 1. So the best fit line becomes a curve.

In regularization we keep the same number of attributes, but reduce the magnitude of coefficients. This is done by penalizing them in two ways:

- Adding them to the loss function (L1 Regularization)
- Adding their squares to loss function (L2 regularization)

Benchmark Models

The author used 4 models[3], which are:

- Multiple Linear regression (LM)
- Random Forest (RF)
- SVM with Radial kernel (SVM radial)
- Gradient Boosting Machine (GBM)

The GBM had a R2 score of 0.97 and RF of 0.92 in the training set. For the testing set the R2 score for GBM was 0.58.

The author provides separate training and testing datasets in his GitHub repository[1] as mentioned above.

Evaluation Metrics

The metrics commonly used to evaluate regression models are:

- Mean absolute error
- Mean squared error (MSE)
- Root Mean Squared error (RMSE)
- R2 score

Project Design

I will follow the following sequence of steps.

1. **Data Visualization**: Visually representing data to find correlations between attributes and target variable. This will also help in finding visible patterns in the dataset.

2. **Preprocessing the data**: Scaling and Normalizing the data. As I will use the data available on the UCI ML repository[2] and not the one provided by the author on GitHub, I will have to split the data into training, validation and testing datasets.

3. **Feature Engineering**: Finding relevant features, engineering new features, etc.

4. **Model Selection**: Experiment with various models and algorithms to find the best one. Ex. Multi-layer perceptron which has not been used by the author. Others are of course regression techniques, SVM and gradient boosting.

5. **Model Tuning**: Fine tune the selected algorithm to increase performance while making sure it does not overfit.

6. **Testing**: Test the model on the testing dataset.

References

[1]https://github.com/LuisM78/Appliances-energy-prediction-data

[2]http://archive.ics.uci.edu/ml/datasets/Appliances+energy+prediction

[3]http://dx.doi.org/10.1016/j.enbuild.2017.01.083