Appliances Energy Prediction

Project Description

- **Domain Background:**

  The background domain of this project is energy usage prediction inside a home. In the usual setting, different sensors are attached inside the home and the energy usage is also calculated. All readings are taken at regular intervals. The goal is to predict energy consumption of appliances. In this era of smart homes, energy usage prediction can lead to efficient energy management. This also helps in preventing frequent power cuts.

- **Project Overview:**

  This project aims to predict the energy consumption by home appliances. With the advent of smart homes and rising need for energy management, existing smart home systems can benefit from accurate prediction. If the energy usage can be predicted for every possible state of appliances, then device control can be optimized for energy savings as well. This is a case of Regression analysis which is part of the Supervised Learning problem. Appliance energy usage is the target variable while sensor data and weather data are the features.

- **Problem Statement:**

  Develop a Supervised learning model using Regression algorithms to predict the appliance energy usage using sensor readings and weather data as features.

Evaluation-metric

Since this is a regression problem, the metric used will be "Coefficient of Determination", in other words denoted as R^2 (R squared) which gives a measure of the variance of target 2 variable that can be explained using the given features. It can be mathematically defined as:

$$R^2 = 1 - \frac{SS_{Regression—}}{SS_{Total}}$$

where,

$SS_{Regression—}$ = Residual sum of squares

SS_{Total} = Total sum of squares

For this project, We are using 'r2_score()' function of the metrics module of scikit-learn library. While "Coefficient of Determination" provides relative measure of the how well the model fits the data, the RMSE (Root Mean Squared Error) gives absolute measure of how well model fits the data i.e. how close are the predicted values to the actual values,

Mathematically, RMSE can be defined as

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(y_j - \hat{y}_j\right)^2}$$

where,

n = number of observations

y_j = Actual value of target variable

$\hat{y}_j$ = Predicted value of target variable

In this project, I will calculate RMSE by calculating the square root of the mean_squared_error() function provided in the metrics module of scikit-learn library. Therefore, the metrics to be used are: i. R2 score ii. RMSE

These two metrics are helpful for this problem because of the following reasons:

i. It is a Regression based problem.

ii. R2 score will show the statistical robustness of the model. iii. RMSE will give an idea about how accurate the predictions are to actual values

Data Summary

The dataset has 28 features and 1 target variable described as follows:

NAME	DESCRIPTION	UNIT
T1	Kitchen Temperature	Celsius
T2	Living Room Temperature	Celsius
T3	Laundry Room Temperature	Celsius
T4	Office Temperature	Celsius
T5	Bathroom Temperature	Celsius
T6	Temperature outside building (North)	Celsius
T7	Ironing Room Temperature	Celsius
T8	Teenager Room Temperature	Celsius
T9	Parents Room Temperature	Celsius
T_out	Outside Temp (Weather station)	Celsius
T_dewpoint	Dew point Temp (Weather station)	Celsius
Date	Time stamp of sensor data	Datetime
RH_1	Kitchen Humidity	%
RH_2	Living Room Humidity	%
RH_3	Laundry Room Humidity	%
RH_4	Office Humidity	%
RH_5	Bathroom Humidity	%
RH_6	Humidity Outside (North)	%
RH_7	Ironing Room Humidity	%
RH_8	Teenager Room Humidity	%
RH_9	Parents Room Humidity	%
RH_out	Outside humidity (Weather station)	%
Pressure	Outside Pressure (Weather station)	mmHg

Windspeed	**Outside Wind speed (weather station)**	**m/s**
Visibility	**Visibility(weather station)**	**km**
Rv1	**Random Variable 1**	**-**
Rv2	**Random Variable 2**	**-**
Lights	**Energy used by lights**	**Wh**
	TARGET VARIABLE	
Appliances	**Total Energy used by appliances**	**Wh**

Since most of the value in the lights column is 0, it won't be playing much role in our model. Hence we drop the lights feature from our dataframe. We don't need random variable features as well.

Therefore, Number of features = 25
Number of target variables = 1 (Appliances)
Number of instances in training data = 15,788
Number of instances in testing data = 3,947
Total number of instances = 19,735
Count of Null values = 0
All features have numerical continuous variables.
There are no categorical or ordinal features in this dataset

Project Design

The general sequence of steps are as follows :

- **Data Visualization:** Visual representation of data to find the degree of correlations between predictors and target variables and find correlated predictors. Additionally, we can see ranges and visible patterns of the predictors and target variables.
- **Data Preprocessing:** Scaling and Normalization operations on data and splitting the data in training, validation and testing sets.
- **Feature Engineering:** Finding relevant features, engineer new features using methods like PCA if feasible.

- **Model Selection:** Experiment with various candidate algorithms to find out the best algorithm for this use case.
- **Model Tuning:** Fine tune the selected algorithm using hyper-parameter tuning to increase performance without overfitting.
- **Testing:** Evaluate the model on testing dataset

Descriptive statistics

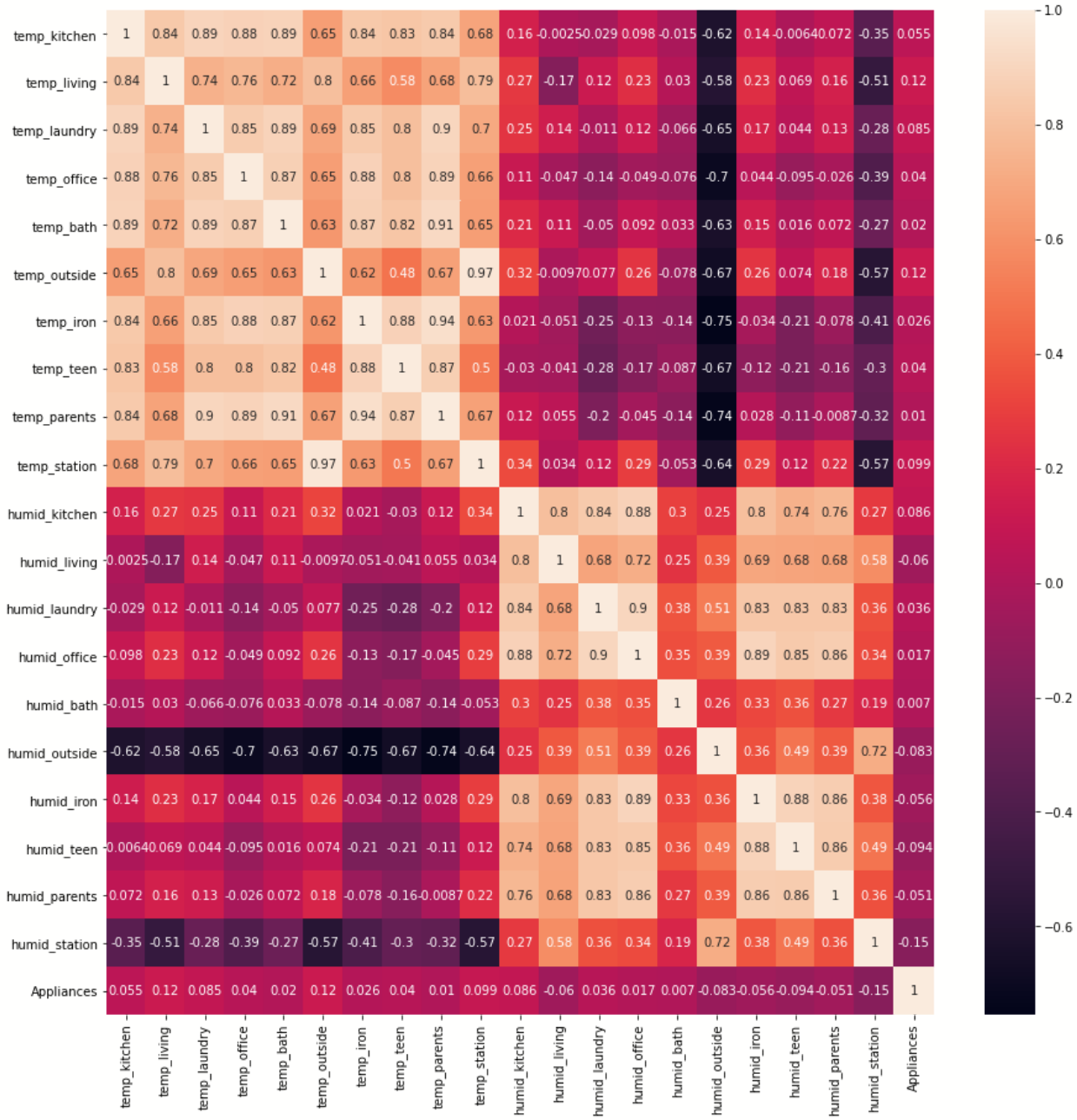
i. Distribution of our features :

	temp_kitchen	temp_living	temp_laundry	temp_office	temp_bath	temp_outside	temp_iron	temp_teen	temp_parents	temp_station
count	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000
mean	21.686571	20.341219	22.267611	20.855335	19.592106	7.910939	20.267106	22.029107	19.485828	7.411665
std	1.606066	2.192974	2.006111	2.042884	1.844623	6.090347	2.109993	1.956162	2.014712	5.317409
min	16.790000	16.100000	17.200000	15.100000	15.330000	-6.065000	15.390000	16.306667	14.890000	-5.000000
25%	20.760000	18.790000	20.790000	19.530000	18.277500	3.626667	18.700000	20.790000	18.000000	3.666667
50%	21.600000	20.000000	22.100000	20.666667	19.390000	7.300000	20.033333	22.100000	19.390000	6.916667
75%	22.600000	21.500000	23.290000	22.100000	20.619643	11.256000	21.600000	23.390000	20.600000	10.408333
max	26.260000	29.856667	29.236000	26.200000	25.795000	28.290000	26.000000	27.230000	24.500000	26.100000

1. Average outside temperature over a period of 4.5 months is around 7.5 degrees. It ranges from -6 - 28 degrees.
2. While average temperature inside the building has been around 20 degrees for all the rooms. It ranges from 14 - 30 degrees.
3. Which implies, Warming appliances have been used to keep the insides of the building warm. There must be some sort of direct correlation between temperature and consumption of energy inside the house.

	humid_kitchen	humid_living	humid_laundry	humid_office	humid_bath	humid_outside	humid_iron	humid_teen	humid_parents	humid_station
count	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000	19735.000000
mean	40.259739	40.420420	39.242500	39.026904	50.949283	54.609083	35.388200	42.936165	41.552401	79.750418
std	3.979299	4.069813	3.254576	4.341321	9.022034	31.149806	5.114208	5.224361	4.151497	14.901088
min	27.023333	20.463333	28.766667	27.660000	29.815000	1.000000	23.200000	29.600000	29.166667	24.000000
25%	37.333333	37.900000	36.900000	35.530000	45.400000	30.025000	31.500000	39.066667	38.500000	70.333333
50%	39.656667	40.500000	38.530000	38.400000	49.090000	55.290000	34.863333	42.375000	40.900000	83.666667
75%	43.066667	43.260000	41.760000	42.156667	53.663333	83.226667	39.000000	46.536000	44.338095	91.666667
max	63.360000	56.026667	50.163333	51.090000	96.321667	99.900000	51.400000	58.780000	53.326667	100.000000

1. Average humidity outside the building has been higher than the average humidity inside.
2. Average humidity at the weather station is significantly higher compared to outside humidity near the building.
3. Average humidity in the bathroom is significantly higher compared to other rooms due to obvious reasons.
4. Kids and parent rooms show a comparatively higher average humidity as well signifying the fact that inhabitants of this building spend most of their time in these buildings.



ii. Observations from Correlation plot :

a. Features related to temperature

Almost all the temperature measures in different rooms are highly linearly correlated with each other. However there is little to no correlation between temperature features and target variables.

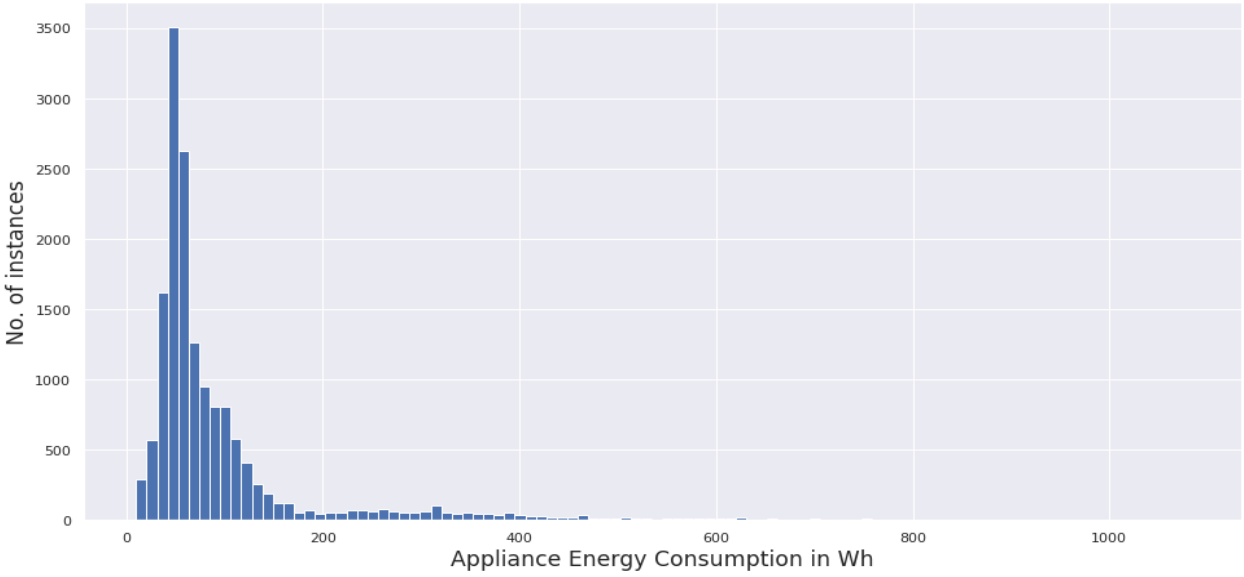
b. Features related to humidity

Almost all the humidity measures in different rooms are highly linearly correlated with each other. However there is little to no correlation between humidity features and target variables.

c. Weather data

Dewpoint shows a higher correlation with most of the temperature and humidity level features than any other weather parameters. However its correlation with the target variable is pretty low. Pressure, wind speed and visibility show little correlation with temperature and humidity features as well as the target variable.

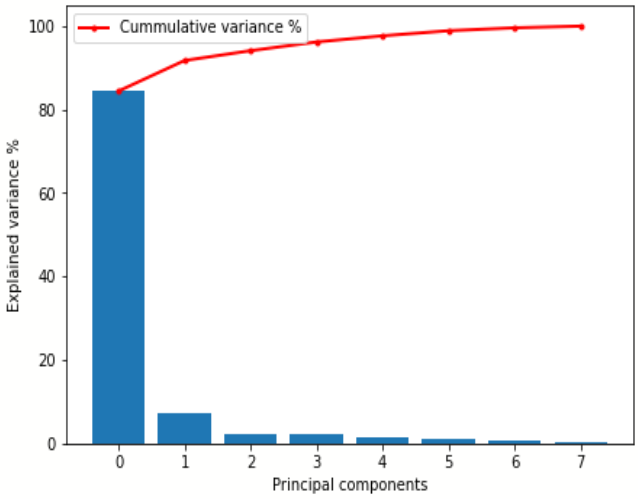
iii. Distribution of the target variable :



- Energy consumption of appliances ranges from 10 Wh to 1080 Wh
- About 75 % of energy values lie below 100 Wh, and about 93 % of them lie below 300 Wh
- Our target variable seems to be highly skewed, and our task is to predict the usual as well as the large surges in energy in the building

Feature Engineering

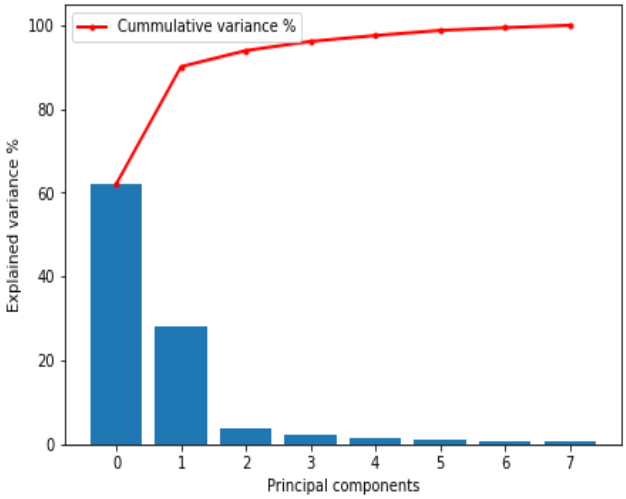
Principal component Analysis on Temperature features :



- Given, temperature levels in different rooms had a very low correlation with target variable, and high correlation among themselves, we reduce the feature set into lower dimensions that could explain maximum variance.

- PCA 1 and 2 explain more than 91 % variance in the temperature levels in different rooms in the building

Principal component Analysis on Humidity features :



- Given, humidity levels in different rooms had a very low correlation with target variable, and high correlation among themselves, we reduce the

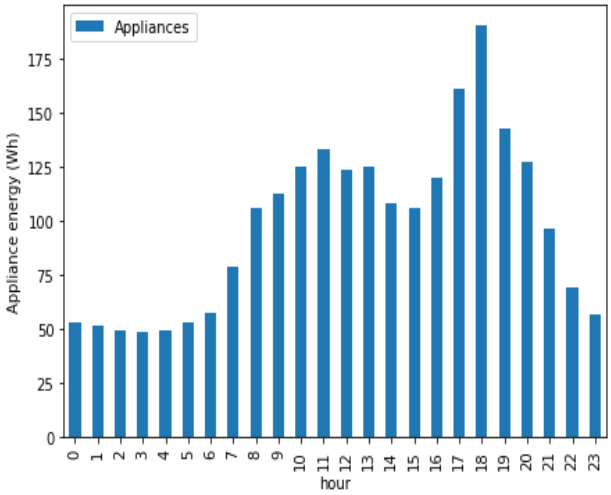
feature set into lower dimensions that could explain maximum variance.

- PCA 1 and 2 explain more than 91 % variance in the humidity levels in different rooms in the building

Building new engineered feature :

Apart from features such as temperature and humidity levels in rooms and outside, we have another important column known as date which holds timestamps for each of the samples from sensor data.

We derive the hour data for each sample from the time stamps and try to observe if there is any particular pattern of energy consumption over the course of a day. The below image describes the average energy consumption of appliances in a given hour of the day over the course of 4.5 months.



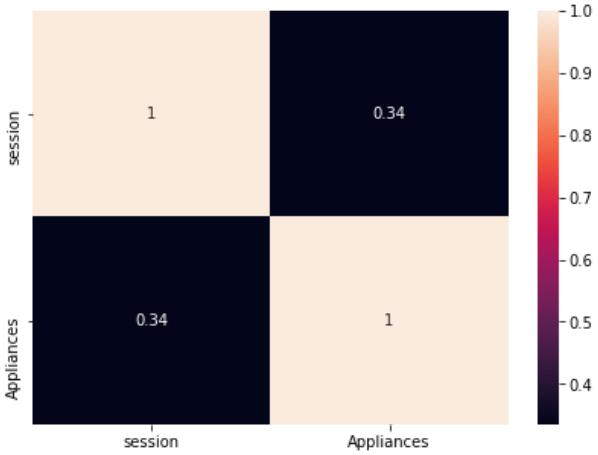
We observe two peak hours. One at 11 am in the morning and other at 6 PM in the evening. While the peak at 11 am is shallow and low, peak at 6 PM is comparatively higher and sharper.

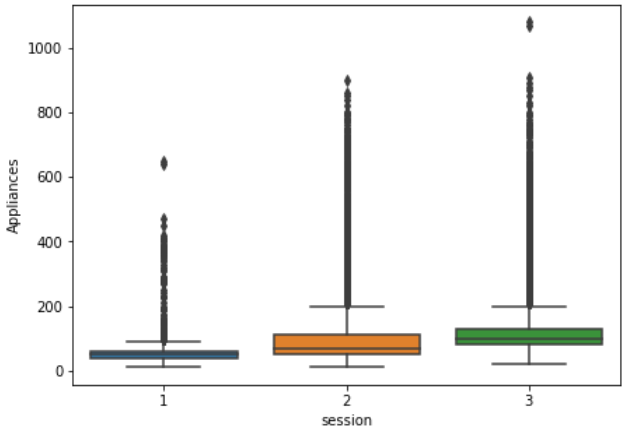
We observe that over the sleeping hours (10 PM - 6 AM) the energy consumption of appliances is around 50 Wh. After about 6 AM, energy consumption starts to rise gradually up until 11 AM (probably due to morning chores). And then gradually decreases to around 100 Wh at about 3 PM. After which the energy consumption drastically shoots up until 6 PM in the evening (probably due to requirement lights in rooms). However energy consumption of appliances reverts back to 50 Wh, as night approaches and people in the house go to bed at around 10 PM.

Based on this observation, we created following classes :

- Class 1 : 10 PM - 6 AM
- Class 2 : 6 AM - 3 PM
- Class 3 : 3 PM - 10 PM

Dependence of Appliance energy consumption our new feature:





Final Features

Feature set - I (With PCA features)

- Session
- Temperature - pca 1
- Temperature - pca 2
- Humidity - pca 1
- Humidity - pca 2
- Temperature outside
- Humidity outside
- Pressure
- Wind speed

Feature set - II (Without PCA features)

- Session
- Temperature of all rooms inside building
- Humidity of all rooms inside building
- Temperature outside
- Humidity outside
- Pressure
- Wind speed

Model Training and Evaluation

We will try the following algorithms for Regression:

The most basic Regression algorithm is Linear Regression. If a Linear model can explain the data well, there is no need for further complexity. As modification to original Least Squares Regression, we can apply Regularization techniques to penalize the coefficient values of the features, since higher values generally tend towards overfitting and loss of generalization. Regularization techniques enhance performance of Linear models greatly. Also, there are very few practical cases when a Linear model can fit the data well without Regularization. In case of Regularization, depending upon whether we add the absolute values of coefficients or their squares to our loss function, the problem of Linear Regression is transformed into Lasso or Ridge Regression respectively.

i. Linear Models

1. Linear Regression
2. Ridge Regression
3. Lasso Regression

The next category of algorithms are of Tree based Regression models. An important advantage of Tree based models is that they are robust to outliers compared to Linear models. We haven't seen a Linear relationship between any feature and the target variable, it is likely that Regression trees will turn out to be better than Linear models.

Given the substantial number of features, it is evident that a Decision Tree will overfit the data. Hence, we have skipped it and directly jumped towards ensemble methods listed below, which include building multiple regressors on copies of same training data and combining their output either through mean, median, mode (Bagging) or growing trees sequentially (i.e. each tree is built from data of the previous tree) and using weighted average of these weak learners (a learner which performs just a little better than chance (50%)) (Boosting).

Random Forests is one of the primary Bagging methods and works well on high dimensional data like ours. Extra Trees Regressor goes one step further by making splits Random. Gradient Boosting Machines is a type of Boosting method. It builds an additive model in a way that performance always increases.

ii. Tree based models

- Random Forests Regressor
- Extra Trees Regressor
- XG Boost Algorithm
- Gradient Boosted Trees

Among other algorithms are:

- K Neighbors Regressor
- Support Vector Regressor
- MLP Regressor

Methodology

Data scaling

Temperature	**-6 to 30 C**
Humidity	**1 to 100 %**
Windspeed	**0 to 14 m/s**
Visibility	**1 to 66 km**
Pressure	**729 to 772 mmHg**
Appliance Energy Usage	**10 to 1080 Wh**

Due to different ranges of features, it is possible that some features will dominate the Regression algorithm. To avoid this situation, all features need to be scaled. Thus, the data was scaled to 0 mean and unit variance using the *StandardScaler* class in *sklearn.preprocessing* module.

Implementation

The model implementation is done in 3 steps:

i. *Created a pipeline* to execute each Regressor and record the metrics.

ii. *Pass each Regressor* to the above pipeline.

iii. Consolidate the obtained metrics into a DataFrame and plot these metrics using a bar graph. List of Algorithms tested:

i. *sklearn.linear_model.Ridge*
ii. *sklearn.linear_model.Lasso*
iii. *sklearn.ensemble.RandomForestRegressor*
iv. *sklearn.ensemble.GradientBoostingRegressor*
v. *sklearn.ensemble.ExtraTreesRegressor*
vi. *xgboost.xgb.XGBRegressor*
vii. *sklearn.svm.SVR*
viii. *sklearn.neighbors.KNeighborsRegressor*

ix. sklearn.neural_network.MLPRegressor

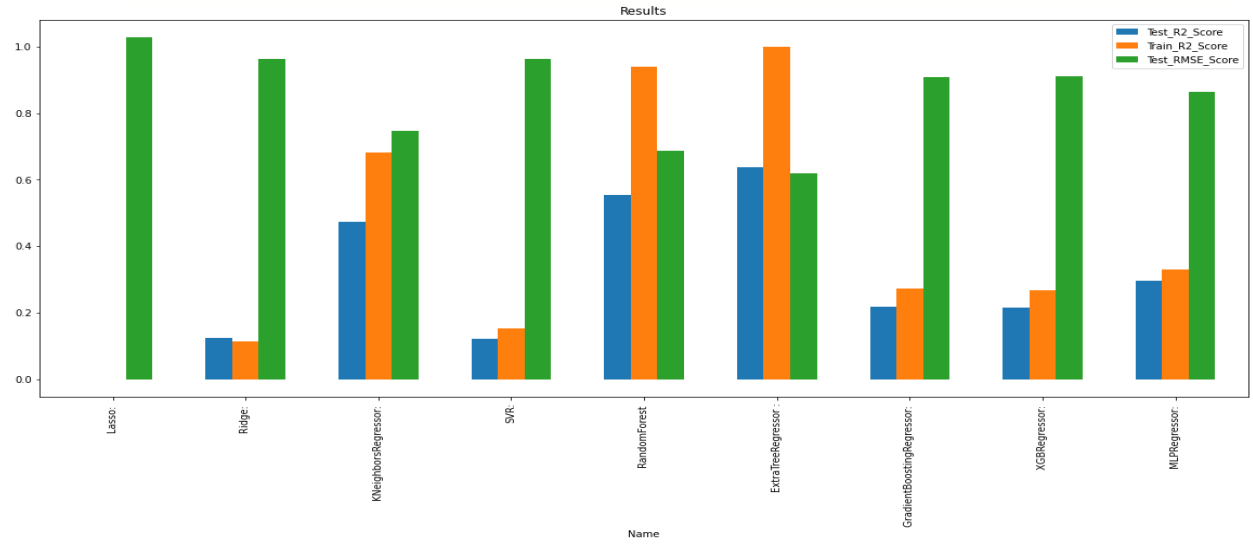
Performance metric used:

R2 score (the r2_score() method mentioned in section 2) which is internally used by the score() method of all Regressors mentioned above.

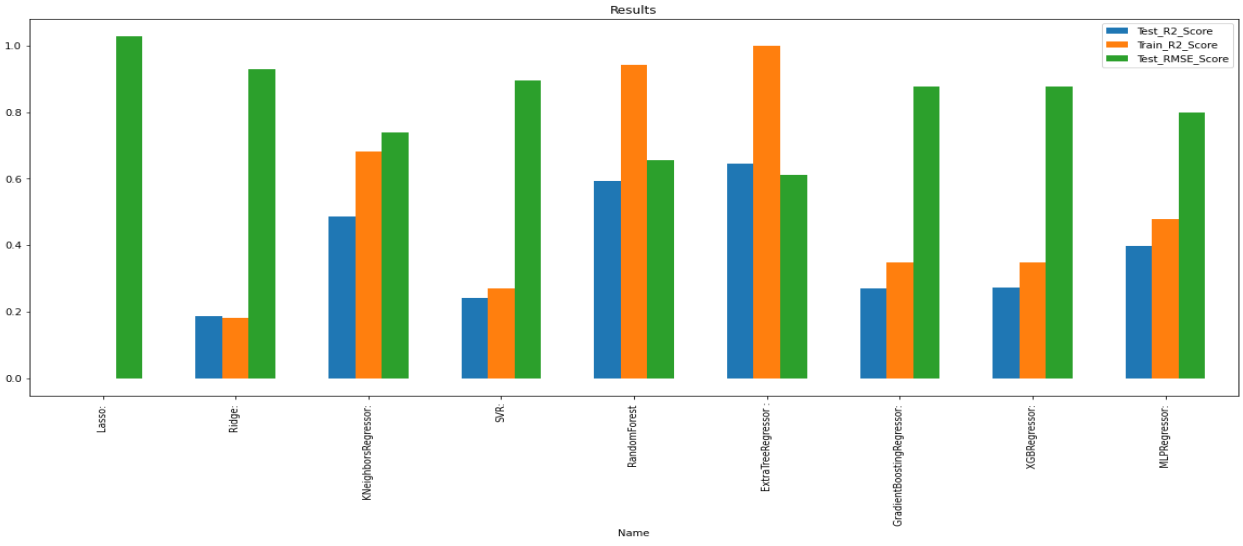
RMSE will be calculated by taking the square root of the MSE value calculated using *mean_square_error()* function.

Results

- **With PCA features**



- **Without PCA features**



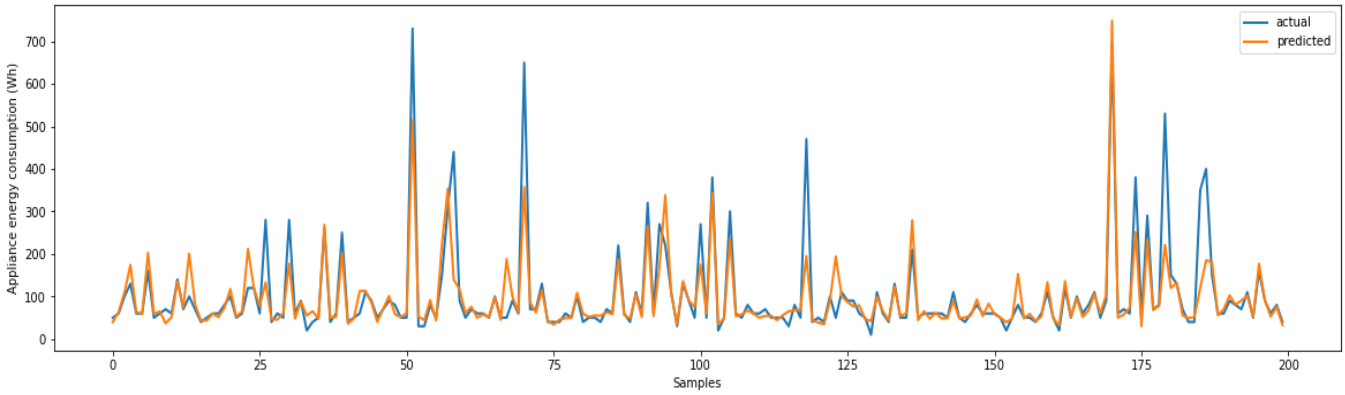
- Both feature sets with or without PCA features perform equally well on our selected tree-based model.

- Although, Extra trees regressor overfits our training data with a R2 score of 1, it also gives, by far, the best performance on the test set as well compared to other models, with a R2 score of 0.645.
- The test RMSE is also comparatively too low compared to other models.
- Hyper parameter tuning of our model doesn't have any significant impact on our test results. We were able to improve the test R2 score (0.694) results by less than 1%.

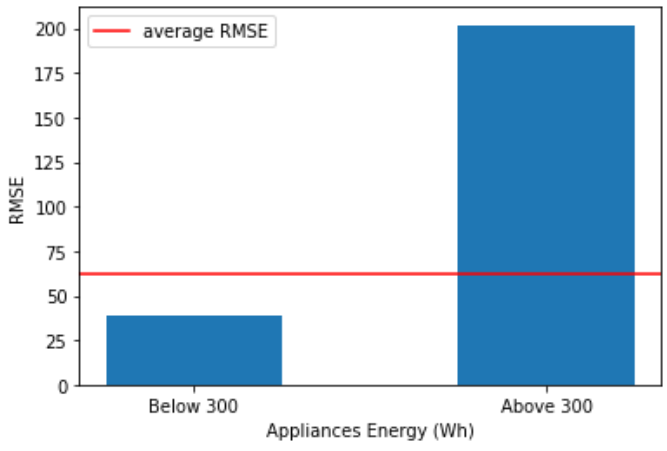
Evaluation metric results for best model

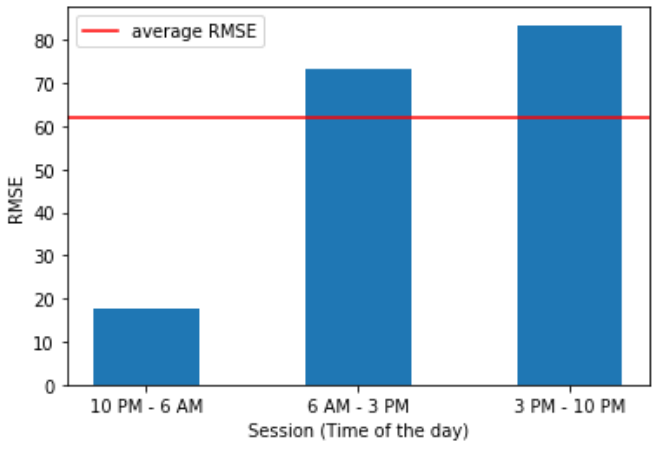
1) R2 score on training data: 1.0
2) R2 score on test data: 0.649
3) RMSE on test data = 0.601 (For calculating RMSE, the data was scaled so that comparison with other models is easier)

Model Analysis



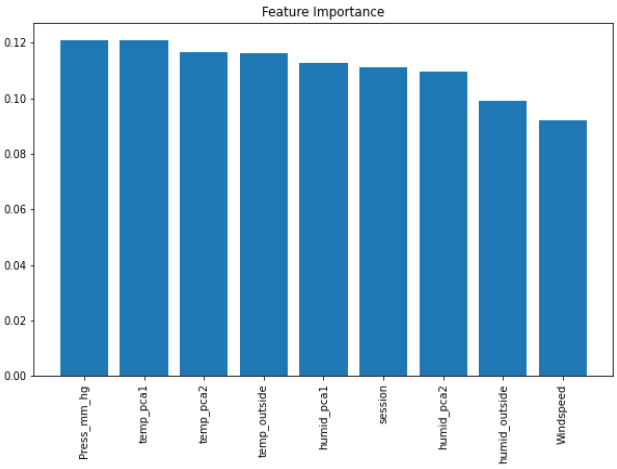
Above figure is a representation of actual and predicted values by our model for 200 samples in the test dataset. The model so far seems to perform quite good with a test R2 score of 0.649.

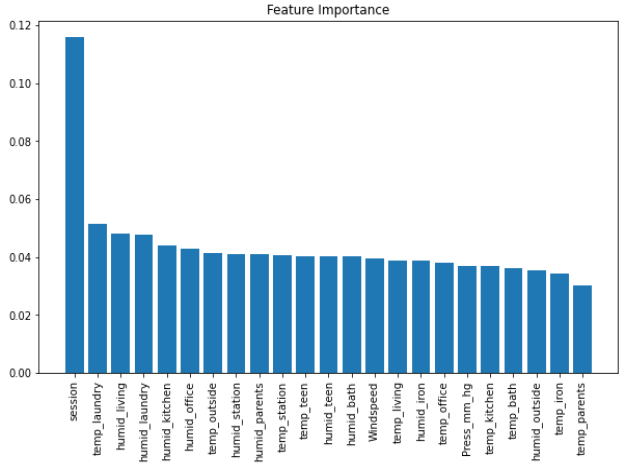




The above two figures reflect performance of our model under different scenarios. From the left figure, we clearly observe that the prediction errors are more prevalent in the samples with higher actual energy consumption values. From the right figure we clearly observe that the model performs better while predicting energy consumption of appliances running in between 10 PM - 6 AM, most certainly due to low variability of energy consumption in the range. However, the model shows higher RMSE for predictions made from 6 AM - 10 PM, most certainly due to high variability in appliances energy consumption in that interval.

Feature importances





Above two figures represent the importance of our features belonging to two different feature set (with and without PCA features) with respect to our best model i.e. Extra Trees Regressor. The figure on right shows feature importance in case of the feature set without PCA features in it. It is observed that the newly engineered feature 'session' holds a significantly higher importance compared to other features. The figure on left shows feature importances for the feature set with PCA features in it. We observe that, when we carefully create the temperature and humidity features from PCA that explains maximum variance in data, and include them in the feature set, the tree-based regression model tends to give almost similar importance to all the features including our engineered feature 'session'.

Conclusion

- The temperature/ humidity features showed little to no linear (pearson) correlation w.r.t target variable (<1%), although being highly linearly correlated within themselves.

- The time zone of the day plays an important role in deciding power consumption of appliances.

- The best Algorithm to use for this dataset is Extra Trees Regressor (tree based algorithm)

- PCA helped us to reduce our feature set dimension considerably without affecting performance of our models significantly.

- The untuned model was able to explain 64.5% of variance (R2 score = 0.645) on test set, while the tuned model was able to explain 64.9% of variance (R2 score = 0.649) on test set which is a tiny improvement of < 1 %

- The least RMSE score on the test data set is found to be around 0.6 by Extra trees regressor model, which is considerably good compared to other models.

- Tree based models are by far the best model while dealing with a data set that has most of its features having no linear correlation with the target variable. For similar reasons, linear models such as linear regression, Ridge and Lasso perform the worst.