Pouria Faridi June, 2022

This file describes the methods used in analysing the given dataset and training the models and the reasons behind different selections. These are also briefly discussed in the provided Jupyter notebook.

Data Analysis and preprocessing:

First of all, the data frame was moved to the date-time format so useful time-relevant features could be extracted. Rows were indexed according to their date-time label and were sorted in an ascending format, from 2010-01-01 00:00:00 to the end.

Dataset was checked for any missing, Nan or duplicate values. There were no Nan values, but 14 hours were missing throughout the entire dataset. Those were left as they were since this can happen in the future test set as well. The duplicate values were averaged to represent only one energy value for an hour of a unique date.

Energy values for each unique day were summed as the goal was to predict the energy consumption for a day in future, as apposed to every hour. The last row was then deleted from the dataset since it was representing only the first hour of a new day, with 23 hours remaining.

Test data (last 20% of the dataset) was then separated from the training data before any observations for feature selection to prevent potential bias. Note that final feature extraction is performed on the overall dataset but only training data should be used for observations and feature selection.

The general plot of the training set indicated that there were some outliers present. However, trying to remove the outliers or filter the noises will lead to the assumption that the test dataset is without any noises/outliers. Therefore, the dataset was not manipulated.

Possible ways to remove the noise/outliers:

- Removing noise by moving the dataset from time domain to the frequency domain. Then filter out the undesired frequencies. This can be done by using a Fourier transform.
- Removing outliers that are n=3 standard deviations above or below the mean or out of the interquartile range.

Observations:

The training data was then used for plotting different distributions, grouped by year, month, week and day of the month. The boxplots showed that “weekends” can be distinguished from “weekdays” as they tend to have lower energy consumption. Also consumption in different “months” were showing variations. However, not a noticeable difference was seen from the “year” and “days of the month” distributions, indicating that they might not be good features for use. Also the auto and partial correlation plots showed that there was a correlation between the current value and the values of 1 and 2 previous days. Also they may be an indirect correlation with 90,180,270 and 365 days before.

Feature extraction:

In addition to the features discussed above, lag and window features are important when dealing with time series datasets. As a result, different features based on the observations mentioned above were extracted such as:

- Median and interquartile range of the consumption for the last 5 days
- Median and interquartile range of the consumption for the last week

- Consumption at 1,2,7,90,180,270 previous days
- Consumption at the last 1 year (the same date)
- 13 features were used in total (month of the year, is_weekend and the ones in the bullet points)

Note: According to the Shapiro-Wilk test, dataset was not following a normal distribution and as a result, median and interquartile range was preferred as apposed to the mean and std. Moreover, other features (other lag values) were also examined but these features seemed to provide better prediction performance. These features were then standardized.

Baseline model:

There were few choices for a baseline model. The chosen strategy here introduced the value of the day before ("1D_before" feature) as the prediction for the current day since it was mostly correlated with the power consumption value. Testing data with the baseline model showed a RMSE of 59,946 where the energy consumptions were in the order ~800,000 MWH. Also the r2 score was 0.68 which is acceptable for a baseline model. Other baselines with taking the average/median of the n previous days were also tested but as expected, the considered model had the best performance among others, due to the higher correlation of "1day" feature to the power consumption.

Parametric models:

2 simple parametric models were trained and tested: a linear regressor and a neural network with 2 layers and a relu activation in between.

Both models were trained using the 13 features of the training dataset and tested afterwards. Their performance in predicting both the training and testing datasets was slightly better than the baseline model: for training: r2 score=0.84 and RMSE=47,000. For testing: r2 score=0.78 and RMSE=49,000.

Models' limitation: These models required manual feature extraction. There may be other useful features that are not included in this analysis and so some information may be lost with manual extraction. This is the problem of conventional machine learning algorithms.

Models' assumption: For using models that require feature extraction, we need to assume that our features are known, for both training and testing sets. Since we are using lag/window features which are dependent on the previous days' targets, we need to assume that for each test data, we have the actual target of the previous days. Using predicted targets as features for the next prediction will increase the error day by day and will result in a failure.

Overall, higher level features are required to capture the long term dependencies and provide better prediction performance.

Discussing a more advanced model: **LSTM networks**

In the advanced deep learning models, feature extraction/engineering is done automatically through the deep layers, starting from low-level features and moving towards more complex ones. This eliminates the need for manually selecting features and losing valuable information.

Moreover, using recurrent deep-learned models allow us to capture the time dependencies between different days and use one prediction to provide feedback for calculating the next prediction in the batch. In recurrent networks, outputs of hidden layers will serve as an additional input (known as memory) to the network during the next training step.

Long short-term memory (LSTM) networks are therefore the ideal candidate for predicting time-dependent labels. They are specialized type of recurrent neural networks (RNN) that learn to predict future trends from sequences of variable lengths and eliminate the vanishing gradient problem that may be present by using a simple RNN. An LSTM network with 2 or 3 hidden layers should provide adequate prediction performance in this problem.

References:

Yildiz B, Bilbao JI and Sproul AB. A review and analysis of regression and machine learning models on commercial building electricity load forecasting. *Renew Sust Energ Rev* 2017; 73: 1104–1122 https://doi.org/10.1016/j.rser.2017.02.023

Lin M. Predicting energy demand with neural networks (2020). https://towardsdatascience.com/forecasting-energy-consumption-using-neural-networks-xgboost-2032b6e6f7e2