Multivariate Time-Series Forecasting *for*

Energy Consumption Prediction

DS 5110

December 8th, 2019

Authors

Saurabh Parkar

Vaibhav Saraf

Sanjan Vijayakumar

Kyu Ha Kim

Multivariate Time-Series Forecasting for Energy Consumption Prediction

Authors: Vaibhav Saraf, Saurabh Parkar, Sanjan Vijayakumar, Kyu Ha Kim

Summary: A large-scale investor/financial institution would only express interest in investing in energy-saving investments if they consider it to be profitable in the long run. Ideally, this would mean that any energy-saving methods employed would require them to wait a considerable duration of time to determine if the methods employed actually save enough energy for it to be a profitable investment. By predicting the future energy consumption of different meters based on past energy consumption, we would be able to provide better estimates of these energy-saving investments thus inclining them to invest in this area to enable progress in building efficiencies.

The project seeks to predict the energy meter reading of buildings across four energy types - (0) Electricity, (1) Chilled Water, (2) Steam, and (4) Hot Water, based on historic usage rates and observed weather over a one-year timeframe. Our primary goal is to develop a model to accurately predict the energy meter reading of each energy type for each of the buildings in our dataset.

Our methods include developing time-series forecasting models using LSTM (Long Short-Term Memory) and classical models using Linear Regression and LGBM (Light Gradient Boosting Method) and determine if our dataset works best by a time-series approach or a regression approach.

Dataset:

The dataset that we used in this project includes three CSV files:

(1) ***Data.csv*** contains hourly data on meter readings of each meter in each building

	building_id	meter	timestamp	meter_reading
0	0	0	2016-01-01 00:00:00	0.0
1	1	0	2016-01-01 00:00:00	0.0
2	2	0	2016-01-01 00:00:00	0.0
3	3	0	2016-01-01 00:00:00	0.0

Figure 1. Data CSV

(2) ***Building_Metadata.csv*** contains a distinctive building id, and various features of a building such as its primary use (e.g. Education, Lodging/Residential, etc), size in square feet, the year it was first built, and how many floors it has.

	site_id	building_id	primary_use	square_feet	year_built	floor_count
0	0	0	Education	7432	2008.0	NaN
1	0	1	Education	2720	2004.0	NaN
2	0	2	Education	5376	1991.0	NaN
3	0	3	Education	23685	2002.0	NaN

Figure 2. Building metadata CSV

(3) ***Weather_data.csv*** contains hourly data on various weather features in each site that affect the building's energy consumption. The weather features include air-temperature, cloud coverage, dew point temperature, precipitation depth level, sea level pressure, wind speed and direction

	site_id	timestamp	air_temperature	cloud_coverage	dew_temperature	precip_depth_1_hr	sea_level_pressure	wind_direction	wind_speed
0	0	2016-01-01 00:00:00	25.0	6.0	20.0	NaN	1019.7	0.0	0.0
1	0	2016-01-01 01:00:00	24.4	NaN	21.1	-1.0	1020.2	70.0	1.5
2	0	2016-01-01 02:00:00	22.8	2.0	21.1	0.0	1020.2	0.0	0.0
3	0	2016-01-01 03:00:00	21.1	2.0	20.6	0.0	1020.1	0.0	0.0
4	0	2016-01-01 04:00:00	20.0	2.0	20.0	-1.0	1020.0	250.0	2.6

Figure 3. Weather data CSV

Feature Engineering and Exploratory Data Analysis:

Exploratory Data Analysis is an important step in any project to analyze and understand the data and the problem better. Feature Engineering is an essential step in data preparation as it helps us consider the important features in the data and perform any necessary transformations to prepare the data for modeling.

(i) Combining the datasets:

The multiple datasets were combined using joins by appropriate keys to have a single combined data CSV

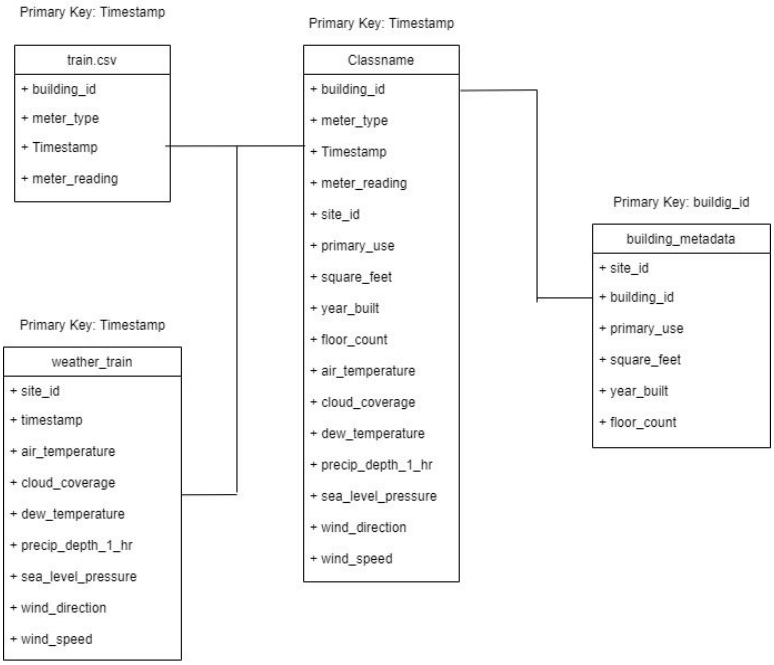

Figure 4. CSV files combined

(ii) Missing Value Imputations:

The dataset features have a lot of missing values. These missing values have been properly imputed before modeling the data. This process is necessary as a large number of missing values will affect the performance of our models on the data.

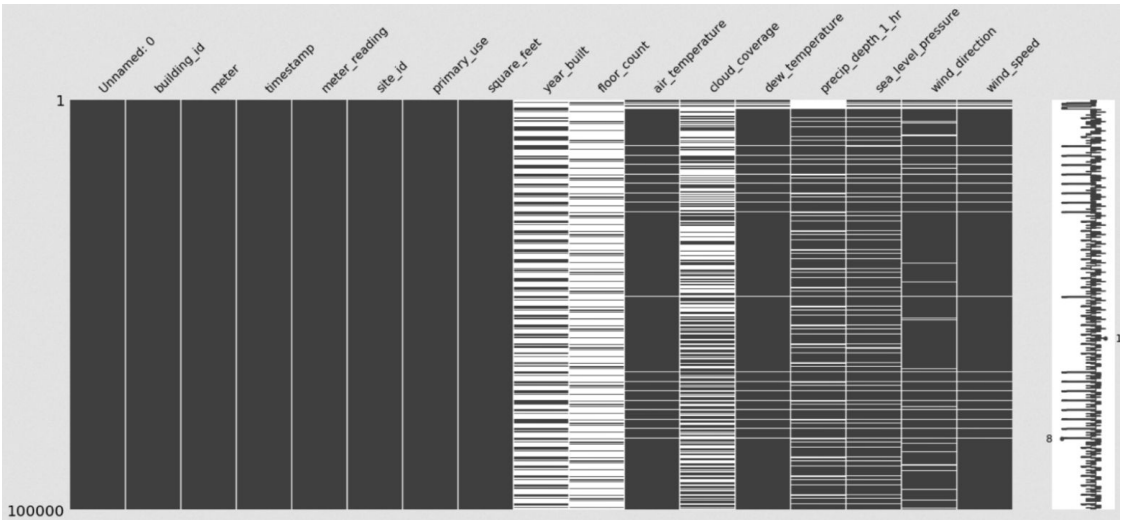

Figure 5. Missing values plot

Figure 5 represents the missing value plot of the dataset. The white spaces in the columns represent missing values. This plot can be used to identify both the number of missing values as well as the pattern of missing values which helps us impute them suitably.

Imputation of missing values in the dataset is performed as shown in Table 1 below.

Variable Name	How values change over time	Type of Values	Frequency of missing values	Percent of missing values	Imputation Method
floor_count	-	-	-	82.65%	Dropped
year_built	-	-	-	59.99%	Dropped
cloud_coverage	Changes erratically	Categorical	Discontinuous over short range	43.65%	Forward Fill
precip_depth_1_hr	Mostly Remains same, Changes erratically	Continuous	Discontinuous over short range	18.54%	Forward Fill
wind_direction	Changes erratically	Continuous	Continuous but over short range	7.16%	Mean Value
sea_level_pressure	Changes Gradually	Continuous	Discontinuous over short range	6.09%	Forward Fill
wind_speed	Changes erratically	Continuous	Continuous but over short range	0.71%	Mean Value
dew_temperature	Changes gradually	Continuous	Continuous but over short range	0.49%	Mean Value
air_temperature	Changes gradually	Continuous	Continuous but over a short range	0.47%	Mean Value

Table 1. Missing value imputations

We have performed forward filling imputation for features that have discontinuous frequency of missing values. Features containing a continuous range of missing values are grouped by their site id and imputed with their means.

Meter reading values up to building #104 had value 0 and thus were dropped.

(iii) Label Encodings:

The feature ‘primary_use’ had values indicating the different types of buildings (eg., Education, Office, Lodging/Residential) and hence label encoding was performed on them to convert these character values to numerical values.

(iv) Feature Transformations:

A log transformation was performed on meter readings to make them easier to work with as meter reading values were completely skewed.

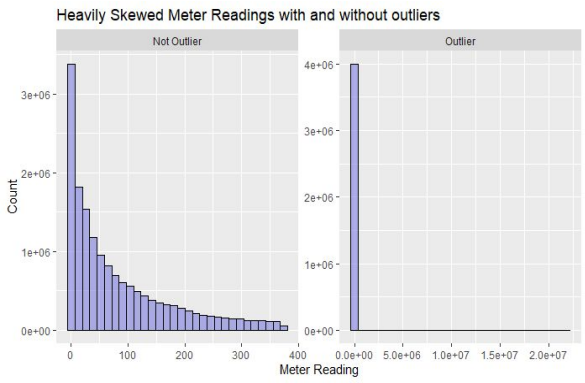

Figure 6a. Skewed meter readings

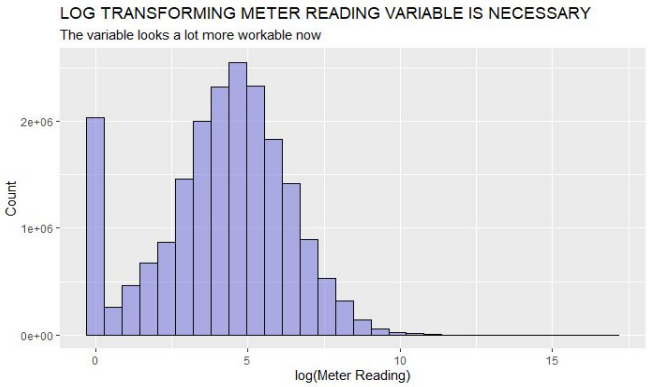

Figure 6b. Log transformation of meter readings

(v) Feature Visualizations:

a. A line plot of the four different meter types throughout the year indicated that electricity and chilled water meter readings increased over the summer and hot water and steam meter readings peaked during the winter.

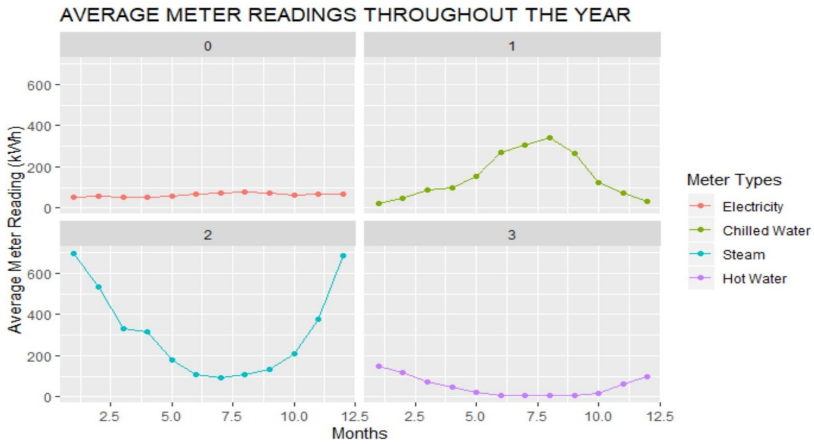

Figure 7. Meter readings for each meter type

b. A density plot for the meter types indicated that the electricity meter was the most efficient meter whereas the hot water meter was the least efficient which is evident from figure 8 below.

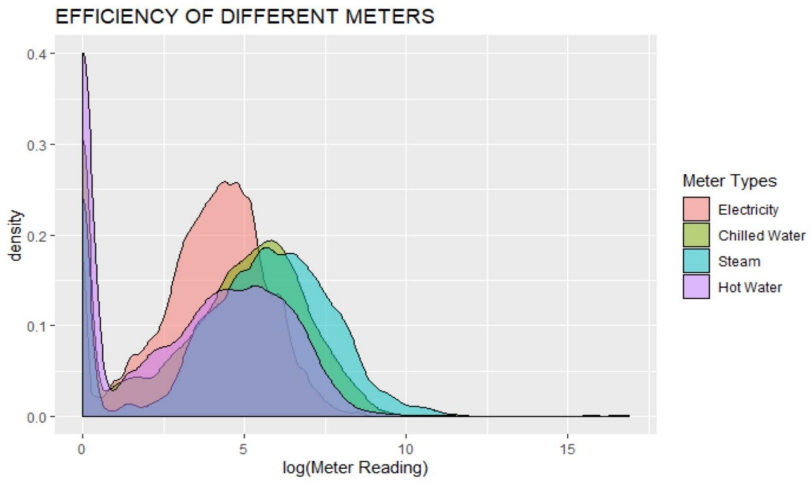

Figure 8. Efficiency of each meter type

c. After plotting the mean meter reading plot for the entire year, we noticed that the meter readings from July to November had an extremely less value of meter readings. Filtering by site_id >> primary_use >> meter type >> building_id where ">>" represents each level of filtering showed us that a similar plot was seen in site_13 >> education use >> meter 2 >> building 1099. This was clearly an outlier as the meter reading values were in the range of 10^7 for this building in comparison to other buildings that had values in the range of 10^3. Thus, we removed the building #1099 from our observations and the resulting plot we obtain was more on the lines of what we expected as shown in the figure.

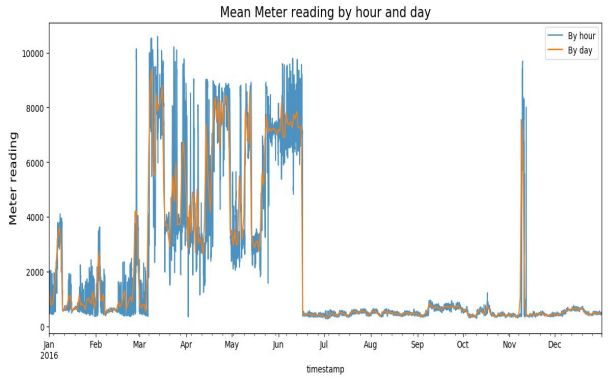

Figure 9a. Meter readings before removing building 1099

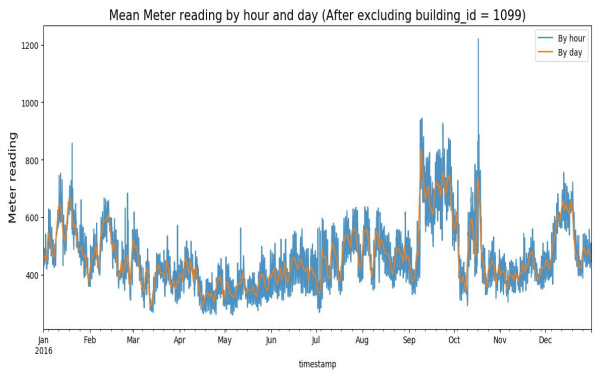

Figure 9b. Meter readings after removing building 1099

Modeling:

Our data is a hierarchical time-series data consisting of equidistant time-ordered observations taken over a period of 1 year. Time series adds an explicit order dependence between observations which is usually modeled through a stochastic process Y(t), i.e. a sequence of

random variables. In a forecasting setting, we find ourselves at time t and we are interested in estimating Y(t+h), using only information available at previous timestamps.

(i) The **Long Short Term Memory** neural network is a type of a Recurrent Neural Network (RNN) that uses previous time events to inform the later ones. LSTMs have gate mechanisms that help to regulate information and a cell state which helps to maintain information about previous predictions over long durations of time. Thus, LSTMs become convenient to model sequence data and works really well for time series forecasting.

To investigate how well these models work, we tried to predict the meter reading for the current timestamp based on the previous 10 observations using LSTMs. The data was sequentially split into 80% train and 20% test data. We obtained an RMSE of 24.80 and an MAE of 4.66 which was not as good.

(ii) **Linear Regression** provides a method to predict the target variable by fitting the best linear relationship between the target and the explanatory variables. The best fit is obtained on the basis of the mean squared distance of each observation point from the line of regression.

(iii) **Decision Tree Regressor** is a supervised learning model that uses a tree structure or is a model of decisions and their possible results, including outcomes, input costs, and utility.

(iv) **Light Gradient Boosting Method** (LGBM) is a newly established variant of Gradient Boosting, developed by Microsoft in 2017. The core part of the LGBM is boosting, one of the important methods in ensemble modeling. Boosting combines the combination of weak learners by applying them iteratively. In each iteration, more weight is given to the high residual data row. In each sample, the number of rows that have high residual is included and thus each model focuses more on the mistakes made by the previous model.

Gradient boosting is a method where the loss is minimized using Gradient Descent. Each tree is fitted on the residuals of the previous model rather than the actual target value. i.e

$$f(t_2) = f(t_1) + h(t_1)$$

Here, t_1 = Iteration 1, t_2 = Iteration 2, $h(t)$ = Model created for the residuals
This means that the model at iteration 2 is the combination of model at iteration 1 ($f(t_1)$) and the model created on the residual of the iteration 1 ($h(t_1)$). In the end, predictions from each tree is scaled with the factor of gradient or learning rate and added together to get the final predictions.

The previous Gradient Boosting models suffered low scalability and efficiency when data size becomes large, as the resampling of larger data based on weights became more time-consuming. LGBM uses Histogram based method for finding the candidate splits in Decision Tree to reduce the time complexity while resampling. LGBM employs GOSS (Gradient-Based One Side Sampling) which retains instances with large gradients while performing random sampling on instances with small gradients.
As our dataset contains more than 20 million rows, we applied LGBM to reduce the training time. We built 100 trees with the local threshold of 1280 leaves per tree.

Discussion:
Our dataset consists of observations taken over just a one-year timeframe which is not enough to capture a significant trend or seasonality required by the time series forecasting models. Moreover, the dataset contains information about 1449 different buildings and each building has 4 different types of meters, which means we have a total of 1449 x 4 = 5796 different time series in our dataset. Training a model for each one of these time series and generating predictions separately is computationally impractical.

Also, subsetting the data according to a particular building and meter type results in loss of metadata features as all the values of these features like 'square feet area' will be the same for a particular building. And these features wouldn't add any variation to the model, so technically the model will not be able to learn anything from these features. Thus, our dataset can be better modeled in the future by considering it as a regression problem instead of a time-series problem.

Results:
Table 2 indicates the RMSE and MAE values obtained by us for the different models-

Sr.No.	Model	RMSE	MAE
1	LGBM (Light Gradient Boosting Method)	1.214	0.552
2	Linear Regression	3.386	1.333
3	Decision Tree Regressor	0.798	0.301
4	LSTM (Long Short-term Memory)	24.8	4.66

Table 2. Model metrics

Conclusion:
Our end goal was to predict the future energy consumption pattern using one year of historic usage data with a minimum error rate. We tried to model this problem using time series forecasting methods, however, due to the hierarchical nature of the dataset, the application of time series modeling seemed to be computationally impractical and inefficient. Thus, after researching more on similar datasets, we found out that our results will be more accurate if we treat it as a regression problem rather than a time series problem that is verified by the RMSE values obtained. Amongst the classical regression models, tree-based models such as LGBM and Decision Tree performed much better than other models.

Statement of contributions:
Saurabh Parkar - EDA and Feature Engineering, LSTM Model
Vaibhav Saraf - EDA and Feature Engineering, Light Gradient Boosting Model
Sanjan Vijayakumar - EDA and Feature Engineering, Decision Tree and Linear Regression
Kyu Ha Kim - EDA and Feature Engineering

References:

[1]https://papers.nips.cc/paper/6907-lightgbm-a-highly-efficient-gradient-boosting-decision-tree.pdf

[2]https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/

[3]https://medium.com/@pushkarmandot/https-medium-com-pushkarmandot-what-is-lightgbm-how-to-implement-it-how-to-fine-tune-the-parameters-60347819b7fc

[4]https://towardsdatascience.com/understanding-gradient-boosting-machines-9be756fe76ab

[5]https://www.geeksforgeeks.org/python-decision-tree-regression-using-sklearn/