Great Energy Predictor III :Predicting Energy Consumption of Buildings

Abdulsalam Yazid Bande

Table of contents

Project definition

This project involves predicting the amount of energy that a building consumes. These building range from skyscrapers to school buildings

APPLICATION AREA

- Residential Areas
- Offices
- School

Methods

METHOD 1 (Neural Network)

- Since the dataset is a huge dataset containing more than twenty Million instance of training set. A reasonable model to apply is a Neural Network(NN) with as LSTM architecture. This is because NN statistically shine and generalize well on a big dataset. The problem with this technique is that it is computationally expensive especially on dataset such as the one in this project. Another problem is that hyper parameter tuning is hard as training (which takes on average 5 hours) has to finish to make some adjustment in the model. But NN got the second best result

Methods

METHOD 2 (Xgboost)

- Another way to model the problem is to use Xgboost. Xgboost can have a lot of regressors at the same time and the final result is based on the average of those classifiers. Xgboost got a better result in this project than NN because the high training time of the NN did not allow for hyper-parameter tuning. In fact even the Xgboost model itself took time to finish training and tune the hyperparameters, so a model that takes less time and memory is necessary.

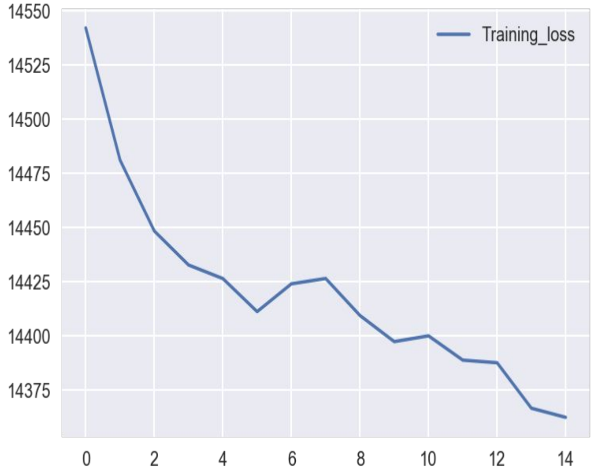
Methods

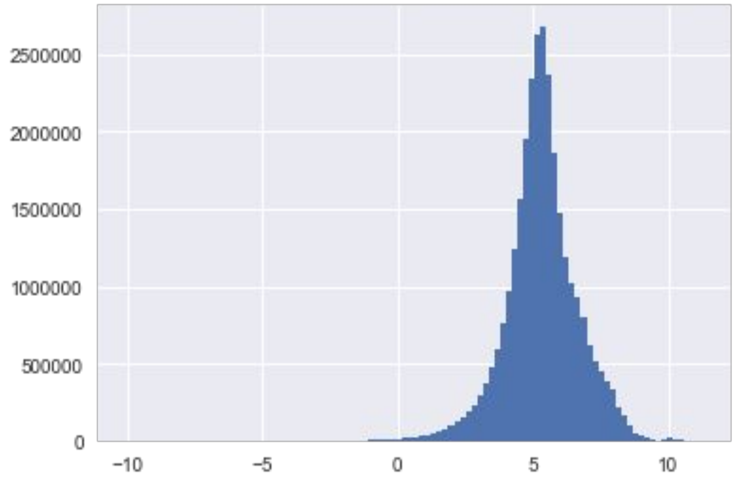
METHOD 3 (Random Forest)

- Random Forest is an ensemble of Decision Trees. It is trained using a bagging method.
- Random Forest introduces randomness when growing trees because it searches for the best features among a random subset of features which leads to a good tree diversity which leads to the trading of high bias for a lower variance.

Results(Neural Network)

Neural Network: The Neural network got the second results out of all the models. This is because the training time was too much for any hyper-parameter tuning, In fact grid search could have been used. It also did not suffer from skewness

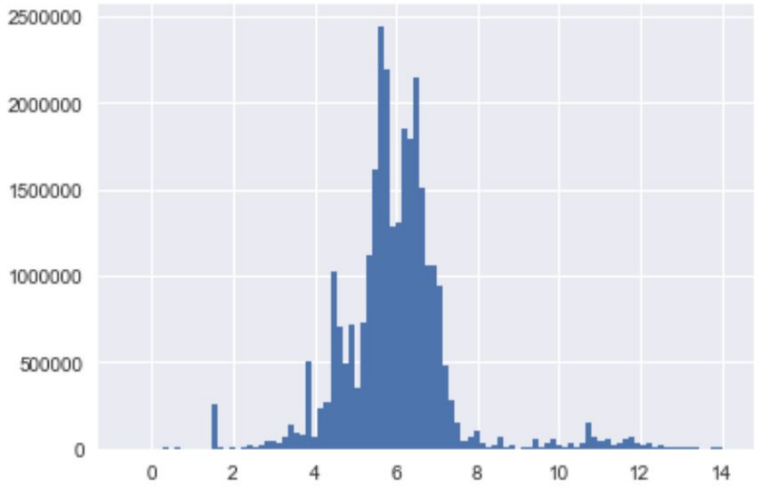




- Left Image shows the loss against the epochs, the loss is MSE, the Y axis is the epoch number while the x axis the the loss. The loss is big because log was not taken.
- The right Image shows the histogram of the log of the predictions of the test set. The result is centered around 5 meter unit reading which is quite accurate. Neural Networks can certainly perform better but computation limitation just wouldn't allow it

Results(Xgboost)

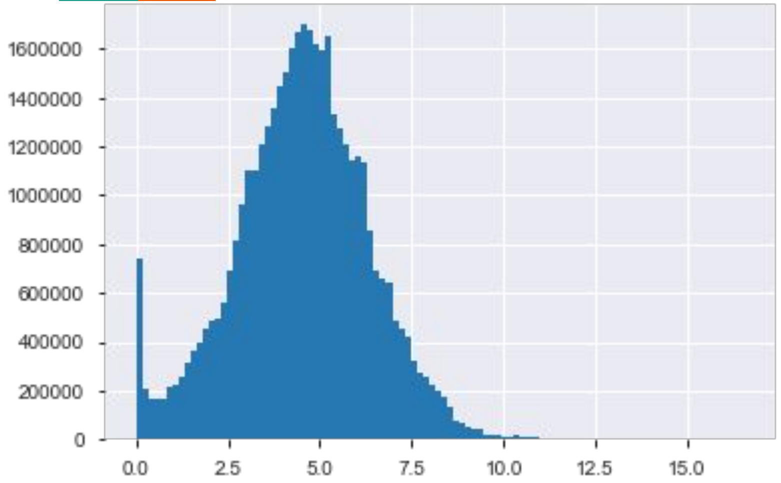
The Xgboost model got a better worst result, but it suffered from left and right skewness



- The Image above shows the histogram of the Log of the prediction of the testing dataset. The data is more centered around 6 meter reading and a left skewness dominates. Furthermore there are a lot of outliers in the range of 10-12 meter reading. The Xgboost took an average of 1 hour 50 minutes to run.

Results(Random Forest)

This model got almost 50% more accurate than Xgboost has it did not suffer from bad skewness .



- The Image above shows the histogram of the Log of the prediction of the testing dataset. The data is more centered around 4.9 meter reading and no any skewness or outliers. Also this model was 21 times faster than Xgboost almost 50 times faster than the NN. This is the model that is submitted on kaggle

Implementation tools,paper and source

TOOLS MAIN TOOLS

- Python
- Pytorch(Used in Neural Network Training)
- Numpy
- Scikit Learn(Used in Random Forest)
- Xgboost (Used in Xgboost)
- Matplotlib

The Computer is a 2012 Macbook Pro 2.3 GHz Dual-Core Intel core i5

Conclusion

Neural Networks can be powerful but their computation requirement limit hyper-parameter tuning. Also Scikit Learn is a very powerful library.