

Prediction and Analysis of Household Energy Consumption by Machine Learning Algorithms in Energy Management

M.Rambabu¹, N.S.S.Ramakrishna^{1,*}, P.Kumar Polamarasetty¹

¹ Department of Electrical & Electronics Engineering, GMR Institute of Technology, Rajam, India

Abstract. Now the world is becoming more sophisticated and networked, and a massive amount of data is being generated daily. For energy management in residential and commercial properties, it is essential to know how much energy each appliance uses. The forecast would be more clear and practical if the task is based purely on energy usage data. But in the real world, it's not the case, energy consumption is strongly dependent on weather and surroundings also. In a home appliances network when measured/observed data is available then algorithms of supervised-based machine learning provide an immeasurable alternative to the annoyance associated with many engineering and data mining methodologies. The patterns of household energy consumption are changing based on temperature, humidity, hour of the day, etc. For predicting household energy consumption feature engineering is performed, and models are trained by using different machine learning algorithms such as Linear Regression, Lasso Regression, Random Forest, Extra Tree Regressor, XG Boost, etc.. To evaluate the models R square is used as the forecasting is based on time. R square tells how much percentage of variance in the dependent variable can be predicted. Finally, it is suggested that tree-based models are giving best results.

1 Introduction

In recent decades, studying energy consumption issues has been a popular academic area. Energy issues are critical to society's security and well-being [1]. Predicting energy consumption in industry and the energy sector is an essential part of macro-planning in the industry sector, according to economic theories. [2]. Energy supply-demand planning on a long-term basis must meet the needs of countries' long-term development. Making accurate estimates about future energy usage can help decision-makers plan and schedule supply system operations more effectively. Forecasting the energy consumption load is a key aspect of power distribution systems' economic and safe operations planning. According to the forecasting horizon, there are primarily three types of energy consumption studies in the literature [3]. Resource management and development spending account for the majority of long-term forecasts (5–20 years). The distribution network's scheduling and analysis are frequently aided by short-term forecasting (for periods ranging from an hour to a week), where as mid-term forecasting (a month to 5 years) is mostly used for planning power production resources and rates [4][5]. The focus of the paper is on short-term forecasting. Because energy demand is affected by a variety of factors such as time, climate, socio economic, and demographic factors, precisely anticipating consumption is both crucial and difficult.

Regression models, as well as ANNs and time-series forecasting, are the most widely utilized methods. The focus of this study is primarily on regression models [6][7]. This study uses regression models such as linear regression, lasso regression, ridge regression, random forest, XGBoost, and Extra tree regressor [8].

The dataset contains measurements of house temperatures and humidity recorded at 10-minute intervals over 4.5 months using a ZigBee Wireless sensor network; the entire experiment is based on this dataset. Our experiment is anticipating the following instances energy usage based on the previous data and providing it to the power plant for generating reasons [3]. It would be simple to anticipate energy use if energy consumption was just based on the appliances in the home. However, the amount of energy used in the home is often determined by the temperature and humidity of the day [7]. As a result, to create good predictions with the data given, these experiments are carried out to find the optimal model with the best attributes. The basic goal is to anticipate the energy consumption of a future instance using current and historical data on temperatures, humidity, and other environmental variables.

The second section of the paper is dedicated to data collection and analysis of study area. The research provides answers to inquiries about how energy consumption varies hourly, day by day, and month by

* Corresponding author: namkrishna231@gmail.com

month. The third section would concentrate on feature engineering and selection. The fourth section would involve modeling and comparing many models to choose the best one. The dataset is good enough and only requires scaling for data processing [9].

2 Study Area and Data Analysis

The average temperature outside is about 7.5 degrees centigrade. The temperature fluctuates from -6 to 28 degrees centigrade. Inside the building, the average temperature has been approximately 20 degrees centigrade in all rooms. It has a temperature range of 14 to 30 degrees centigrade. Warming appliances have been put in place to keep the insides of the building warm, according to this. As a result, there would be a link between indoor temperature and energy use. Outside the building, the average humidity has been higher than within. The average humidity in the children's and parent's rooms is significantly higher, indicating that the residents of this building spend most of their time in these areas. Because the humidity levels are so high, some appliances, such as dehumidifiers, may be necessary. As a result, there would be a link between humidity and energy use.

There are two instances of peak hour activity. The first is at 11 a.m., and the second is at 6 p.m. The peak at 11 a.m. is shallow and low, whereas the peak at 6 p.m. is taller and sharper. The energy usage of appliances is roughly 50Wh during the sleeping hours (10 PM - 6 AM). After 6 a.m., energy use gradually increases until 11 a.m. (probably due to morning chores). At roughly 3 p.m., it progressively drops to around 100 Wh. After that, until 6 p.m., energy use skyrockets (probably due to requirement lights in rooms). However, as night falls and everybody in the house retires to bed around 10 p.m., appliance energy consumption drops to 50 Wh. In Belgium, it appears that the month has no impact on energy use. However, due to the warm weather, may month's energy use may be lower. Energy trend over months for every hour are shown in figure 1.

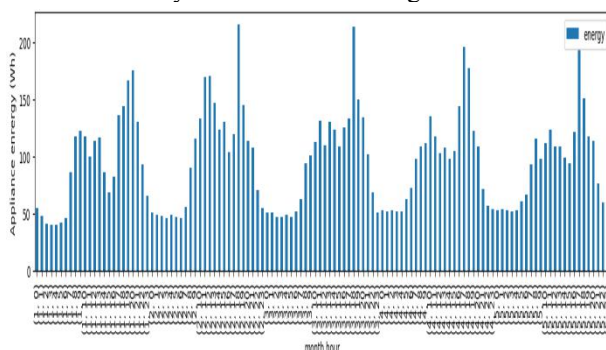


Fig. 1. Energy trend over months for every hour in Belgium
 Weekend energy use is higher than weekday energy usage.

This could be because weekends are holidays, and most of the building residents stay at home and use appliances. Hourly energy trend on weekdays and weekends are depicted in figure 2.

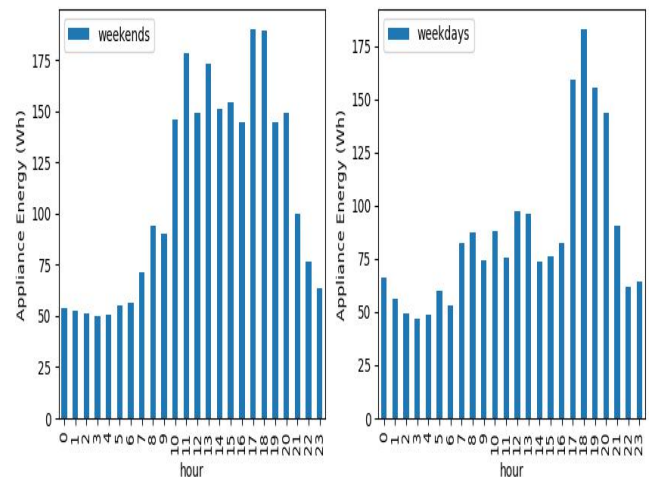


Fig. 2. Hourly energy trend on weekdays and weekends in Belgium

The data does not follow a normal distribution. Because the data is skewed, log10 is used. Because many models have the underlying assumption of normal distribution, this type of transformation has a positive impact on modeling.

The Pearson correlation is used to determine how interdependent the features are. It has been discovered that characteristics have a range of -0.09 to 0.02 with energy, which is lower than expected. This necessitates a concentration on feature engineering to obtain a greater number of linked features with the dependent feature for modelling purposes. Energy distribution before and after log transformation respectively are shown in figure 3.

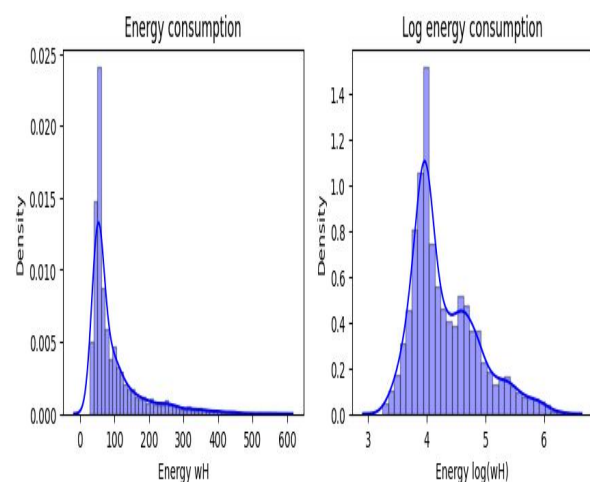


Fig. 3. Energy distribution before and after log transformation respectively in Belgium

3 Feature Engineering

According to the above analysis, we discuss about feature engineering. Different significant features are retrieved from the features provided in this section. Because the data is skewed, a new feature named log energy is added to the energy feature by performing the log10 transformation to the energy feature. Temperature and humidity features are highly reliant on their own

features, thus by averaging them, a new feature is created from one of the temperature features and one from the humidity features. Temperature and humidity are also inversely connected, according to the data analysis. This reason can be used to multiply each of the temperature and humidity features of the same room, yielding an additional 9 features. Apart from features like temperature and humidity levels in rooms and outside, there is another crucial column date, which contains timestamps for each of the sensor data samples. The hour data for each sample is derived from the timestamps, and the goal is to see if there is a pattern of energy usage over a day. For 4.5 months, the figure 4 below depicts the average energy usage of appliances at a given hour of the day.

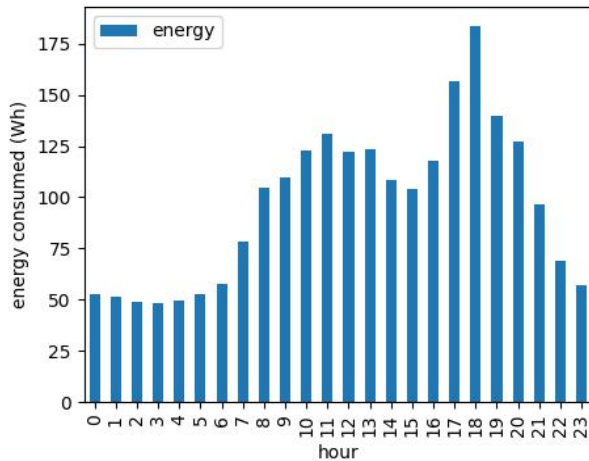


Fig. 4. Average hourly energy consumption trend

Hours and their relationship with energy consumption were explored in the analysis section. The 24 hour day can be divided into three classes based on energy consumption patterns: Class 1: 10 PM - 6 AM, Class 2: 6 AM - 3 PM, and Class 3: 3 PM - 10 PM so that the model can learn from multi-class classifier [10].

An additional 13 features are retrieved by feature engineering, and observed characteristics are now more dependent on energy use, with correlations ranging from -0.21 to 0.55.

4 Modelling

Now having good features, The next stage is to model using different regression models and compare them to find the optimum regression model for energy prediction. Because this is a regression problem, the metric utilized will be R2 (R squared), which gives a measure of the variance of the target to a variable that can be explained using the given characteristics. It is a measure of how well the model fits the data in a practical sense. It can be expressed numerically as equation (1):

The percentage of R² score is

$$R^2 = (1 - \frac{SS_{Regression}}{SS_{Total}}) * 100 \quad (1)$$

where,

$SS_{Regression}$ = Total Residual Sum of Squares computed from the best fit line

SS_{Total} = Total residual sum of squares measured from the mean

Using the 'r2_score()' function of the metrics module of the scikit-learn library.

4.1 Linear Models

Linear Regression is the most basic Regression algorithm. There is no need for further complexity if a Linear model can adequately describe the facts[11]. We can use Regularization approaches to penalize the coefficient values of the features as a modification to the original Least Squares Regression because higher values contribute to over fitting and loss of generalization. Linear models benefit tremendously from regularization approaches. In addition, there are only a few circumstances in which a Linear model may fit the data effectively without Regularization. Regularization transforms the problem of Linear Regression into Lasso or Ridge Regression, depending on whether we add the absolute values of coefficients or their squares to our loss function. The R2 score of linear regression without feature engineering, which is used as a baseline model, is 17%.

4.1.1 Linear Regression :

The bias term is added to the weighted sum of the input features in a linear model to produce a forecast in equation (2) (also called the intercept term)[12]

Model prediction

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n \quad (2)$$

\hat{y} = predicted value.

n = number of features.

x_i = i th feature value.

θ_j = j th model parameter (including the bias term θ_0 and the feature weights $\theta_1, \theta_2, \dots, \theta_n$)

Setting the parameters of a model so that it best fits the training set is referred to as training it. To do so, we'll need a metric to determine how well (or poorly) the model matches the training data.

Since it may be used to a wide range of problems, Gradient Descent is an extremely versatile optimization technique. Gradient Descent's general concept is to iteratively change parameters to minimize additional cost functions. The learning rate hyper parameter determines the size of the steps, which is an important parameter in Gradient Descent [12].

Regularization of a linear model is often accomplished by restricting the model's weights. Ridge Regression, Lasso Regression, and which implement alternative techniques to restrict the weights.

4.1.2. Ridge Regression

Ridge Regression (also known as Tikhonov regularization) is a regularized variant of Linear Regression in which the cost function is given a regularization term of equation (3)

$$\alpha \sum_{i=1}^n \theta_i^2 \quad (3)$$

Learning algorithms are compelled to adjust to the data and keep weights as low as feasible when this is enforced. Remember that the cost function should only be subjected to regularisation when it is being trained. To evaluate the model's performance using the unregularized performance metric once it's been trained from equation (4).

$$F(\theta) = \text{MSE}(\theta) + \alpha \frac{1}{2} \sum_{i=1}^n \theta_i^2 \quad (4)$$

The type of regularization term to utilize is determined by the penalty hyper parameter. When you choose "l2," SGD will add a regularization term to the cost function equal to half the square of the l2. Ridge Regression is simply the norm of the weight vector [12][13].

4.1.3. Lasso Regression

Least Absolute Shrinkage and Selection Operator Regression (also known as Lasso Regression) is another regularized variant of Linear Regression: it adds a regularization factor to the cost function, same as Ridge Regression, but it employs the l1 norm instead of half the square of the l2 norm, uses the weight vector. The fact that Lasso Regression tries to fully exclude the weights of the least significant variables is a key aspect from equation (5) (i.e., set them to zero) [13].

$$F(\theta) = \text{MSE}(\theta) + \alpha \sum_{i=1}^n \theta_i \quad (5)$$

4.2 Models based on trees

Tree-based Regression models are the next type of method [12]. When compared to Linear models, Tree-based models have a significant advantage in terms of robustness against outliers. Because no linear link has been found between any attribute and the target variable, regression trees are likely to outperform linear models.

Given a large number of characteristics, a Decision Tree will almost certainly overfit the data. As a result, it could be skipped and instead go straight to the ensemble methods listed below, which include building multiple regressors on copies of the same training data and combining their output either through mean, median, mode (Bagging), or growing trees sequentially (i.e. each tree is built from the data of the previous tree) and using a weighted average of these weak learners (a learner that performs just as well as the strongest learner) [14].

Random Forests is a common Bagging technique that works well with high-dimensional data like ours. Extra Trees Regressor takes it a step further by randomizing splits. A form of boosting approach is Gradient Boosting Machines. It creates an additive model in which performance is continually improving[13].

4.2.1 Random Forests

Using the bagging method (or occasionally pasting), Random Forests are an ensemble of Decision Trees with maximum samples set to the training set's size.[15][16]. Instead of searching for the best feature, the Random Forest approach introduces additional randomization by

searching for the best feature among a random collection of features. As a result, there is a greater variety of trees, which reduces the variance while increasing the bias.

4.2.2 Extra-Trees

When building a tree in a Random Forest, only a random subset of the characteristics are examined for splitting at each node (as discussed earlier). Instead of searching for the best possible thresholds, it is feasible to make trees even more random by employing random thresholds for each feature (like regular Decision Trees do). Extremely Randomized is the name given to a forest made up of such extremely random trees (or Extra-Trees for short) [17]. Once again, higher bias is exchanged for reduced variance. It also makes Extra-Trees significantly faster to train than standard Random Forests, because one of the most time-consuming jobs of tree growth is determining the best feasible threshold for each feature at each node. The ExtraTreesRegressor class in Scikit-Learn can be used to generate an Extra-Trees classifier. It has the same API as the RandomForestRegressor class.

4.2.3 Gradient Boosting

Gradient Boosting is another prominent boosting strategy. Gradient Boosting, like Ada Boost, operates by successively adding predictors to an ensemble, with each one correcting the one before it. Unlike Ada Boost, this method seeks to adapt the new predictor to the residual errors created by the prior predictor rather than modifying the instance weights at each iteration. The paper [18] has clearly shown the mathematical procedure behind gradient boosting. A subsample hyper parameter is also supported by the Gradient Boosting Regressor class, which sets the fraction of training instances to be used for each tree's training. If subsample=0.25, for example, each tree is trained on 25% of the training cases, chosen at random. As you would have guessed, a bigger bias is exchanged for lower variance. It also makes training much more efficient [19][20].

4.2.4 XG Boost

XGBoost is a machine learning method that has recently won Kaggle contests for structured or tabular data. XGBoost is a high-speed and high-performance implementation of gradient boosted decision trees.

Along with these models neural networks can also be used. With the help of PCA dimensional reduction can be done. [22] address PCA to select the transformed features that can be more related to the target variable. The discipline of tweaking the hyper parameters of these algorithms to achieve optimal performance is known as hyper parameter optimization. One of the most obvious methods to perform is manually selecting hyper parameters but it is not feasible as we cannot select values that would help in the optimization. Because there are so many choices and options, this technique does not scale [23]. Many strategies for hyper parameter optimization have been proposed. Grid search, Random

search, and Bayesian optimization are three well-known strategies .

All of the models are modeled using the steps below:

The final set of features used are really important. These features are solely responsible for the better performance of the model [24[25]]. With the help of a correlation plot, these features are chosen. These are the final set of features taken into consideration for modeling.

final_features = ['th_kitchen', 'th_living', 'th_laundry', 'th_office', 'session', 'th_teen', 'th_parents', 'press_mm_hg', 'windspeed', 'avg_house_temp', 'hour', 'avg_house_hum', 'temp_outside', 'humid_outside', 'weekday', 'month', 'log_energy']. Here for the train-test split 80-20 is used. Steps for modeling of ML algorithm are shown in figure 5.

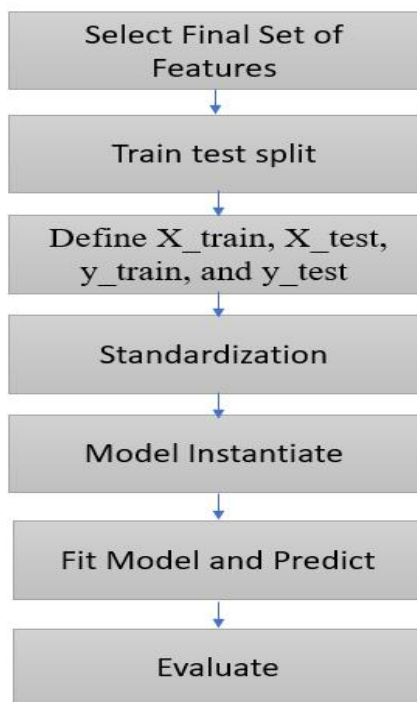


Fig. 5. Flow chart for modeling of ML algorithm

5 Result Analysis

5.1 Case1: Results for Baseline model

Table-1 shows the baseline model results. For the baseline model, the selected models are linear models and the modelling is done before feature engineering with the raw data. When the modelling is done with raw data only 17% of variance is only predicted.

Table 1. R2 score of Linear Models before feature engineering

Name	Train_Time	Train_R2_Score	Test_R2_Score
Lasso	0.00996	0.0000	-0.0777

Linear	0.021046	16.9169	16.9655
Ridge	0.008213	16.9168	16.9659

Fig 6. Shows the prediction graph for 100 samples. Blue colour line indicates the target value which is actual value and the prediction done with linear regression algorithm is shown by yellow line. It is observed that the model was not able to perform well.

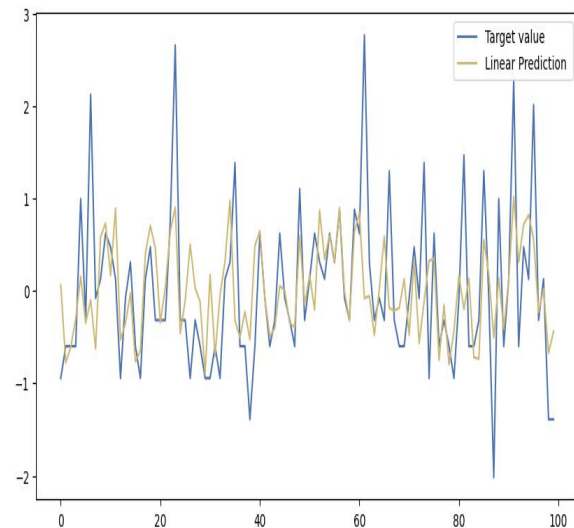


Fig. 6. Energy Prediction of Baseline model (Linear Regression) before feature engineering

5.2 Case2: Results after feature engineering without hyper parameter tuning

Table 2. shows the results for different algorithms after feature engineering. Linear and Ridge has 28% of R2 score on test data. Random forest has 67% and Extra-tree regressor has 71%. of line.

Table 2. R2 score of different algorithms after feature engineering

Name	Train_Time	Train_R2_Score	Test_R2_Score
Lasso	0.010408	0.0000	-0.0110
Linear	0.017416	30.0561	28.7402
Ridge	0.007331	30.0561	28.7407
K-Neighbors	0.250363	75.7320	61.0030
Random Forest	22.978521	95.6592	67.4624
Extra Trees	6.360532	100.0000	71.000213

Regressor			
Gradient Boosting	7.515340	47.1071	42.2821
XGBoost	1.823460	46.9366	42.1626

Fig. 7. Demonstrates the bar plot visualizing the results.

It can be seen that extra-trees is performing high than all other algorithms. It can be seen that linear and ridge are having very close test and train scores. Fig 8. Illustrates the prediction graph for 100 samples. Blue colour line indicates the target value which is actual value. The prediction done with extra tree algorithm is shown by green and random forest with red color. It can be observed that extra tree algorithm is able to mimic the original values and was able to generalize more than all the other models.

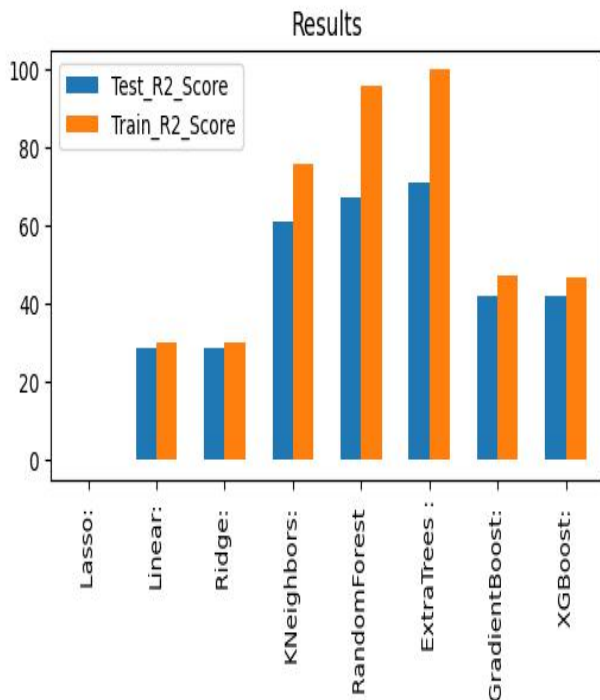


Fig. 7. R2 score for train and test sets before hyper parameter tuning

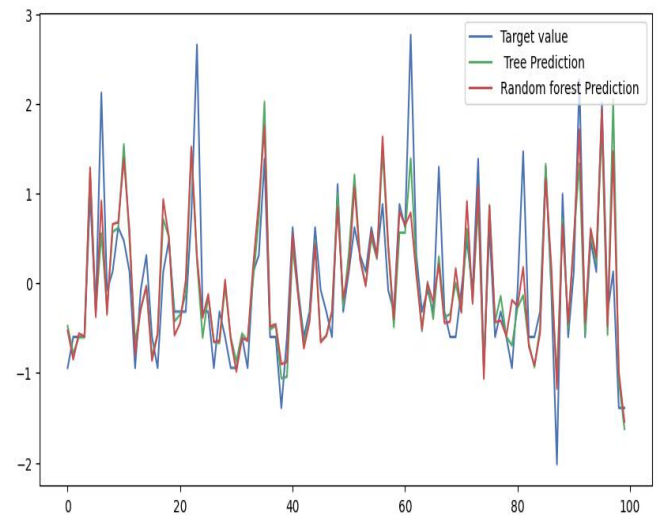


Fig. 8. Energy Prediction for Extra Trees and Random forest algorithms.

5.3 Case 3: Results with Hyper parameter tuning

The results for different algorithms after feature engineering and hyper parameter tuning are demonstrated as shown in the table 5. Linear and Ridge has 27% R2 score on test data. Random forest has 70% and Extra-tree regressor has 74% of scores.

Table 3. R2 score of different algorithms after hyper parameter tuning

Name	Train_Time	Train_R2_Score	Test_R2_Score
Lasso	0.013325	0.0000	-0.0110
Linear	0.01014	26.9782	27.1519
Ridge	0.007331	26.9782	27.1520
K-Neighbors	0.035377	77.3739	64.0166
Random Forest	15.88206	95.7882	70.1538
Extra Trees Regressor	4.9309	100.0000	74.4962
Gradient Boosting	5.104979	43.06144	40.2229
XGBoost	1.23693	43.13197	39.9348

The bar plot visualizing the results are illustrated in Fig. 9. It can be seen that extra-trees is performing high than all other algorithms. It can be seen that linear and ridges are having very close test and train scores. After hyper parameter tuning it is observed that the test and train scores have come slightly closer. It seems the models more promising now than before.

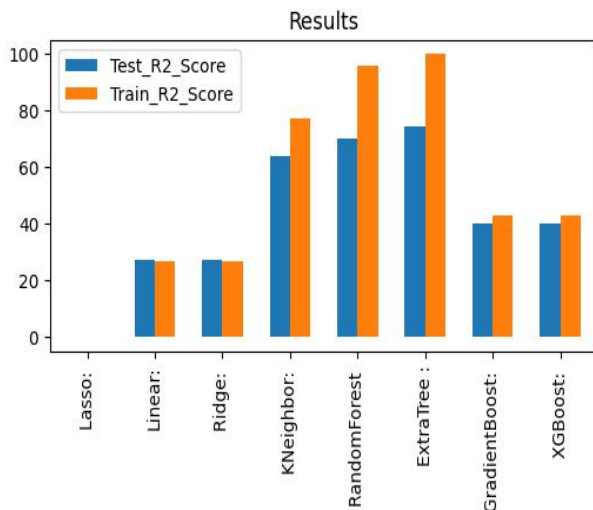


Fig. 9. R2 score for train and test sets after hyper parameter tuning

Fig 10 shows the prediction graph for 100 samples. Blue color line indicates the target value which is actual value. The prediction done with extra tree algorithm is shown by green and random forest with red color. It can be observed that extra tree algorithm is able to mimic the original values and was able to generalize more than all the other models. The winner model is extra-tree regressor. The prediction rate has increased by 4% after hyper parameter tuning. However, the sudden peaks are difficult to predict as they are not happening in general

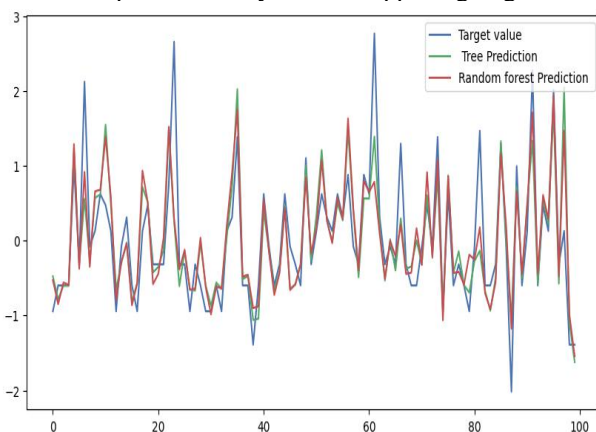


Fig. 10. Energy Prediction for Extra tree and Random forest algorithms after tuning.

Features important here are average house humidity and multiplicative features engineered to have more importance as shown in Fig. 11. Extra tree regressor is performing incredibly well, and Random forest regressor is also working well, as shown in the table above. Additional hyper parameter tuning was done on the Extra tree regressor, which resulted in an R^2 score of 74.5% of which utilising the optimal parameters as {'max_depth': 80, 'max_features': 'sqrt', 'n_estimators': 250}. As a result, predictions can be made using these models.

Lasso is at worst in this case this might be because features that are highly correlated are discarded by

making the coefficients zero except only the one which is selected. As a result, the model is not at all able to capture the pattern here.

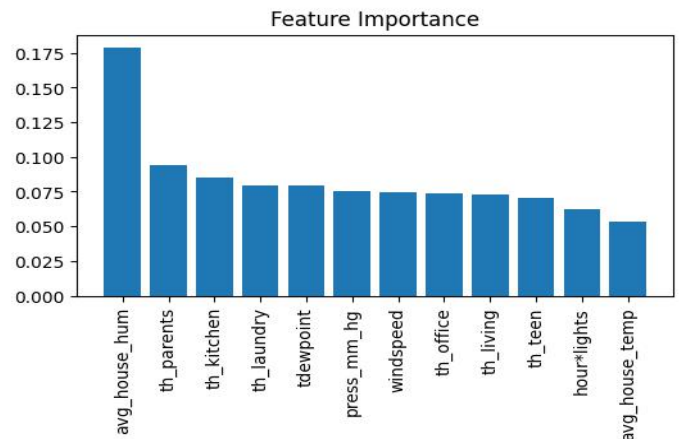


Fig. 11. Feature importance for Extra Trees regressor model

6 Conclusion

Despite being substantially linearly associated within themselves, the temperature/humidity features exhibited little to no linear (Pearson) connection with the objective variable. Although fascinating characteristics were engineered by modifying features, when it comes to energy use, the time of day is crucial. Extra Trees Regressor is the best algorithm to deploy for this dataset (tree-based algorithm). The untuned model could explain 71% of the variance (R^2 score = 0.710) on the test set, while the tuned model could explain 74.5% of the variance (R^2 score = 0.7449). Average house humidity and other features which are engineered are showing more feature importance. When dealing with a data set with most of its features having no linear correlation with the objective variable, tree-based models are by far the best model.

References

1. Md. Sumon Shahriar and M. Sabbir Rahman. Urban Sensing and Smart Home Energy Optimisations: A Machine Learning Approach. In Proceedings of the 2015 International Workshop on Internet of Things towards Applications (IoT-App '15). Association for Computing Machinery, New York, NY, USA, 19–22. (2015).
2. B Dutta, V Adhinarayanan, and W C Feng. GPU power prediction via ensemble machine learning for DVFS space exploration. In Proceedings of the 15th ACM International Conference on Computing Frontiers (CF '18). Association for Computing Machinery, New York, NY, USA, 240–243. (2018).
3. Feinberg E.A., Genethliou D. Load Forecasting. In: Chow J.H., Wu F.F., Momoh J. (eds) Applied Mathematics for Restructured Electric Power Systems. Power Electronics and Power Systems. Springer, Boston, MA. (2005)

4. Ahmad, T., Chen, H., Huang, R., Yabin, G., Wang, J., Shair, J., Akram, H.M.A., Mohsan, S.A.H. and Kazim, M., Supervised based machine learning models for short, medium and long-term energy prediction in distinct building environment. *Energy*, **158**, 17-32. (2018).
5. Ghalekhondabi, I., Ardjmand, E., Weckman, G.R. and Young, W.A., An overview of energy demand forecasting methods published in 2005–2015. *Energy Systems*, **8**(2), 411-447. (2017).
6. Sarker, I.H., Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*, **2**(3), 1-21. (2021)
7. Ng, A., Advice for applying machine learning. In *Machine learning* (2011)
8. Sarker, I.H., Alqahtani, H., Alsolami, F., Khan, A.I., Abushark, Y.B. and Siddiqui, M.K., Context pre-modeling: an empirical analysis for classification based user-centric context-aware predictive modeling. *Journal of Big Data*, **7**(1), 1-23. (2020).
9. Azuaje, Francisco & Witten, Ian & E, Frank.. Witten IH, Frank E: *Data Mining: Practical Machine Learning Tools and Techniques*. Biomedical Engineering Online - BIOMED ENG ONLINE. **5**. 1-2. (2006)
10. Basu, K., Debusschere, V. and Bacha, S., Appliance usage prediction using a time series based classification approach. In *IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society* (1217-1222). IEEE. October (2012)
11. D Bishwajit & A Vignesh & F Wu.. GPU power prediction via ensemble machine learning for DVFS space exploration. 240-243. (2018) 10.1145/3203217.3203273.
12. G Hackeling.. *Mastering Machine Learning With scikit-learn*. Packt Publishing. (2014)
13. Géron, A. 'Hands-on machine learning with Scikit-Learn and TensorFlow concepts, tools, and techniques to build intelligentsystems' , O'Reilly Media , Paperback . (2017)
14. A. Zeng, H. Ho, Y. Yu, Prediction of building electricity usage using Gaussian Process Regression, *Journal of Building Engineering* (2019) doi: <https://doi.org/10.1016/j.jobe.2019.101054>
15. Chen, Y. and Tan, H., Short-term prediction of electric demand in building sector via hybrid support vector regression. *Applied energy*, **204**, 1363-1374. (2017)
16. Ahmad T, Chen H, Nonlinear autoregressive and random forest approaches to forecasting electricity load for utility energy management systems, *Sustainable Cities and Society* (2018)
17. Geurts, Pierre & Ernst, Damien & Wehenkel, Louis. Extremely Randomized Trees. *Machine Learning*. 63. 3-42. (2006) 10.1007/s10994-006-6226-1.
18. Chen, T. and Guestrin, C., August. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (785-794). (2016)
19. Wang, R., Lu, S. and Feng, W., A novel improved model for building energy consumption prediction based on model integration. *Applied Energy*, **262**, 114561. (2020)
20. Touzani, S., Granderson, J. and Fernandes, S. Gradient boosting machine for modeling the energy consumption of commercial buildings. *Energy and Buildings*, **158**, 1533-1543. (2018)
21. Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* 12, null (2/1/2011), 2825–2830.
22. Kangji Li, Chenglei Hu, Guohai Liu, Wenping Xue, Building's electricity consumption prediction using optimized artificial neural networks and principal component analysis, *Energy & Buildings* (2015)
23. Putatunda, S. and Rama, K., 2018, November. A comparative analysis of hyperopt as against other approaches for hyper-parameter optimization of XGBoost. In *Proceedings of the International Conference on Signal Processing and Machine Learning* (6-10). (2018)
24. Dodier, R. H., and Henze, G. P. (February 12, 2004). "Statistical Analysis of Neural Networks as Applied to Building Energy Prediction ." *ASME. J. Sol. Energy Eng.* **126**(1): 592–600, February (2004)
25. Alawadi, S., Mera, D., Fernández-Delgado, M., Alkhabbas, F., Olsson, C.M. and Davidsson, P. A comparison of machine learning algorithms for forecasting indoor temperature in smart buildings. *Energy Systems*, 1-17. (2020)