Machine-Learning Assisted Energy Benchmarking

Sammie Haughton, Grant Royster, and Praveen Wunnava

Abstract— For "pay for performance financing", building owners make payments for energy consumption based on actual vs. expected usage, incentivising energy conservation. Models can be trained on data consisting of weather data, building metadata, and meter readings, in order to provide accurate expectations for electricity, steam, chilled water, and hot water consumption. Several linear regression and neural network models were trained, and different data processing techniques were applied in an attempt to decrease the models' error rates.

Index Terms—energy, benchmarking, neural network, linear regression, machine learning

Link to Github Repository— https://github.com/praveen-wunn/COMP-562_Group/

Introduction

For both economic and environmental reasons, building owners are incentivized to minimize their energy usage by optimizing their hardware and resource consumption patterns. Under "pay for performance financing", building owners make payments based on the difference between their energy consumption and *expected* energy consumption [1]. This not only incentivizes changes to decrease consumption, but it provides a measure of effectiveness for those changes. However, in order to measure energy efficiency progress, it is important to have accurate estimates of what energy usage may have been, if behavior continued as is. In other words, this project aims to predict energy usage in buildings based on historical data, in order to provide a means of predicting future usage which could then be used as a benchmark to measure energy savings. Several models were trained and tested to accurately predict energy and resource consumption based on building and weather data.

1 Background

In the United States, most electricity is produced using fossil fuels including coal and natural gas [5]. Fossil fuels are also used to fuel district energy systems, which produce steam, hot water, and chilled water for groups of commercial buildings like college campuses [6]. Since energy production via natural gas produces greenhouse gasses and pollutes the air, minimizing energy consumption has a positive environmental impact.

In 2012, space-heating, lighting, refrigeration, ventilation, and computer and office equipment accounted for most energy consumption in commercial buildings. Building usage also heavily impacted energy consumption: for instance, mercantile buildings accounted for 15% of energy usage, while health care facilities accounted for 8% [6].

Due to the economic and environmental implications this problem presents, energy consumption prediction is a large area of research in the machine learning community. Zhao and Magouès conducted a review of past and present techniques that have been used, including complex methods involving "thermal dynamics" and comprehensive aspects of a buildings' environment, as well as more simplified methods that rely on specific components, such as building load or temperature [8]. Specific components of different algorithms cited included White and Reichmuth's [7] use of average monthly temperatures, and Barnaby and Spitler's method for calculating the "load" or heating/cooling requirements of a residential building [3].

2 Methods

A Kaggle dataset was used to train and test two different models using google collaboratory notebooks, tensorflow, and pandas libraries [1]. An outline of the process followed is: (1) combining, manipulating, and cleaning the data, (2) splitting the data into training and testing subsets, (3) training Linear Regression and Neural Network models on the training subset, (4) using these models to make predictions for the test data, and (5) measure the accuracy of the models using and comparing results.

2.1 Data

The main dataset consists of three main parts: (1) building energy consumption measurements, including the consumption amount and timestamps, (2) building metadata, including square footage and primary use of the building (e.g. Office), (3) supplemental weather information collected at different sites. There is data available for a total of 1449 scattered across 16 different weather sites. Meter readings and weather data were collected hourly for each site. The final merged dataset used for training consisted of over 13,000,000, with a test set containing over 6,000,000 rows.

Although a separate testing dataset was also available on Kaggle, the decision was made to just use data from a single year, due to the volume of data available and the fact that Kaggle does not make the labels for its test data available until predictions are submitted.

2.1.1 Data Processing

Right away it was apparent that there were several missing values, especially in the weather dataset. Columns that had no data available for an entire site were dropped, namely sea level pressure, cloud coverage, and precipitation depth. Rows were then sorted first by site ID, then time stamp, so that data was available in chronological order. Assuming the weather does not drastically change from hour to hour, a pretty good estimate for a missing timestamp would be the previous hour, future hour, or combination of the two. Therefore, the following steps were taken to fill in missing values for the remaining columns. For each unique weather site, rows with null values for a particular column were identified. For each of those rows, if available, the values for that column from the previous row and next row were averaged together. If one of those values was also null, only the non-null value was used. If both values are null, an average value for that site at that hour for the entire month was used.

Once the weather data was merged with the training data, timestamps that were completely missing from the weather dataset were identified. A similar process was used (i.e. incorporating previous and/or future weather data) to add new rows to the weather dataset. This time, however, the process had to be repeated a total of three times, due to the fact that in some cases there were several subsequent missing time values.

There were also several categorical variables in the dataset. To make the timestamp attribute more valuable, hour, weekday and month values were extracted. These attributes, along with the building's primary use and meter type (e.g. chilled water) were transformed into one-hot-encoded variables.

Finally, lag variables for a few of the weather variables were included by selecting the values from a previous or several previous timestamps. This was done under the assumption that energy consumption of a building might take a little longer to respond to changes in the environment so a past temperature might be more valuable for the estimate than the current temperature.

2.2 Linear Regression

A linear regression model was built using the Tensorflow Keras machine learning API for Python. In order to understand the features better, we created a heat map of their correlations, shown below in Figure 1.

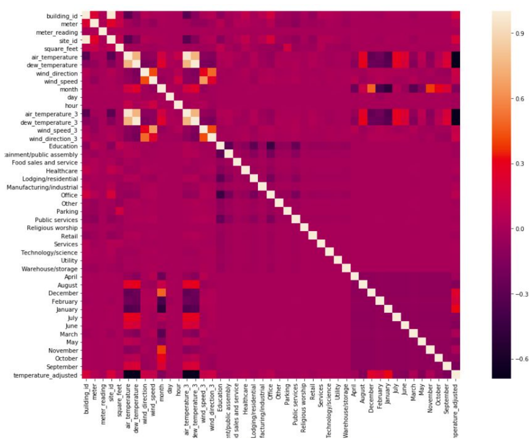

Figure 1. Training Feature Correlations

While none of the variables appeared noticeably correlated with the target variable ("meter_reading"), some of the attributes were correlated with each other. In order to mitigate potential performance hindrances on the model, if two features were highly correlated, only one was selected [2]. For building the actual model, a tutorial on training regression models using tensorflow was followed, with a few adjustments [4]. A linear activation function was selected, along with the RMSprop optimizer.

The first attempt at training the model occurred prior to categorical variables being one-hot encoded. The final set of attributes used included air temperature, air temperature with a lag of three hours, wind speed, building square footage, month, hour, and meter. Upon initializing the fit function, it was quickly determined that the dataset was much too large to invoke the original 1000 epochs recommended by the tutorial for training. The model was reset and trained using eight epochs, which took about 140 minutes to train.

Once the adjustments to the categorical data had been invoked, a second attempt at training a linear regression model took place. Some challenges with normalizing the much larger dataset (39 attributes, instead of 9) led to the decision to train using unnormalized data instead in order to save RAM and computation time. 12 epochs were set to train the model, and it took about 175 minutes to train.

After the second attempt's performance only marginally improved, it was decided to split the data into four separate sections—one for each meter type. Since the meter reading values varied so greatly between electricity, chilled water, steam and hot water, the goal was to specialize each model to more accurately predict within each type's range. This also helped improve runtime performance of the model, with each taking about 60 minutes to complete 15 epochs.

2.3 Neural Network

A neural network model was built using the Tensorflow Keras machine learning API for Python. The idea was to check its performance in contrast to the baseline regression model. In order to keep a level playing field, the training and test dataset used were exactly the same. Thus, the model had exactly the same input variables and the same split of unnormalized data. Like linear regression, multiple iterations of this model had to be run due to infrastructure constraints. Finally, 10 epochs were set to train the model which took around 60 minutes to train.

3 Results

The performance of the first attempt at a linear regression model was very poor, with almost all of the values predicted lying somewhere around 50kWh. This is likely due to categorical variables, such as month and hour stilling being included as numerical values, as well as the fact that training and optimization only occurred during eight passes through the data. However, valuable insights were still drawn, as led to a better understanding of the data and how to better process the attributes to improve model performance.

The second model performed marginally better, in the sense that the prediction values varied some. However, the final mean absolute error, mean square error, and loss values were still very high. It is interesting to note in Figure 2, that the model predicted instances of negative amounts of energy used, even though there were no instances of values less than zero in the training or test labels. This could be due to a relatively large number of zero readings in the training dataset (~9%). Additionally, the range of energy consumption values was quite large (0kWh to over 21,000,000 kWh). Perhaps with more training the model would have been able to learn these more extreme behavior, or maybe outlier detection would have been helpful. There are some values towards the top right corner of the Figure 2, around the 3,500 kWh range. However, the maximum amount the model predicted was only about 5,000 kWh.

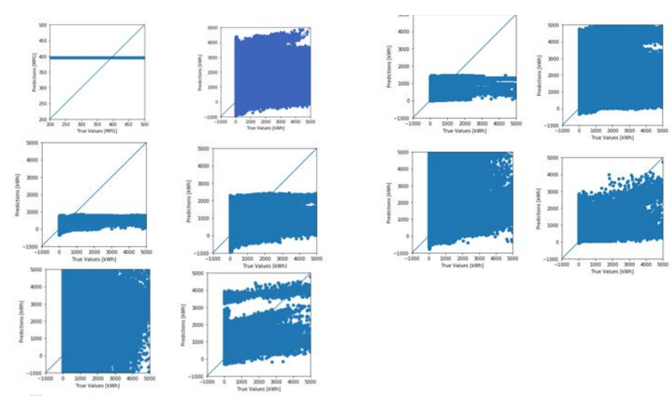

Figure 2. Linear Regression Attempt 2 outputs vs. Test Labels
[Top: Linear (L)1, L2, Neural Network (NN) Electricity (E), NNCW
Middle: LE, LCW, NNCW, NNHW, NNS
Bottom: LHW, LS]

The predictions seemed to moderately improve with the four separate models for each meter type. The error metrics decreased by orders of magnitude for all four categories, and the mean absolute error score for the electricity linear regression predictions was the lowest of all the models.

Model	Meter Type	Mean Absolute Error	Mean Squared	Loss
Linear Regression 1st Attempt	All	2.22E+03	2.35E+10	2.35E+10
Linear Regression 2nd Attempt	All	2.82E+03	2.94E+10	2.94E+10
Linear Regression 3rd Attempt	Electricity	1.25E+02	1.17E+05	1.17E+05
Linear Regression 3rd Attempt	Chilled Water	6.37E+02	6.31E+07	6.31E+07
Linear Regression 3rd Attempt	Steam	2.47E+04	2.19E+11	2.19E+11
Linear Regression 3rd Attempt	Hotwater	6.11E+02	8.80E+06	8.80E+06
Neural Network (no validation set)	Electricity	9.97E+04	9.97E+04	9.97E+04
Neural Network (no validation set)	Chilled Water	6.32E+07	6.32E+07	6.32E+07
Neural Network (no validation set)	Steam	1.75E+11	1.75E+11	1.75E+11
Neural Network (no validation set)	Hotwater	5.99E+06	5.99E+06	5.99E+06

Table 1. Error Results for all Models

Interestingly, the neural network mean squared error and loss rates tended to be lower for the neural network models than those of the linear regression model. Additionally, the prediction-true plots, shown in Figure 2 illustrate that many of the models had trouble predicting higher values. This could be due to a greater number of zero meter readings, which could potentially be considered errors. Removing these samples and training the models again could potentially help improve performance. The models where a slight upward slope starts to appear are the second attempted linear regression model, and both hot water models.

4 Conclusion

The third LR attempt with the reduced dataset gave much better results indicating that each iteration gives better insight into the data. The one hidden-layer neural network model didn't perform well. The idea was to increase the number of hidden layers or increase the epochs, but the system kept running out of memory and space. Overall, apart from the experience gained on how to build models, the main lessons learned were overcoming challenges faced during the set-up of data. As discussed in class, this is true most of the time. More on challenges faced are outlined in the next section.

From an industry point of view, counterfactuals models could help in estimating the 'actual' energy savings of a building. But, since these models are based on weather and location, it is highly likely that these could be utilized/transferred for a different purpose.

4.1 Challenges

One of the biggest challenges with this project was determining how to best fill in the missing data. Several different strategies for filling in the missing weather data were considered, including averaging values across different amounts of time, excluding attributes all together, deleting rows with null values, or using data from the closest timestamps available. Particularly challenging values to fill in were those from sites that were missing particular values from an entire hour for the entire month, and whether or not averaging values from other months and/or other sites would still lead to accurate estimates or be useful.

The large size of the dataset selected led to challenges when it came to filling in the missing values, as well as other aspects of the model building process. The google collaboratory jupyter notebook made it easy to share code and ensure all the components necessary to use Tensorflow were properly installed, however often during runtime, the RAM would run out and the system would crash and restart. In order to account for this, new training and test files were exported into csv files so that not all code would need to run at once. However, once the new features were added, normalizing the dataset still proved to be too computationally heavy, and new files were unable to be created. The large amount of data also led to long runtimes for fitting the model, which made it challenging to modify parameters and limited the number of different model attempts that occurred.

Despite these challenges, the experience of finding, cleaning, and training a model with a large dataset was a valuable experience, as the problems we encountered are likely reflective of real-world machine learning challenges.

4.2 Extensions

One way to extend this project that may help improve performance for predicting energy consumption in buildings would be to apply different machine-learning algorithms and see how they compare to the neural network and linear regression model. For example, a Bayesian network may be able to better account for variable dependence (e.g. missing values that were filled in using other samples). A polynomial regression may also be a good option, in order to better capture the relationship between, for example, the air temperature and a building's heating and cooling system.

Other methods for filling in missing values could be applied, particularly for the attributes that were removed due to sparsity. For example, in order to fill in the values for the year a building was built, which may indicate whether or not energy efficient appliances are installed and could therefore affect the consumption in that building, one estimate for missing values could be the most common year other buildings were built (i.e. the mode). Also, outliers could be identified and potentially removed.

In general, our dataset included a large number of features and samples, so compressing the data may lead to runtime improvements, which would allow for the normalization of the dataset, which could lead to improved performance. On the other hand, adding more samples, or "noise" to the dataset may help to reduce the likelihood of overfitting, which could lead to performance improvements as well. Other types of data, such as the geographic region, or water consumption and occupancy rates of different buildings could also be included, since they may impact how much energy is being used in a building at any given point in time. Finally, allowing the models to train for longer periods of time, may lead to improved performance as they would better be able to optimize the necessary parameters.

References

[1] ASHRAE - Great Energy Predictor III. (n.d.). Retrieved December 8, 2019, from https://www.kaggle.com/c/ashrae-energy-prediction.

[2] Badr, W. (2019, January 22). Why Feature Correlation Matters .... A Lot! Retrieved from https://towardsdatascience.com/why-feature-correlation-matters-a-lot-847e8ba439c4.

[3] Barnaby, C. S., & Spitler, J. D. (2005). Development of the Residential Load Factor Method for Heating and Cooling Load Calculations. *ASHRAE Transactions*, *111*(1).

[4] Basic regression: Predict fuel efficiency : TensorFlow Core. (n.d.). Retrieved from https://www.tensorflow.org/tutorials/keras/regression.

[5] Environmental Protection Agency (EPA) - Learn about Energy and Its Impact on the Environment. (n.d.). Retrieved December 8, 2019, from https://www.epa.gov/energy/learn-about-energy-and-its-impact-environment

[6] U.S. Energy Administration (September 2018) - Use of energy explained: Energy use in commercial buildings . Retrieved from https://www.eia.gov/energyexplained/use-of-energy/commercial-buildings.php

[7] White, J. A., & Reichmuth, R. (1996, August). Simplified method for predicting building energy consumption using average monthly temperatures. In *IECEC 96. Proceedings of the 31st Intersociety Energy Conversion Engineering Conference* (Vol. 3, pp. 1834-1839). IEEE.

[8] Zhao, H. X., & Magoulès, F. (2012). A review on the prediction of building energy consumption. *Renewable and Sustainable Energy Reviews*, *16*(6), 3586-3592.