Appliances’ Energy Prediction

EE 660 Project Type: Individual
Author : Siyal Sonarkar
Email : sonarkar@usc.edu
Date : 12/06/2018

1. Abstract

The project focuses on different models used for predicting the energy use of the appliances. Data used include measurements of temperature and humidity sensors from a wireless network, weather from a nearby airport station and recorded energy use of lighting fixtures. Four statistical models were trained with repeated cross validation and evaluated in a testing set: (a) Support Vector regressor, (b)Random Forest, (c) Ridge Regression and (d) Lasso Regression. The best model (Random forest) gave an error of 41.63 when test on test

2. Introduction

2.1. Problem Type, Statement and Goals

The dataset consists of energy of appliances, light fixtures in the house, temperature and humidity in different rooms, etc. The temperature and humidity conditions in the dataset were monitored with a wireless sensor network. The goal of the project is to predict the energy of the appliances using various features. Regression model is used in this problem for prediction of energy use of appliances . The problem is non-trivial as it has non-linear behaviour, that is, it is difficult to predict the energy based on the given features as it is on dependent on all of them. Also, the data had time-series feature which needed some specific pre-processing.

2.2. Literature Review

There are some articles and papers related to modelling of appliances and factors related to it that helped me to understand different data and methodologies used to understand appliances’ energy use. The methods and algorithm from the paper [1] helped me a lot in this problem. I (A study on) understood the multiple regression models used, their approach and their results on prediction of energy from [2] which was helpful.

2.3. Prior and Related Work - NONE

2.4. Overview of Approach

Different regression models like Random Forest Regressor, Linear Regression, Lasso, Ridge, SVR (Support Vector Machine Regressor), Gradient Boosting Regressor, Decision Tree Regressor and SGD Regressor were used for prediction. PCA technique is used for feature dimensionality reduction before the models were trained. The models were once compared during pre-training technique and then their hyper-parameters were tuned while using validation set. The models were compared based on their mean absolute error. The root mean squared errors were also recorded for the final model.

3. Implementation

3.1. Data Set

The dataset used for this project is ‘Appliances energy prediction dataset’ from UCI website [3]. The focus of this analysis is the energy (Wh) data logged after every 10 min for the appliances. Along with this, the house temperature and humidity conditions were also monitored using Zigbee wireless sensor network. Then, this wireless data was averaged for 10 minutes time period and merged with the energy data set by date and time. The time span of the dataset is 4.5 months. Weather data from the nearest airport weather station (Chievres Airport, Belgium) is merged with the experimental data set using date and time column. There are two random variables for testing the regression model and to filter out the non-predictive parameters. From the given time series variable, extra feature is generated : number of second until midnight (NSM).

The dataset has total of 19735 data points and 29 attributes out of which 28 attributes are integer type and 1 attribute is string.

The input variables are:

Sr no.	**Data Variables**	**Description**	**Data type**
1.	lights	Energy use of light fixtures in the house	Integer
2.	T1	Temperature in kitchen area	Integer
3.	RH_1	Humidity in kitchen area	Integer
4.	T2	Temperature in living room area	Integer
5.	RH_2	Humidity in living room area	Integer
6.	T3	Temperature in laundry room area	Integer
7.	RH_3	Humidity in laundry room area	Integer

8.	T4	Temperature in office room	Integer
9.	RH_4	Humidity in office room	Integer
10.	T5	Temperature in bathroom	Integer
11.	RH_5	Humidity in bathroom	Integer
12.	T6	Temperature outside the building	Integer
13.	RH_6	Humidity outside the building	Integer
14.	T7	Temperature in ironing room	Integer
15.	RH_7	Humidity in ironing room	Integer
16.	T8	Temperature in teenager room 2	Integer
17.	RH_8	Humidity in teenager room 2	Integer
18.	T9	Temperature in parents room	Integer
19.	RH_9	Humidity in parents room	Integer
20.	T_out	Temperature outside (from Chievres weather station	Integer
21.	Press_mm_hg	Pressure (from Chievres weather station)	Integer
22.	RH_out	Humidity outside (from Chievres weather station)	Integer
23.	Windspeed	Wind speed (from Chievres weather station)	Integer
24.	Visibility	Visibility (from Chievres weather station)	Integer
25.	Tdewpoint	Tdewpoint (from Chievres weather station)	Integer
26.	rv1	Random variable 1	Integer
27.	rv2	Random variable 2	Integer
28.	Date	Date and time format	String

Output variable:

Sr no.	**Data Variable**	**Description**	**Data type**
1.	Appliances	Energy used by appliances	Integer

3.2. Preprocessing, Feature Extraction, Dimensionality Adjustment

Preprocessing:

- First of all, the output variable which is to be predicted was separated from the dataset.
- Out of the rest 28 features, 1 feature (Date) is string type. This “Date” feature is time-series feature and had date and time format, so it was used

to generate some other features. The time series data was processed using ‘datetime’ package in python.

- The processed data is now used to generate new feature NSM (number of seconds until midnight) using “Date” feature and then “Date” feature was dropped.
- Now, all the features are in integer type and they are standardized using the StandardScalar() function.
- PCA (Principle Component Analysis) technique was implemented to the standardized data. PCA is normally used for dimensionality reduction purpose. It selects the best features and hence reduces the dimension. This step is important as it will help the data set from overfitting. The inbuilt PCA() function was used from sklearn.

Correlation between ‘output variable’ and ‘input variables’ was calculated and plotted for better understanding between the features. Correlation tells us about the strength of association between the two variable. It is 1 for total positive correlation, -1 for total negative correlation and 0 for no correlation.

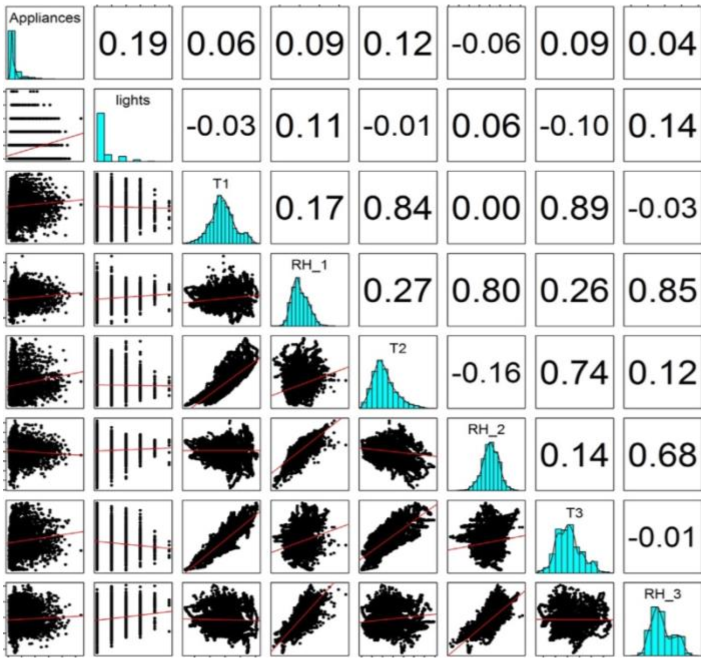
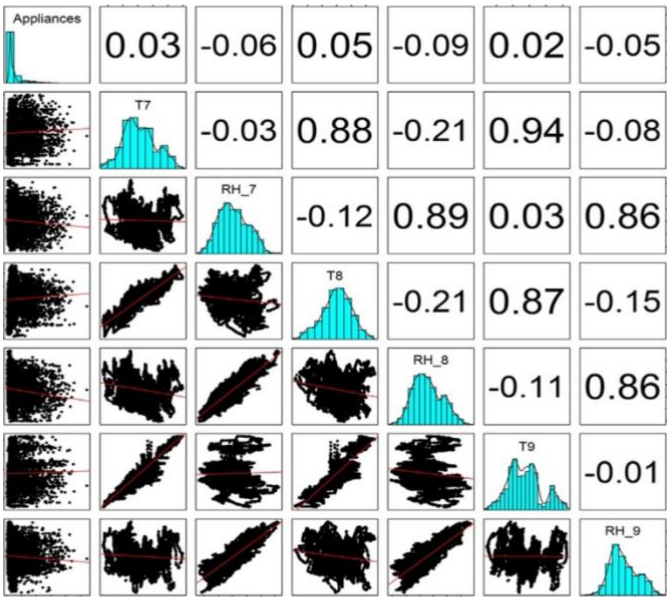


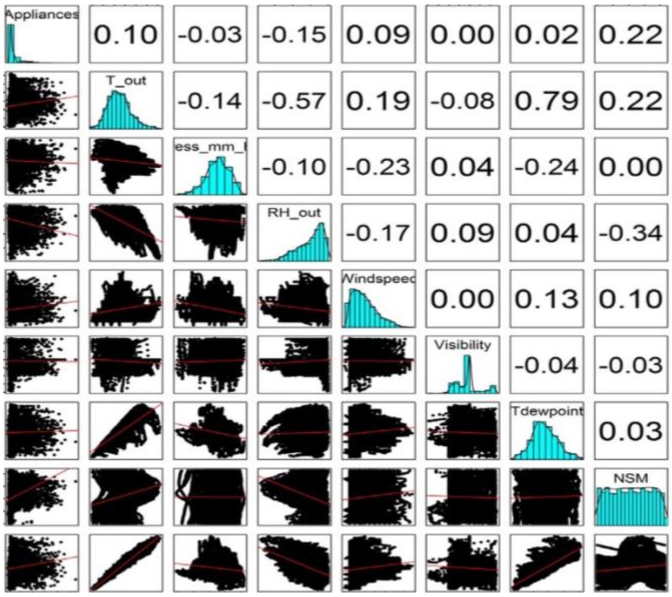
Figure 1.a: Correlation plot

In the figure 1.a, we can see that ‘appliances’ have positive correlation with ‘lights’, ‘T1’, ‘RH_1’, ‘T2’, ‘T3’ and ‘RH_3’. But it has negative correlation with ‘RH_2’.



In the figure 1.b, we can see that 'appliances' have positive correlation with 'T7', 'T8' and 'T9'. But it has negative correlation with 'RH_7', 'RH_8', 'RH_9'.

Figure 1.b: Correlation plot



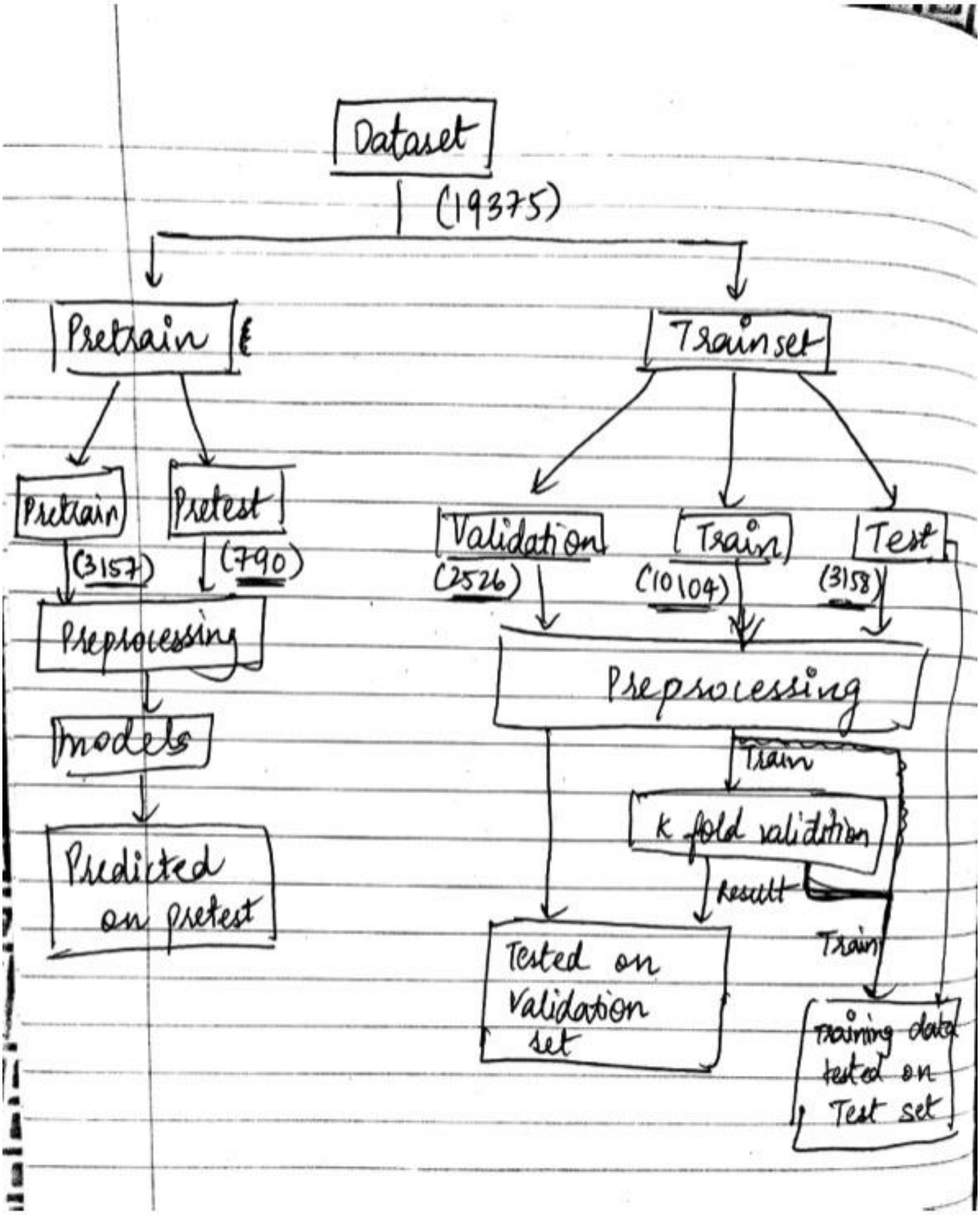
In the figure 1.c, we can see that 'appliances' have positive correlation with 'T_out', 'Windspeed', 'Tdewpoint' and 'NSM'. But it has negative correlation with 'Press_mm_Hg' and'RH_out'. It have 0 correlation with 'Visibility'.

Figure 1.c: Correlation plot

The correlation plot helped in telling the features which give positive correlation and when used for training the model might give good results. And the features having negative correlation won't have much effect on the prediction.

3.3. DatasetMethodology

A total of 19735 data point is available. This data set was divided into pre-training set, training set, validation set and test set in the following manner :



As seen from the above figure, we can see how the data is divided into different sets and then trained for different models.

- The dataset was first divided into training set (80%) and pre-training set(20%).
- The pre-training set was divided into pre-train(80%) and pre-test(20%)
- Now, the training set is further is divided into train(80%) and validation set(20%). This train set is again divided into train(80%) and test set(20%). So, now I have train validation and test sets separate which are non-overlapping.
- The pretrain set was used to find the best models for the given dataset. I took best 4 models using pretest set. Their performance was compared based on their mean absolute errors.
- Once the best 4 models were obtained, hyperparameters for these models were tuned and the best parameter was selected using the k-fold validation set. The k-fold validation was done for the train set and the number of folds used were 5. This was done for the models which were selected after the pretraining process.
- As from the flowchart, it is seen that the cross-validation is used for all the models to find the best parameter and then it is evaluated on the validation set using the best parameter.
- The validation set was used to give the best models for the given best parameter and the performance was compared based on their mean absolute error.
- Now, the best model with the best parameter was used on the test set which wasn't seen yet by the data to find the evaluation of the dataset.

3.4. TrainingProcess

I used lot of regression processes to train my model in the beginning such as

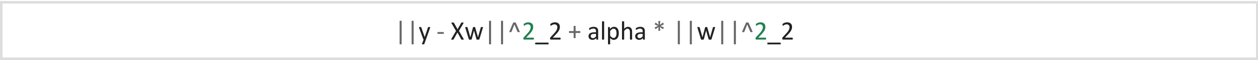
- Linear Regression
- Ridge Regression
- Lasso Regression
- Decision Tree Regressor
- Support Vector Regression
- Random Forest Regressor
- Gradient Boosting Regressor
- SGD Regressor

The best 4 processes were selected using pre-train set. And they were used for k-fold validation using training test to compare it's performance on validation set:

- Support Vector Regression (SVR):
  SVR uses the same principles as the SVM for classification but there are some difference. The result of the regression should be a real number and it is quite difficult to predict the information which has infinite possibilities. So, in case of regression, a margin of tolerance (epsilon) is set in approximation to SVM. Therefore, it is seen we have two hyperparameters for SVR here; epsilon and alpha.
  For linear SVR, the equation is:
  $$y = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*). \langle x_i, x \rangle + b$$
  And for non-linear SVR, we use the kernel function to make it linear.
  I used an inbuilt SVR function from sklearn: SVR(alpha = a, epsilon)
  I just used the 'alpha' parameter while cross validation.

- Random Forest Regressor:
  Random forest for regression operates by constructing a multitude decision trees at training time. It uses number of estimators that fit a number of decision trees on various sub-samples of dataset. It uses averaging to improve the predicted results and controls over-fitting. The main advantage of random forest regression is that prevents over-fitting. Random Forest is good at handling numerical and categorical data and it is also able to capture the non-linear interaction between the features and the target.
  I used an inbuilt sklearn.ensemble.RandomForestRegressor(n_estimators)
  I used n_estimators as my hyperparamter to tune the model and found the best parameter using validation set.

- Ridge Regression:
  Ridge Regression analyses multiple regression data that suffer from multicollinearity. The standard approach is ordinary least squares linear regression. It is also called as L2 regularization. The relation between the variable and the target is estimated and the model takes the form:
  $$\|A\mathbf{x} - \mathbf{b}\|_2^2 + \|\Gamma\mathbf{x}\|_2^2$$
  I used an inbuilt function sklearn.linear_model.Ridge(alpha = a)
  Here, the optimization objective for Ridge is:



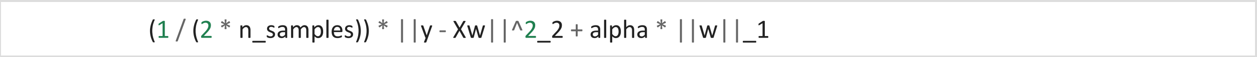
```
||y - Xw||^2_2 + alpha * ||w||^2_2
```

The parameter ‘alpha’ was used to tune the model using k-fold validation.

- Lasso Regression:
  Lasso is a regression analysis method that performs both variable selection and regularization in order to enhance the prediction accuracies. Lasso is a linear model trained with L1 prior as regularizer. Lasso is generally defined for least squares. Lasso Regression takes the form:

$$\min_{\beta_0,\beta}\left\{\frac{1}{N}\left\|y-\beta_0 1_N - X\beta\right\|_2^2\right\} \text{ subject to } \|\beta\|_1 \leq t\,.$$

I used the inbuilt function sklearn.linear_model.Lasso(alpha=a)
Here, the optimization objective for Lasso is:



```
(1 / (2 * n_samples)) * ||y - Xw||^2_2 + alpha * ||w||_1
```

I used some linear and non-linear method, I was not sure about the dataset which I was working on that if it is linear between the variables and the target. This approach helped to optimize my problem as used most of the regressor to see which gives me the best result and for which parameter.

The validation set gave me one best model using k-fold validation which is, Random Forest Regressor for my dataset based on the mean absolute error. The best parameter obtained was used and the mean absolute error for the test set was determined. Moreover, Random Forest helped to prevent the overfitting of the data as well.

3.5. Model Selection and Comparison of Results:

Model Selection for Pre-training method:
The models were selected based on their performance on pre-test set. The results found for each model was as follows:

- Linear Regression : mean_absolute_error = 57.83457159621355
- Ridge Regression : mean_absolute_error = 53.19240506329114
- Lasso Regression : mean_absolute_error = 57.829765255753905
- Decision Tree Regressor : mean_absolute_error = 66.0
- Support Vector Regression : mean_absolute_error = 48.35771097237954
- Random Forest Regressor : mean_absolute_error = 52.783240970487995
- GradientBoostingRegressor: mean_absolute_error= 58.065209761479586
- SGD Regressor : mean_absolute_error = 60.076407445604524

Validation Set and training set:
So, from the above models, best 4 models were selected for validation. The results obtained are as follows:

- Support Vector Regression(SVR)
  The model was trained for parameter ‘C’ and the best parameter was selected based on its performance. For each value of ‘C’, the mean_absolute_error (MAE) was calculated. The results are found from the k-fold validation set:

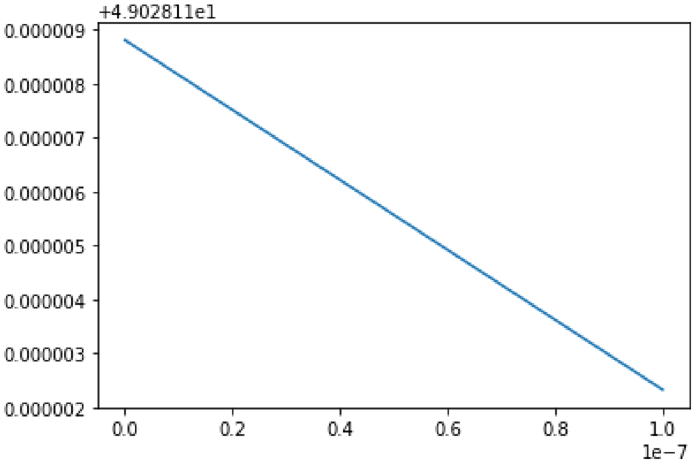
```
optimal_C :=  1e-07   MAE=  49.02811232350326
optimal_C :=  4.641588833612782e-08   MAE=  49.02811580024293
optimal_C :=  2.1544346900318822e-08   MAE=  49.02811741400255
optimal_C :=  1e-08   MAE=  49.02811816304339
optimal_C :=  4.641588833612773e-09   MAE=  49.02811851071737
optimal_C :=  2.1544346900318866e-09   MAE=  49.02811867209331
optimal_C :=  1e-09   MAE=  49.028118746997414
optimal_C :=  4.6415888336127913e-10   MAE=  49.028118781764796
optimal_C :=  2.1544346900318867e-10   MAE=  49.028118797902394
optimal_C :=  1e-10   MAE=  49.02811880539281


For kfold validation set :
optimal_C :=  1e-07   MAE=  49.02811232350326
```

  Using the above best parameter, the model’s performance for validation set was calculated.

```
For validation set :
optimal_C :=  1e-07   MAE=  48.52295262732591
```

  The graph between the parameter ‘C’ and MAE was plotted. It is observed that as the parameter value increases, the error decreases. And the relation is linear.



- Random Forest Regressor

  The model was trained for parameter ‘n_estimators’ and the best parameter was selected based on its performance. For each value of ‘n_estimators’, the mean_absolute_error (MAE) was calculated. The results are found from the k-fold validation set:

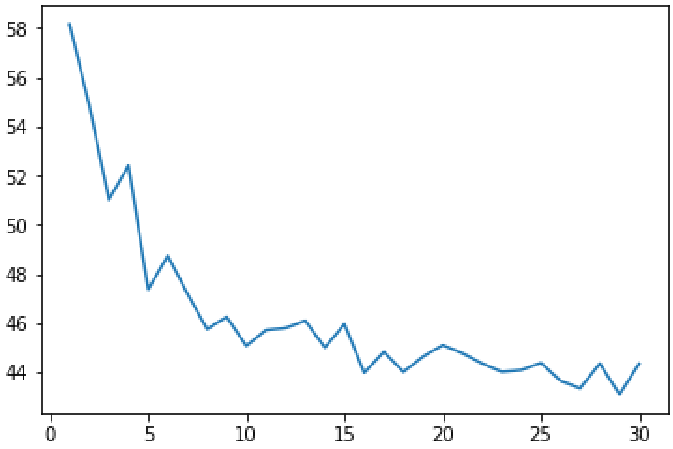
```
optimal_est :=  1   MAE=  58.16831683168317
optimal_est :=  2   MAE=  54.868811881188115
optimal_est :=  3   MAE=  51.01980198019802
optimal_est :=  4   MAE=  52.41955445544554
optimal_est :=  5   MAE=  47.375247524752474
optimal_est :=  6   MAE=  48.75165016501651
optimal_est :=  7   MAE=  47.20226308345121
optimal_est :=  8   MAE=  45.75928217821782
optimal_est :=  9   MAE=  46.25962596259625
optimal_est :=  10   MAE=  45.07871287128713
optimal_est :=  11   MAE=  45.72367236723672
optimal_est :=  12   MAE=  45.804455445544555
optimal_est :=  13   MAE=  46.0997715156131
optimal_est :=  14   MAE=  45.01202263083451
optimal_est :=  15   MAE=  45.96897689768977
optimal_est :=  16   MAE=  43.98329207920792
optimal_est :=  17   MAE=  44.83110075713454
optimal_est :=  18   MAE=  44.02035203520352
optimal_est :=  19   MAE=  44.639656070870245
optimal_est :=  20   MAE=  45.105445544554456
optimal_est :=  21   MAE=  44.77652050919377
optimal_est :=  22   MAE=  44.36251125112511
optimal_est :=  23   MAE=  44.02346104175635
optimal_est :=  24   MAE=  44.09261551155116
optimal_est :=  25   MAE=  44.380594059405944
optimal_est :=  26   MAE=  43.65536938309216
optimal_est :=  27   MAE=  43.35295196186286
optimal_est :=  28   MAE=  44.35731966053748
optimal_est :=  29   MAE=  43.102424035507
optimal_est :=  30   MAE=  44.34620462046205


For kfold validation set :
optimal_est :=  29   MAE=  43.102424035507
```

  Using the above best parameter, the model’s performance for validation set was calculated.

```
For validation set :
optimal_Est :=  29   MAE=  42.13339885876539
```

The graph between the parameter ‘n_estimators’ and MAE was plotted. It is observed that as the parameter value increases, the error decreases. But after some time, the error starts increases and so the best parameter which has least error is selected.



- Ridge Regression

The model was trained for parameter ‘alpha’ and the best parameter was selected based on its performance. For each value of ‘alpha’, the mean_absolute_error (MAE) was calculated. The results are found from the k-fold validation set:

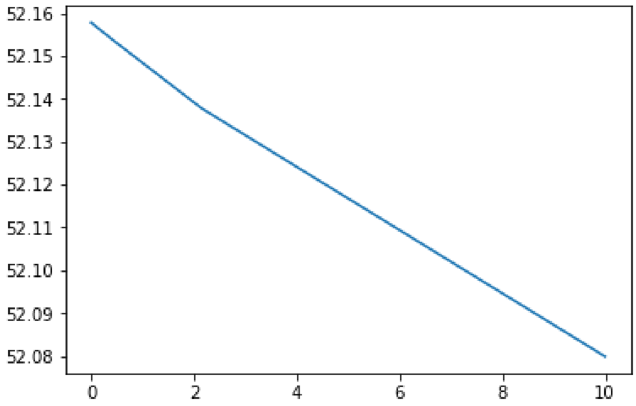
```
optimal_a :=  1e-05   MAE=  52.1577564461866
optimal_a :=  4.641588833612782e-05   MAE=  52.157756092768444
optimal_a :=  0.00021544346900318823   MAE=  52.15775445235114
optimal_a :=  0.001   MAE=  52.15774683830319
optimal_a :=  0.004641588833612777   MAE=  52.157711499064135
optimal_a :=  0.021544346900318822   MAE=  52.15754890871682
optimal_a :=  0.1   MAE=  52.15679773516685
optimal_a :=  0.46415888336127725   MAE=  52.153330610478214
optimal_a :=  2.154434690031882   MAE=  52.137734716306056
optimal_a :=  10.0   MAE=  52.07989266477602


For kfold validation set :
optimal_alpha :=  10.0   MAE=  52.07989266477602
```

Using the above best parameter, the model’s performance for validation set was calculated:

```
For validation set :
optimal_alpha :=  10.0   MAE=  51.74735848993796
```

The graph between parameter ‘alpha’ and MAE was plotted. And it was observed that as the value of alpha increases, the error decreases but after some value, the error starts increasing, so the optimal alpha was selected using the k-fold method.



- Lasso Regression

  The model was trained for parameter ‘alpha’ and the best parameter was selected based on its performance. For each value of ‘alpha’, the mean_absolute_error (MAE) was calculated. The results are found from the k-fold validation set:

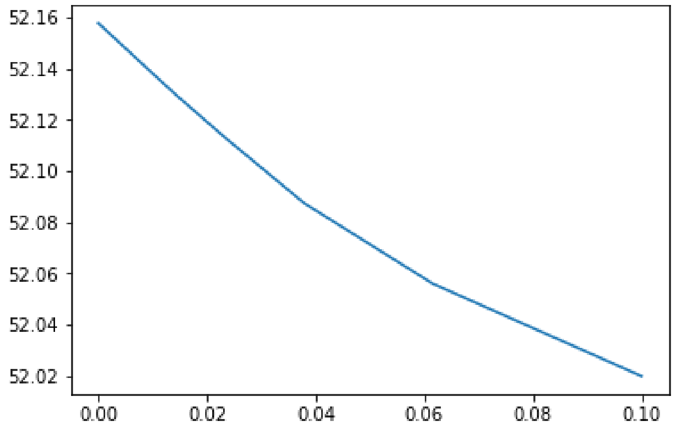
```
optimal_a :=  1e-05   MAE=  52.15773679845672
optimal_a :=  1.623776739188721e-05   MAE=  52.15772448185268
optimal_a :=  2.6366508987303556e-05   MAE=  52.15770448243754
optimal_a :=  4.281332398719396e-05   MAE=  52.15767200785243
optimal_a :=  6.951927961775606e-05   MAE=  52.15761927637652
optimal_a :=  0.00011288378916846884   MAE=  52.15753367047519
optimal_a :=  0.00018329807108324357   MAE=  52.157395039513744
optimal_a :=  0.00029763514416313193   MAE=  52.15717021189538
optimal_a :=  0.0004832930238571752   MAE=  52.156805142038344
optimal_a :=  0.0007847599703514606   MAE=  52.156212350096304
optimal_a :=  0.0012742749857031334   MAE=  52.15524978832966
optimal_a :=  0.00206913808111479   MAE=  52.15368680292295
optimal_a :=  0.0033598182862837810   MAE=  52.151148863575834
optimal_a :=  0.005455594781168515   MAE=  52.147031396657795
optimal_a :=  0.008858667904100823   MAE=  52.14034767895506
optimal_a :=  0.01438449888287663   MAE=  52.1295554429152
optimal_a :=  0.023357214690901212   MAE=  52.11299774404569
optimal_a :=  0.03792690190732246   MAE=  52.08734597152981
optimal_a :=  0.06158482110660261   MAE=  52.05591689637865
optimal_a :=  0.1   MAE=  52.01979767365462

For kfold validation set :
optimal_alpha :=  0.1   MAE=  52.01979767365462
```

Using the above best parameter, the model’s performance for validation set was calculated:

```
For validation set :
optimal_alpha :=  0.1   MAE=  51.687955434164095
```

The graph between parameter ‘alpha’ and MAE was plotted. It was observed that the relation was not linear and the optimal result for the parameter was recorded.



The best model from this method was Random Forest and the best parameter is also noted.

4. Final Results and Interpretation

For test set, the best model obtained after validation was used and the best parameter as well. The model performance was now tested on the test set and the results were recorded as follows:

```
optimal_Est :=  29   MAE=  41.63143412460964
```

It was observed that the random forest gives best results as it prevents overfitting and optimizes the data. It is also seen that the error is not that low and this might be due to the dataset available. The main reason for this is that appliances’ consumption profile is highly variable. The features which were present were not all dependent on the target as observed from the correlation results and that resulted into high error. Whereas, elimination of these features could have helped a little but they play different roles for different regressors and that’s why it was little difficult to determine which one to consider and which one to eliminate for a regressor. Therefore, I just found which features were more important for the model performance and arranged them in an order where top one is the best feature and last one is the least important feature. The result found is as follows:

1. NSM
2. Lights
3.Press
4.RH5
5.T3
6. RH3
7. Tdwpt
8.T5
9.T8
10. RH1
11. RH9
12. RH7
13. T7
14. Visibility
15. RH4
16. RH2
17. T1
18. T2
19. T9
20. T4
21. WindSpeed
22.RH8
23.RH6
24. RHout
25. Tout
26. T6

Conclusion:
The Random Forest Regressor was the best model when compared with rest all models for this data set. For all the models, the feature ‘NSM’ is marked as the important feature. The mean absolute error for Random Forest Regressor is:

optimal_Est := 29 MAE= 41.63143412460964

This regressor worked the best because it prevents the data from overfitting and plus it predicted very well based on all the features which is important. Few features were not useful for some models but to create a common platform they were tested for all features. The error obtained is less than 10 when compared with other models.
The only drawback of this dataset was that it was collected from one single house and that’s why we can’t expand for future scope.

6. Summary and conclusions

Problem: Regression Problem
Best Model : Random Forest Regressor
Best parameter: 29
Mean absolute error: 41.63

7. References

[1]https://www.sciencedirect.com/science/article/pii/S0378778816308970?via%3Dihub
[2] https://www.sciencedirect.com/science/article/pii/S0360544212002903
[3] https://archive.ics.uci.edu/ml/datasets/Appliances+energy+prediction