Mehdi Salehi
Springboard Data Science Career Track
Capstone Two Final Report

Appliances’ Energy Consumption Prediction Based on House Temperature and Humidity Conditions and Meteorological Variables

Executive Summary

Appliances' Energy consumption data set along with a machine learning workflow was used to build and tune predictive models using three algorithms (Random Forest, Gradient Boosting, and Xtreme GB). Xtreme GB was the best model after hyperparameter tunning in terms of maximum accuracy and minimum MAE and resulted in an accuracy of 74% compared to the historical accuracy of 61% (a 12% improvement). The model can be used to predict appliances and households’ energy consumption in extreme conditions (hot or cold, dry to extreme humidity) and enable the power company to adjust its energy supply accordingly.

Problem Statement

These days we are very dependent on energy and all households require energy to power numerous home devices and equipment (heating and air conditioning, water heating, lighting, refrigeration, televisions, cooking appliances, clothes washers, consumer electronics including computers, tablets, smartphones, video game consoles, and internet streaming devices). Just imagine having no access to all of these for a day and you will realize the importance of a reliable energy supply. Many factors affect the amount of energy a household uses such as geographic location and climate, type of home and its physical characteristics, number, type, and efficiency of energy-consuming devices in the home, the amount of time they are used, and the number of household members.

In an effort to be able to provide reliable energy to households and also have a prediction of how household conditions affect the energy usage, a power company (XL) and an appliance manufacturer company (AP) joined forces to sponsor a study to determine how a consumer’s house environmental conditions and meteorological variables will affect appliances' energy consumption and to develop a predictive machine learning model to estimate appliance energy consumption from those attributes. The power company is looking at reducing load by predicting when/how customers’ appliances will draw more power and adjusting the supply based on environmental conditions. This study also offers the potential to enable significant insights and energy automation in buildings. AP company is looking to identify energy efficiency improvements, predict equipment failure, and maximize cost savings by offering appliances that use less power based on where the appliances will be utilized (environmental and geographical locations). By using the Appliances' Energy consumption data, I was able to build and tune predictive models using three algorithms (Random Forest, Gradient Boosting, and Xtreme GB) with an average precision of 0.73. The models can be used to predict appliances and households' energy consumption in extreme conditions (hot or cold, dry to extreme humidity) and enable the power company to adjust its energy supply accordingly.

Data Set Information

The data set was obtained from the UC Irvine Machine Learning Repository of appliances energy use in a low energy building (https://archive.ics.uci.edu/ml/machine-learning-databases/00374/) and is collected at 10 min intervals for about 4.5 months (January to April 2016). The house temperature and humidity conditions were monitored with a ZigBee wireless sensor network. Each wireless node transmitted the temperature and humidity conditions around 3.3 min. Then, the wireless data was averaged for 10 minutes periods. The energy data was logged every 10 minutes with m-bus energy meters. Weather from the nearest airport weather station (Chievres Airport, Belgium) was downloaded from a public data set from Reliable Prognosis and merged together with the experimental data sets using the date and time column.

Data Wrangling

This step of the Data Science Method focuses on collecting data, organizing it, cleaning it, and ensuring it's well defined. The raw data set contained 19735 rows (entries) with 29 attributes (columns). All the columns were numerical, except for the date column. There were also two random variables included. 'Appliances' is the electricity usage in Wh for appliances in the house (our target variable) and the other columns are potential features. Inspecting the data set revealed no missing or duplicate values. Columns were renamed for more readability based on the variable description file included with the data set. Two random variables were removed and the date column was converted into date type. Here is a snapshot of the cleaned data.

In [10]:
```
appliances_data.head().T
```

Out[10]:

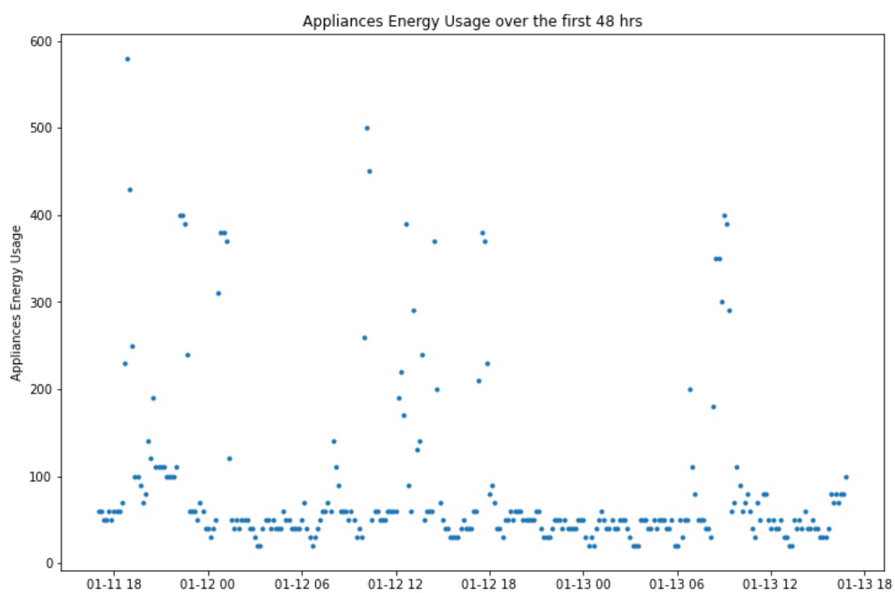
	0	1	2	3	4
date	2016-01-11 17:00:00	2016-01-11 17:10:00	2016-01-11 17:20:00	2016-01-11 17:30:00	2016-01-11 17:40:00
Appliances_E_Wh	60	60	50	50	60
Lights_E_Wh	30	30	30	40	40
T_Kitchen	19.89	19.89	19.89	19.89	19.89
H_Kitchen	47.596667	46.693333	46.3	46.066667	46.333333
T_Living	19.2	19.2	19.2	19.2	19.2
H_Living	44.79	44.7225	44.626667	44.59	44.53
T_Laundry	19.79	19.79	19.79	19.79	19.79
H_Laundry	44.73	44.79	44.933333	45.0	45.0
T_Office	19.0	19.0	18.926667	18.89	18.89
H_Office	45.566667	45.9925	45.89	45.723333	45.53
T_Bathroom	17.166667	17.166667	17.166667	17.166667	17.2
H_Bathroom	55.2	55.2	55.09	55.09	55.09
T_Building_out_NS	7.026667	6.833333	6.56	6.433333	6.366667
H_Building_out_NS	84.256667	84.063333	83.156667	83.423333	84.893333
T_Ironning	17.2	17.2	17.2	17.133333	17.2
H_Ironing	41.626667	41.56	41.433333	41.29	41.23
T_Teenager	18.2	18.2	18.2	18.1	18.1
H_Teenager	48.9	48.863333	48.73	48.59	48.59

Exploratory Data Analysis

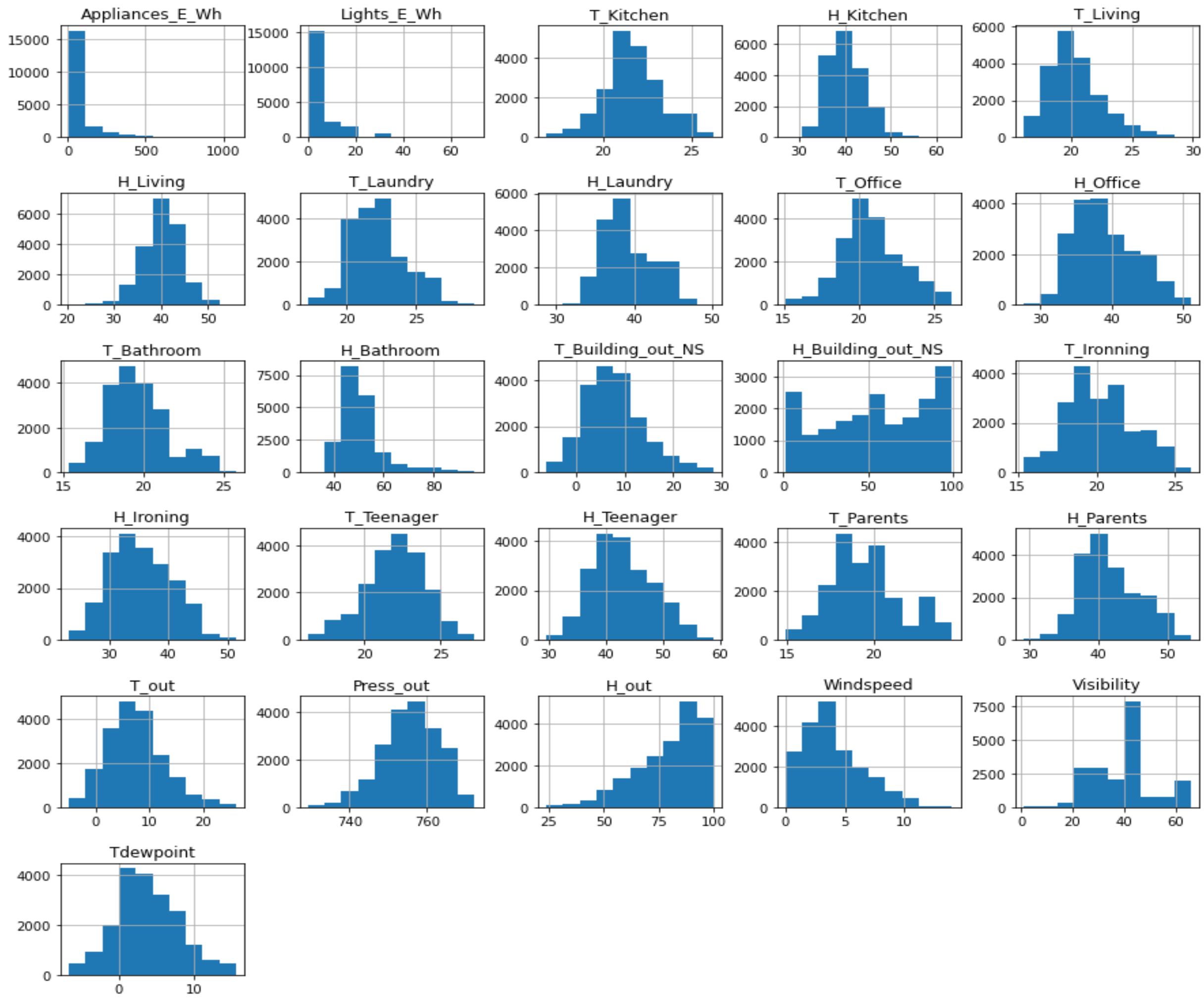
This step of the Data Science Method process focuses on EDA analysis with the goal of getting familiar with the features in our dataset, investigating the relationships between features, and generally understanding the core characteristics of the dataset. We will continue to clean, transform, and visualize data and correlations.

The first step was to look at the statistical summary of the numerical columns using the describe() function. The summary revealed that Appliances_E_Wh values range from 10 Wh to 1080 Wh with a mean of 97.7 and a standard deviation of 102.5. Appliances_E_Wh std is higher than the mean value! The same is true for Lights_E_Wh! These two seem to have abnormal distributions.
The inside temperature of the house ranges from 14.9 to 29.9ºC. The humidity of the inside ranges from 20.5 to 63.4%, with the exception of the Bathroomwith 96% humidity (which is expected). Outside temperature ranges from -6 to 28.3ºC. Outside humidity is ranging from 24.0 to 100%. The H_Building_out_NS humidity (humidity for the north side of the building) has a minimum value of 1 % which seems very low (H_out from a nearby weather station shows a minimum humidity of 14.9.%). Since this is for the north side of the building where it gets more sunshine, humidity could get that low! Appliances_E_Wh std is higher than the mean value! The same is true for Lights_E_Wh! These two seem to have abnormal distributions.

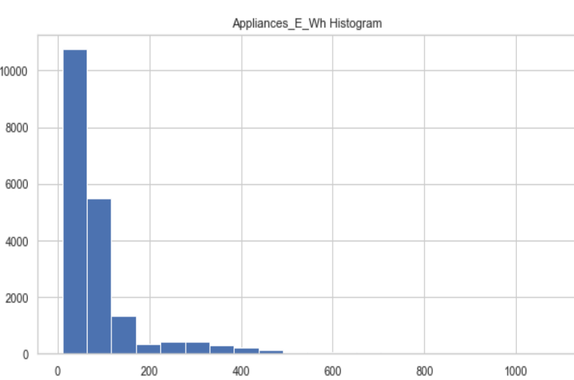
I also took an at the variation of Appliances_E_Wh over time. The appliances' energy consumption is cyclic and there are peak times (mainly in the evening) when the usage is higher during the day.

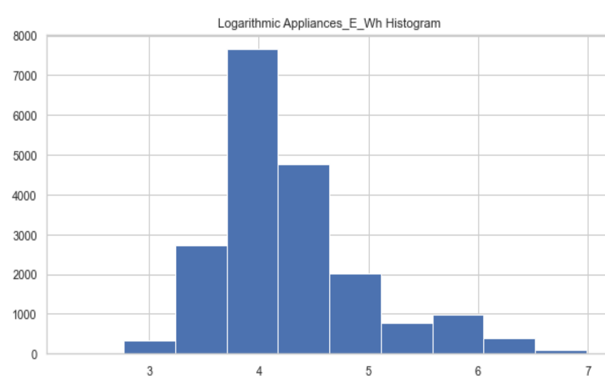


The next step was to create histograms of the numerical columns. Appliances_E_Wh and Lights_E_Wh are heavily skewed to the left. H_Building_out_NS has a strange distribution and that for H_out is skewed to the right. Visibility also looks abnormal.

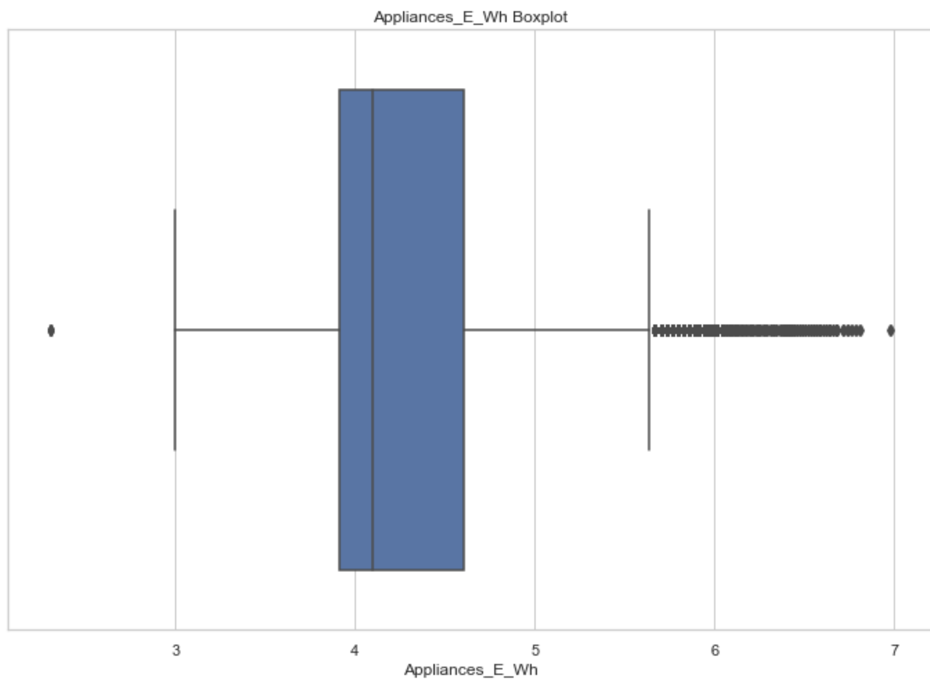


Taking a closer look at the target variable (Appliances_E_Wh) shows that by performing a log transformation on the data the distribution would look more normal. This change was performed permanently on the target variable in the data set.





The figure below shows the boxplot for the log-transformed target variable.



The next step was to look at the relationships of the target variable (Appliance_E_Wh) to other features of the data set. Scatter plots of Appliance_E_Wh vs. all the features were created and then we looked at the correlation coefficients using the .corr() correlation matrix.

```
df_corr = df.corr()['Appliances_E_Wh'].sort_values().abs().sort_values(ascending=False)  # Correlation matrix
df_corr
```

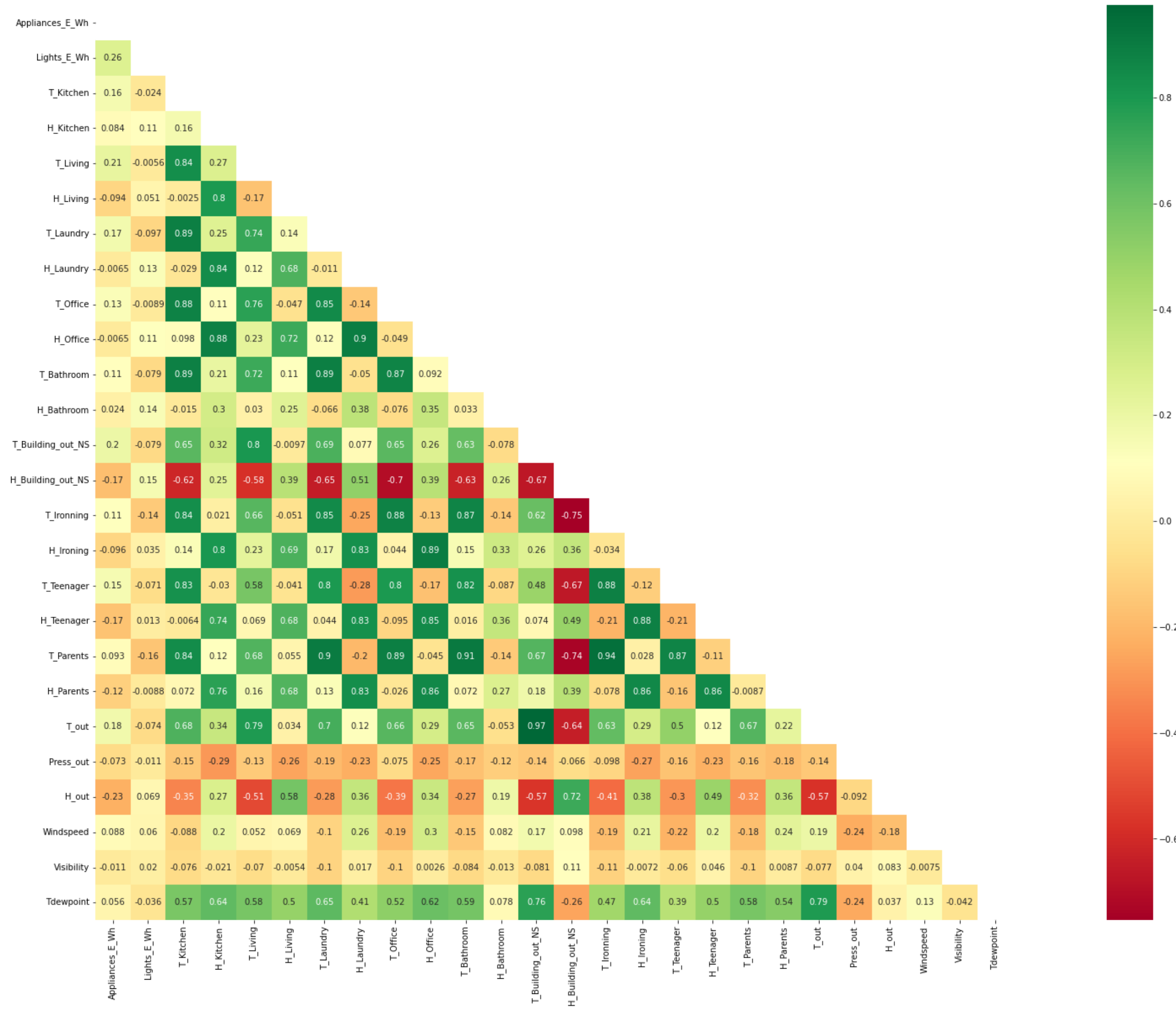
The following table summarizes the absolute values of the correlation coefficient of each feature with respect to the target (Appliance_E_Wh).

```
Appliances_E_Wh       1.000000
Lights_E_Wh           0.261442
H_out                 0.226185
T_Living              0.214756
T_Building_out_NS     0.196546
T_out                 0.176161
H_Building_out_NS     0.174133
T_Laundry             0.167221
H_Teenager            0.165397
T_Kitchen             0.160747
T_Teenager            0.153917
T_Office              0.132359
H_Parents             0.115582
T_Ironning            0.110415
T_Bathroom            0.110099
H_Ironing             0.096231
H_Living              0.093674
T_Parents             0.092553
Windspeed             0.087722
H_Kitchen             0.084457
Press_out             0.072632
Tdewpoint             0.056241
H_Bathroom            0.024312
Visibility            0.010970
H_Office              0.006533
H_Laundry             0.006462
Name: Appliances_E_Wh, dtype: float64
```

Lights_E_Wh, H_out, T_Living, T_Building_out_NS, T_out, and H_Building_out_NS are the top six most contributing factors according to the correlation matrix.

The following figure shows the heatmap of the correlation matrix. Appliances_E_Wh is generally positively correlated with temperature and negatively with humidity. Appliances_E_Wh is showing the highest positive correlation with Lights_E_Wh, T_Living, and T_Building_out_NS. It is negatively correlated with H_out, H_Building_out_NS, and H_Teenager.

Temperatures are strongly and negatively correlated with H_Building_out_NS and to less extent with H_out.

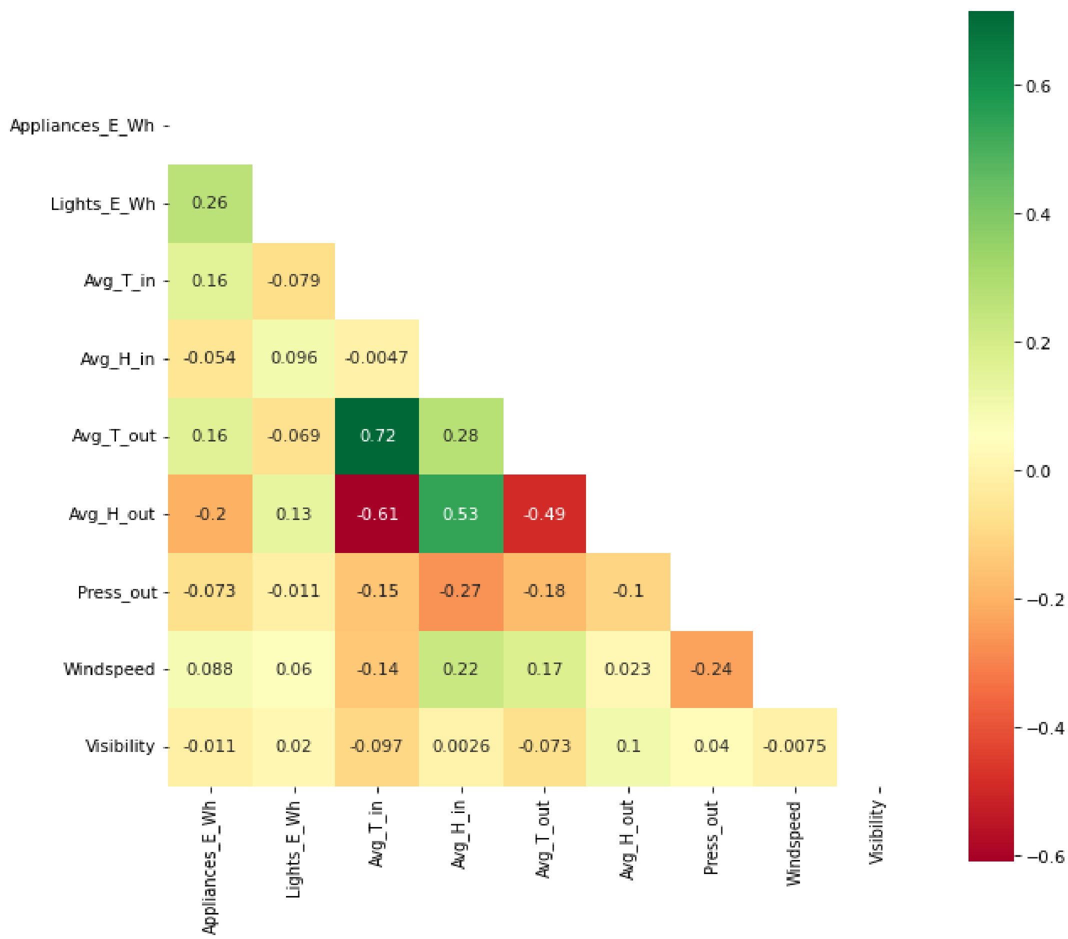


I also performed some feature engineering by averaging the inside and outside temperature and humidity values and confirmed that these average features exhibit similar correlations with the target variable as the individual ones. The table below shows the new data frame with a reduced number of features.

```
df_FE.head().T
```

	0	1	2	3	4
date	2016-01-11 17:00:00	2016-01-11 17:10:00	2016-01-11 17:20:00	2016-01-11 17:30:00	2016-01-11 17:40:00
Appliances_E_Wh	4.094345	4.094345	3.912023	3.912023	4.094345
Lights_E_Wh	30	30	30	40	40
Avg_T_in	18.435	18.439167	18.421667	18.39625	18.40875
Avg_H_in	46.7425	46.672708	46.562917	46.46875	46.462917
Avg_T_out	6.308889	6.172222	6.008889	5.894444	5.8
Avg_H_out	60.518889	60.421111	60.085556	60.141111	60.597778
Press_out	733.5	733.6	733.7	733.8	733.9
Windspeed	7.0	6.666667	6.333333	6.0	5.666667
Visibility	63.0	59.166667	55.333333	51.5	47.666667

The heat plot shows that the target variable is positively correlated with Lights_Wh and average temperatures and negatively with humidity, as we have seen with the original data set.



Preprocessing and Training Data Development

This step focuses on Pre-processing & Training Data Development. The goal of this step is to normalize and standardize all the features in your data, as well as create a validation set.

The first step was to define our X (features) and y (target variable) and then create a 70/30 train and test split.

```
y = df['Appliances_E_Wh']

X = df.drop(["Appliances_E_Wh", "date"],axis=1)  # considering all the variables except the date

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.30, random_state = 42)
```

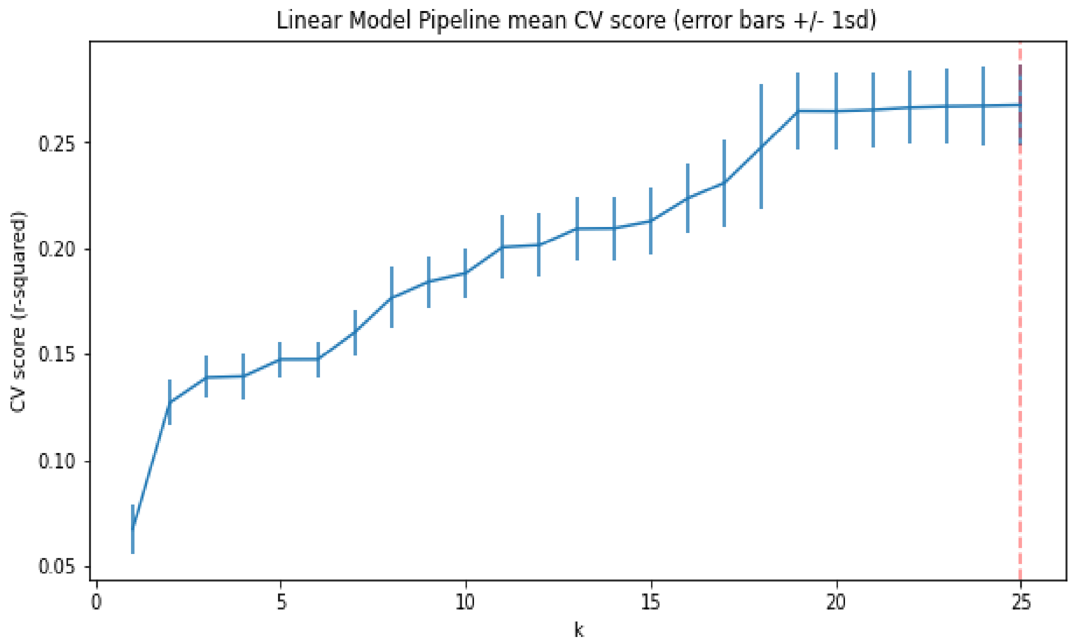
At this stage, features were also standardized by scaling the values. Note: We need to fit() our scaler on X_train and then use that fitted scaler to transform() X_test. This is to avoid data leakage while we standardize our data.

```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

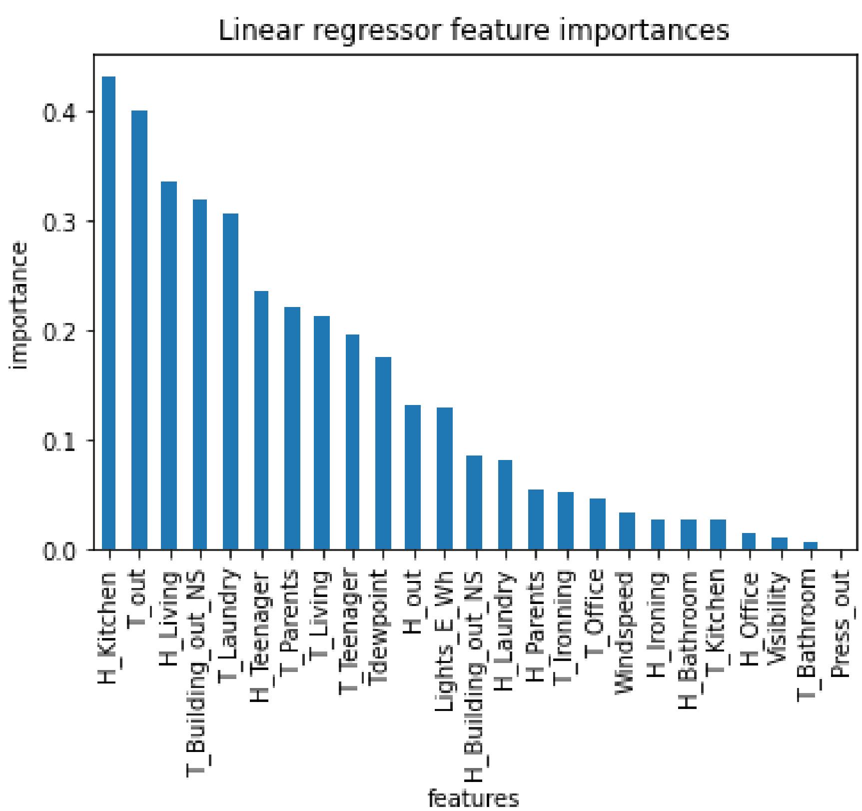
Using a Linear Regression model (from sklearn.linear_model) with scaled data, an R-Squared coefficient of 0.28 was obtained on the test data set, which means the linear model explains only about 28% of the variation from the mean. There's more work to do since the linear model can not capture the trends in the data set. This value was also confirmed using the OLS Linear model from Statsmodel. Using scaled or non-scaled data made no difference in model performance, so going forward, all the models will use scaled data.

Investigating optimum number of features using linear model

Using a pipeline with a linear model combined with StandardSscaler and SelectBest (make_pipeline(StandardScaler(), SelectKBest(f_regression), LinearRegression()), I also looked at the sensitivity of the model to the number of features to see how reducing the number of features would affect the model performance (using cross-validation). The figure below shows that the best performance is obtained when all the 25 features are used during the training (Red dashed line is drawing best_k = lm_grid_cv.best_params_['selectkbest__k']).

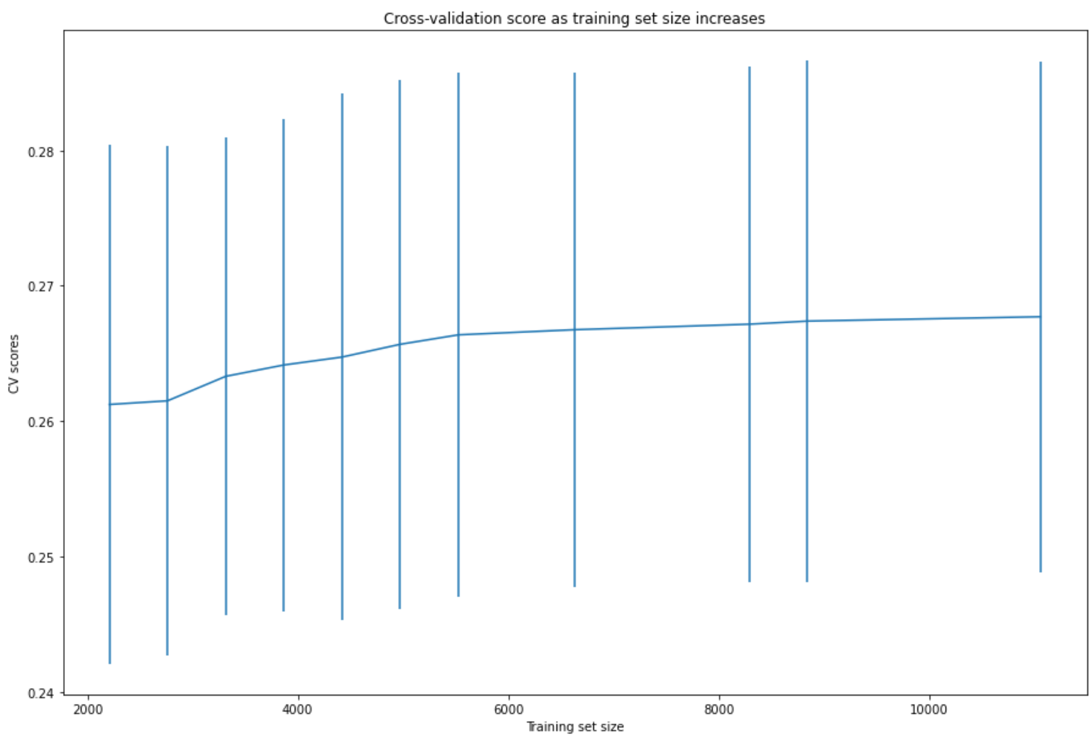


Using a GridSearch and varying the number of features, the same result was obtained. Using the best_estimator function, the following figure shows the importance of the features obtained from the linear model. H_Kitchen, T_out, and H_Living were the top three important features.



Data Quantity Assessment

We also need to know if we have enough data points in our training set or need to increase or undertake further data collection. Would more data be useful? We're often led to believe more data is always good, but gathering data invariably has a cost associated with it. We can examine this trade-off by looking at how performance varies with different data set sizes. The learning_curve function can be used to perform this task conveniently.



This figure above shows that there is an initial improvement in model scores as the sample size increases and as one would expect, but model performance essentially levels off by around a sample size of 8840. So, we have plenty of data (our training data set includes 13814 entries).

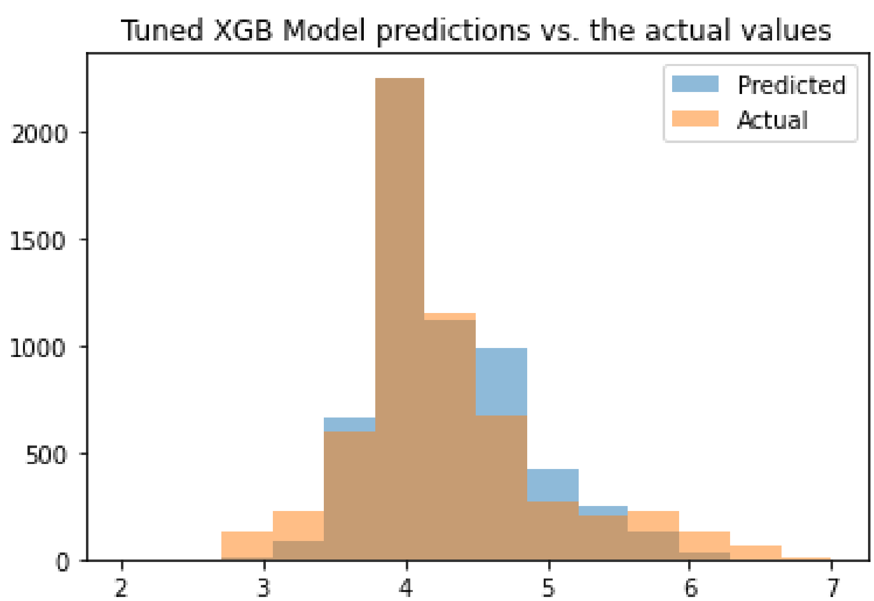
At this stage, a base random forest model was used to assess the data. The base model resulted in an R-Squared coefficient of 0.67 on the test data set, more than double the value obtained from the linear model. In the next section, we will look at the modeling process in more detail.

Modeling and Optimization

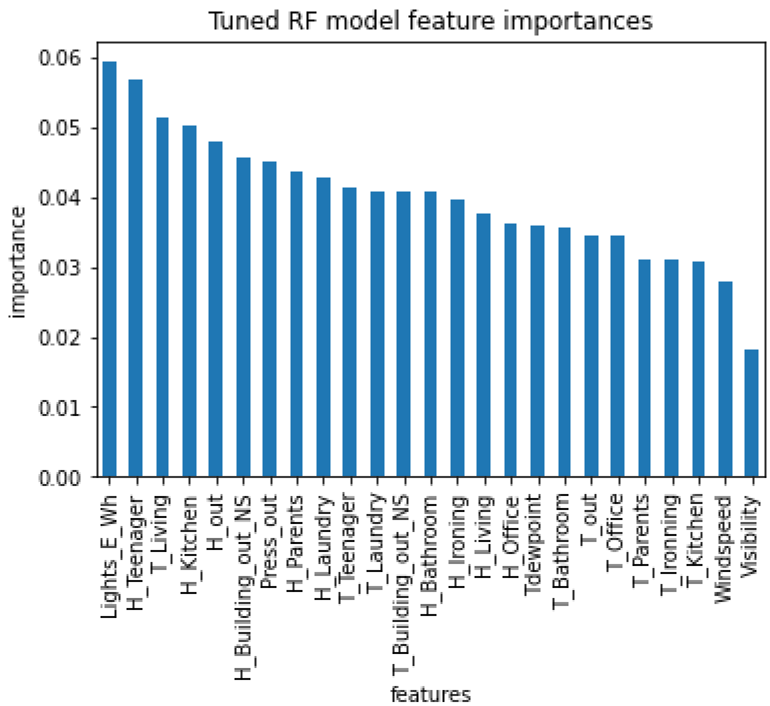
This step focuses on creating and testing different models for our data, as well as model performance evaluation. Hyperparameter tuning is also performed to optimize the models and then re-evaluate the model's performances using the tuned models. For hyperparameter running, the randomized grid search was used. I considered four models for use in this step of the process: Linear, RandomForest, GradientBoosting, and XtremeGB. The table below summarizes the models and their performances. It can be seen that tuned XtremeGB and RandomForest are the best models of all, resulting in the highest R-squared value and lowest MAE and MAPE values. Linear Regression is the worst and Tuned XGB is the best model.

Model	R-Square (Test Set)	MAE	MAPE (%)
Linear Regression	0.280	0.395	8.9
GradientBoosting (GB)	0.379	0.359	8.2
Xtreme GB	0.618	0.274	6.3
RandomForest (RF)	0.670	0.249	5.6
Tuned GB	0.714	0.235	5.4
Tuned XGB	0.728	0.226	5.2
Tuned RF	0.724	0.227	5.2

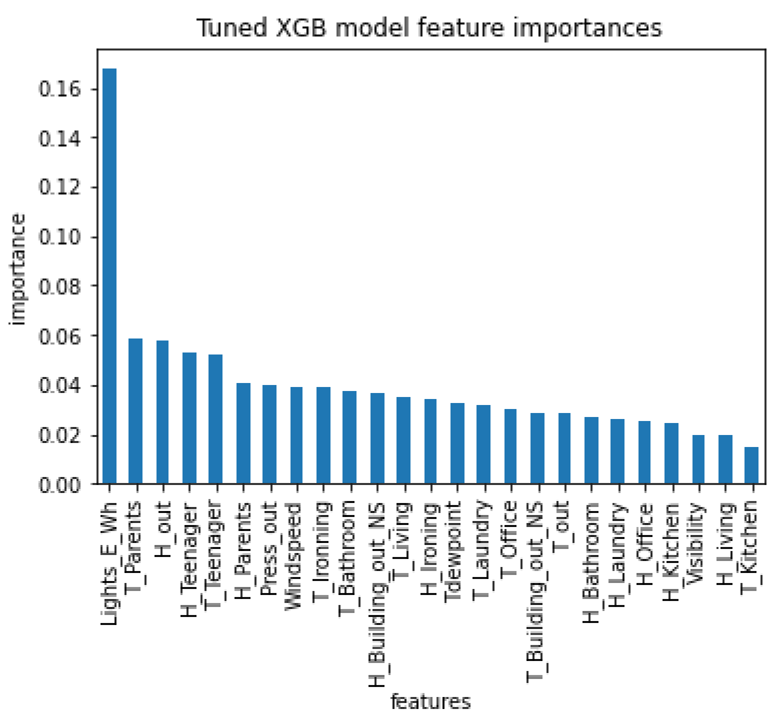
Figures below compare the predicted values vs. the actual values for the test set using the XtremeGB tunes model. The models overall predict the values very accurately with the prediction values having a narrower range and overestimating in some instances.



The following figures show the feature importance obtained from tuned RF and XGB models.



The top six features of the tuned RF model are: Lights_E_Wh, H_Teenager, T_Living, H_Kitchen, H_out, and H_Building_out_NS.



The top six features of the tuned XGB model are: Lights_E_Wh, T_Parents, H_out, H_Teenager, T_Teenager, and H_Parents. The common and top features in a household affecting the appliance energy consumption are: Lights_E_Wh, H_Teenager, and H_out.

The models can be used to simulate and forecast the energy consumption of appliances in many situations considering the interior and exterior conditions of a household. For example, how installing a humidifier or dehumidifier in a house will affect energy consumption. They can also be used in combination with the weather forecasts to predict the possible increases or decreases in energy loads, especially when considering more than one individual household in a neighborhood or zip code, or even a city to scale the process.

Here are some ideas to improve the model in the future:

- Expand the study over a few households with different orientations of the house, appliances manufacturers, and number of people
- Consider approximately to the weather station (airport)
- Extend the study over a longer period of time to capture seasonality effects

Conclusions

A predictive model was developed that predicts the energy consumption with an accuracy of of 74% compared to the historical accuracy of 61% (a 12% improvement). Out of 7 supervised regression models, the tuned Extreme Gradient Boosting provided the best results. All 25 features were used in the modeling and with a 70%-30% splitting, the test data set resulted in an MAE of 0.226. The top features in the subject household affecting the appliance energy consumption are: Lights_E_Wh, H_Teenager, and H_out.