Siddhartha Shankar
https://github.com/sid0712
https://www.linkedin.com/in/sid-business/
siddhartha.shankar@icloud.com, 469-509-4807

Problem Statement

Task I: Developing and Implementing linear and logistics regression using gradient descent algorithm and tuning various parameters to predict the energy usage of appliances based on temperature, humidity and weather attributes.
Task II: Implementing supervised learning models using Logistic Regression, Scalar Vector Classifier (SVC), Decision Tree Classifier and AdaBoost Classifier to classify high and low energy household usage of appliances based on temperature, humidity and weather attributes and come up with the best model on the basis of model evaluation parameters of classifiers. The Dataset can be downloaded from UCI_ML_Repository.

Description of Data:

- Dataset consists of 19735 observations and 29 variables. Description can be found by visiting the link.
- Dependent variable (y) is Appliances, energy use in Wh. The main features are temperature, humidity and pressure. The features were monitored with a sensor and averaged for 10 minutes periods.
- There are no missing values in the data. Hence, no imputation is required.

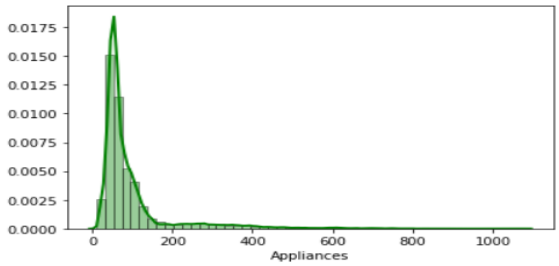
Exploratory Data Analysis:

Preparing the Dataset

Feature Range	
Temperature	-6 to 30 deg
Humidity	1 to 100 %
Windspeed	0 to 14 m/s
Visibility	1 to 66 km
Pressure	729 to 772 mm Hg
Appliance Energy Usage	10 to 1080 Wh

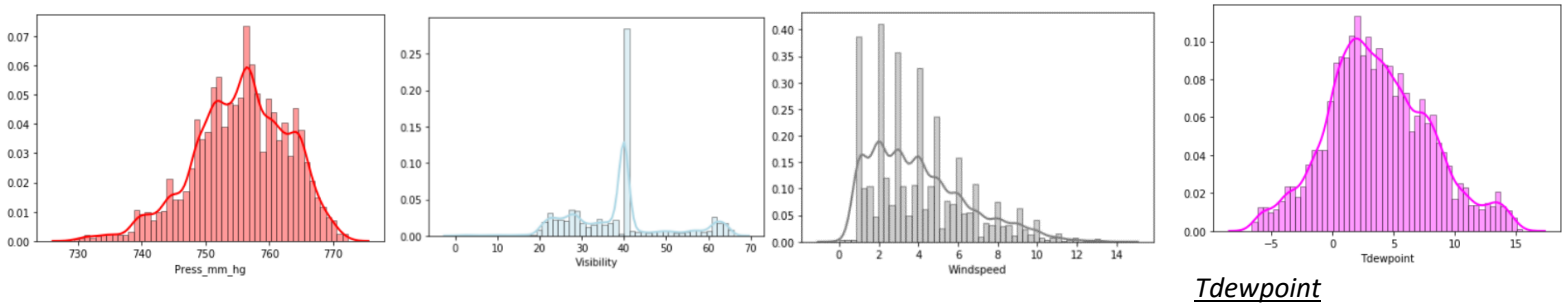
- The feature 'Light' is removed as we are interested in estimating the energy use of appliances.
- The random variables 'rv1' and 'rv2' are removed as both have approx. same values for all the records.
- The 'Date' column is removed as it provides the consumption information on a given date.
- 'T6' & 'T_Out' both shows outside temperature. Hence, 'T6' is removed.

Distribution plot of dependent feature: 'Appliances'



'Appliances' is *skewed right* which shows that there are a greater number of low energy consumption records than high energy consumption records.

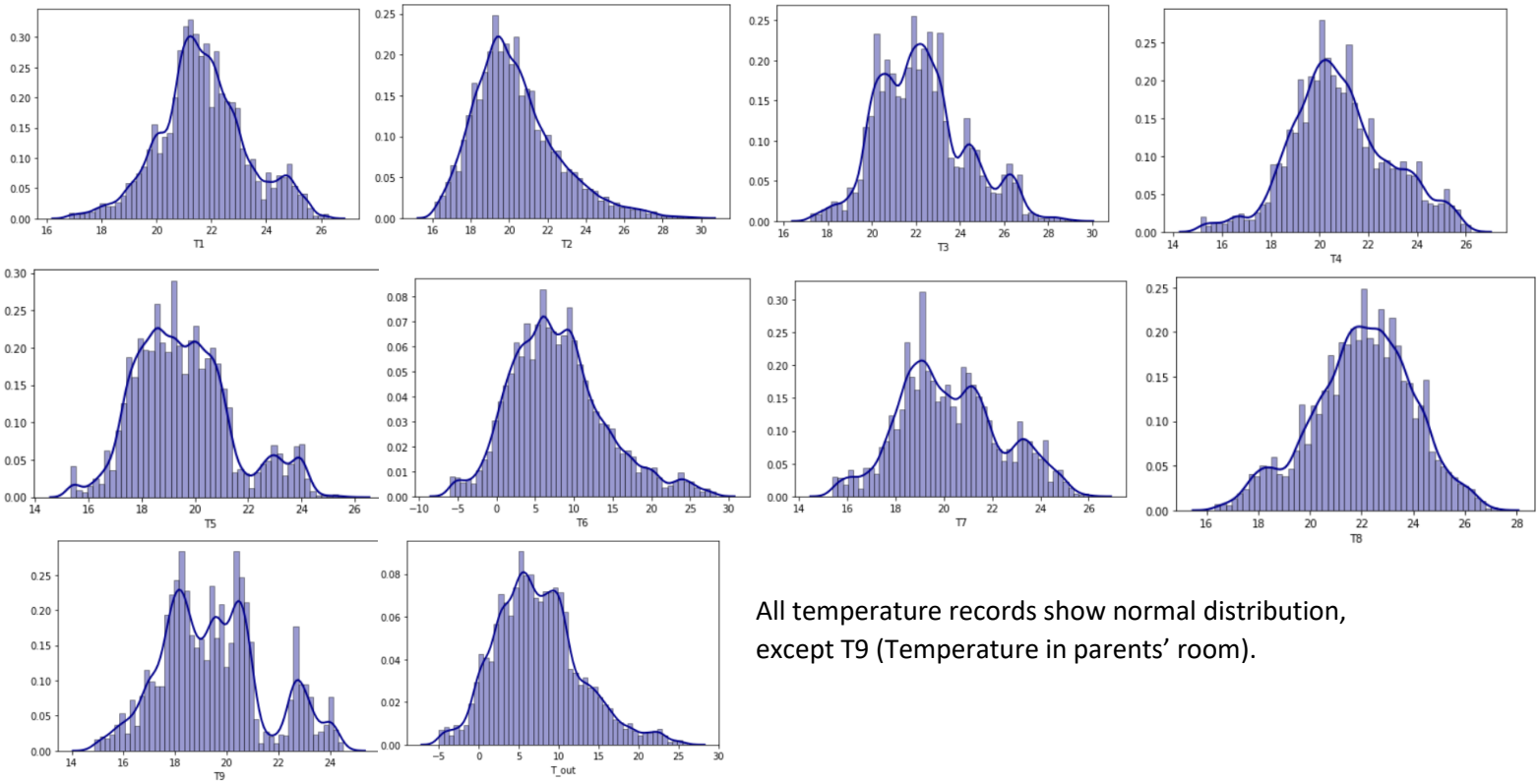
Distribution plots of Visibility, Windspeed, Pressure and Tdewpoint



'Visibility' and 'Windspeed' are skewed. However, Pressure and Dewpoint follow Normal distribution

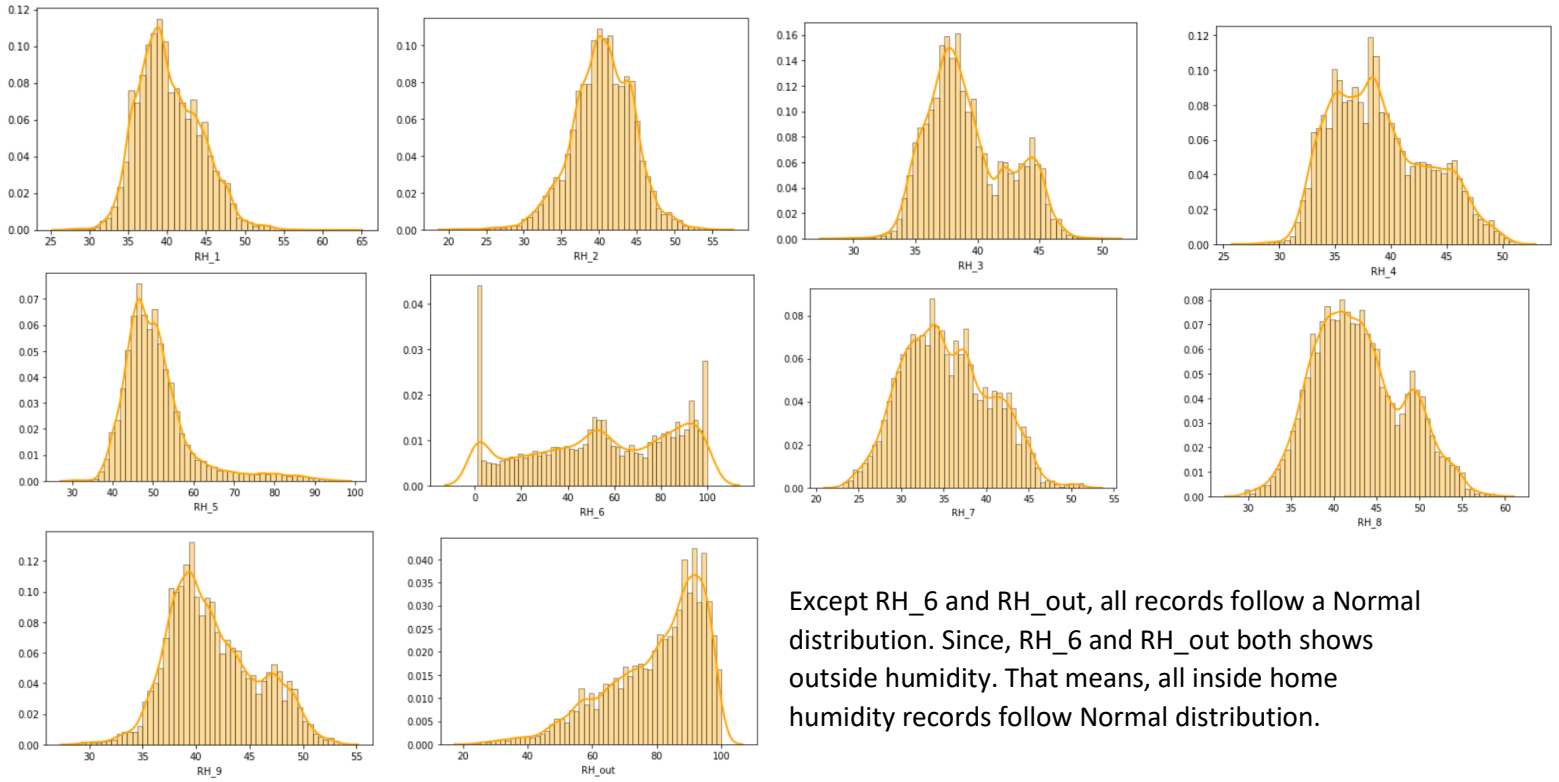
Siddhartha Shankar
https://github.com/sid0712
https://www.linkedin.com/in/sid-business/
siddhartha.shankar@icloud.com, 469-509-4807

Distribution Plots of Independent feature: Temperature



All temperature records show normal distribution, except T9 (Temperature in parents' room).

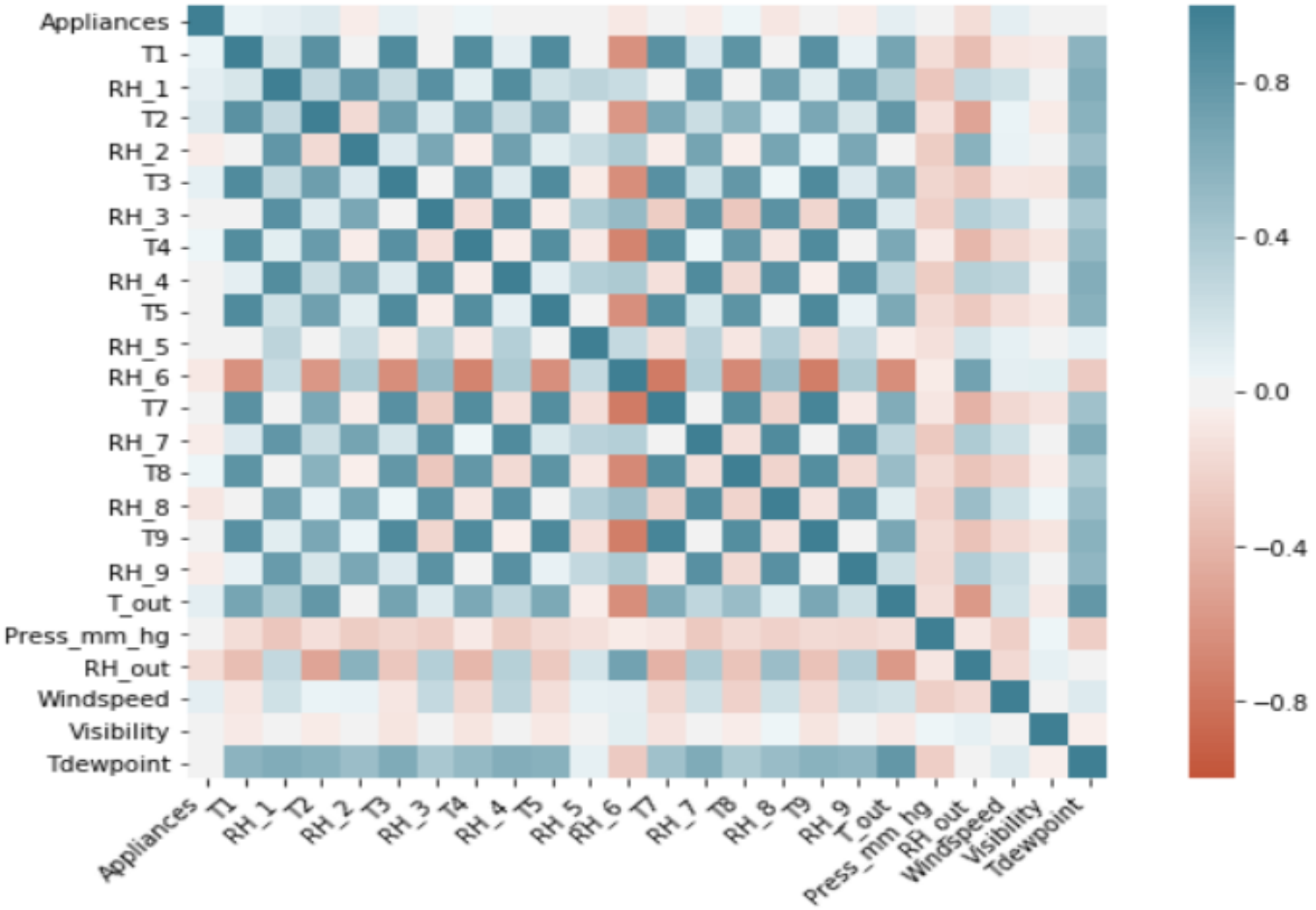
Distribution Plots of Independent feature: Humidity



Except RH_6 and RH_out, all records follow a Normal distribution. Since, RH_6 and RH_out both shows outside humidity. That means, all inside home humidity records follow Normal distribution.

Siddhartha Shankar
https://github.com/sid0712
https://www.linkedin.com/in/sid-business/
siddhartha.shankar@icloud.com, 469-509-4807

Correlation Matrix:



Observations from Correlation matrix:

- Temperature - Temperature variables from 'T1-T9' and 'T_out' are positively correlated with the target variable 'Appliances' . Correlation between temperatures inside the rooms are high, as expected. 'T9' is highly correlated with T3,T5,T7,T8. Hence, 'T9' is removed from the dataset.

- Humidity — There are no significantly high correlation records (> 0.9) for humidity sensors.

- Weather attributes — Visibility, Tdewpoint, Press_mm_hg have low correlation values with target variable.

Based on the above observations, 'date', 'lights', 'T6', 'RH_6', 'T8', 'RH_8', 'T9', 'RH_9', 'rv1', 'rv2', 'Visibility' columns are removed from the dataset.

Hence, there are no features having a linear relationship with the target variable.

Siddhartha Shankar
https://github.com/sid0712
https://www.linkedin.com/in/sid-business/
siddhartha.shankar@icloud.com, 469-509-4807

Goal

Implementing a linear regression model on the dataset to predict the energy usage of appliances.

Algorithm Preparation and Parameters tuning

Task 1: **Loaded the dataset and partitioned it randomly into train and test set using a 70/30 split.**
We build our model over training datasets and test its accuracy over Test data. Before splitting the dataset into 70:30 ratio, dataset was scaled as per = $(X - X_{mean}) / (X_{max} - X_{min})$ with sklearn module. No. of Training examples = 13814 and No. of Test data points = 5921.

Task 2: **Designing a Supervised Learning model using linear and logistics regression models to estimate the energy usage of appliances with 17 input variables**

The Linear Regression Model I:

Appliances, y = $\beta 0 + \beta_1$*T1 + β_2*RH_1 + β_3*T2 + β_4*RH_2 + β_5*T3 + β_6*RH_3 + β_7*T4 + β_8*RH_4 + β_9*T5 + β_{10}*RH_5 + β_{11}*T7 + β_{12}*RH_7 + β_{13}*T_out + β_{14}*Press_mm_hg + β_{15}*RH_out + β_{16}*Windspeed + β_{17}*Tdewpoint

MODEL 1: Original variable selected model

Dependent Variable, y is Appliances, (energy use in Wh)
Independent input variables (x_i) are:
T1, Temperature in kitchen area, in Celsius
RH_1, Humidity in kitchen area, in %
T2, Temperature in living room area, in Celsius
RH_2, Humidity in living room area, in %
T3, Temperature in laundry room area
RH_3, Humidity in laundry room area, in %
T4, Temperature in office room, in Celsius
RH_4, Humidity in office room, in %
T5, Temperature in bathroom, in Celsius
RH_5, Humidity in bathroom, in %
T7, Temperature in ironing room , in Celsius
RH_7, Humidity in ironing room, in %
To, Temperature outside (from Chievres weather station), in Celsius
Pressure (from Chievres weather station), in mm Hg
RH_out, Humidity outside (from Chievres weather station), in %
Wind speed (from Chievres weather station), in m/s
Tdewpoint (from Chievres weather station), Â°C

Task 3: **Implementing the gradient descent algorithm with batch update rule using the cost function of sum of squared errors.**

Cost Function: $$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

The gradient descent algorithm is:

repeat until convergence:

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

Algorithm is designed based on cost function of sum of squared errors mentioned above.
Initial parameter values:
Theta = 0
Iterations = 2000

Siddhartha Shankar
https://github.com/sid0712
https://www.linkedin.com/in/sid-business/
siddhartha.shankar@icloud.com, 469-509-4807

Convergence threshold = 0.0000001

Optimal theta values =

```
[[-0.44138156 -0.05990599 -0.62022486  1.05075211  0.68866234 -0.52511559
   0.79322695 -1.39215709 -0.59693255 -0.40380169  2.34266116  1.76617572
   0.91376956  0.91109303  0.59657632 -0.23284555 -0.11054334  1.02651376]]
```

Task 4: **Converting this problem into a binary classification problem. The target variable is converted into two categories : High consumption(1) and Low Consumption(0). Implemented logistic regression to carry out classification on this data set and accuracy/error metrics for train and test sets is reported.**

Target Variable: 'Appliances' is converted into categorical variables based on energy consumption.
If the energy consumption of 'Appliances' is less than or qual to 100 Wh, it is assigned value 0; else 1.

	Appliances
count	19735.000000
mean	0.214492
std	0.410480
min	0.000000
25%	0.000000
50%	0.000000
75%	0.000000
max	1.000000

Mean of 'Appliances' suggests that 21.45% of the data points have value assigned as 1 i.e. high consumption.

Before splitting the dataset into 70:30 ratio, dataset is scaled as per = $(X - X_{mean})$ / (standard deviation) with sklearn module

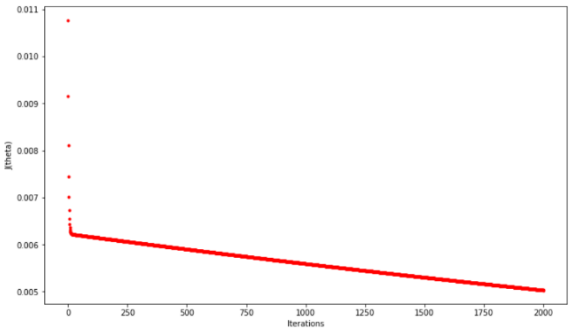
Experimentation 1

Experimenting with parameters for linear regression (e.g. learning rate ∝, iterations, convergence threshold) and the error/accuracy for train and test sets is plotted. Optimal parameter values are reported.

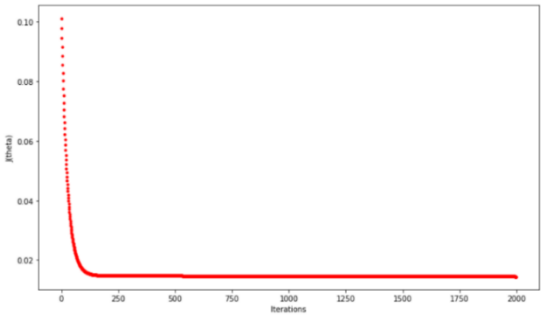
Parameters input values:

Theta = 0
Iterations = 2000
Convergence threshold = 0.0000001
Learning rate, ∝ = 0.1, 0.01, 0.003, 0.001

Learning rate,∝	cost	Iterations	MSE Train	MSE Test
0.001	0.00663	1487	172.45911	78.52625
0.003	0.00661	615	171.97546	78.2728
0.01	0.0066	258	171.71378	78.12241
0.1	0.00529	2000	139.03518	62.66538

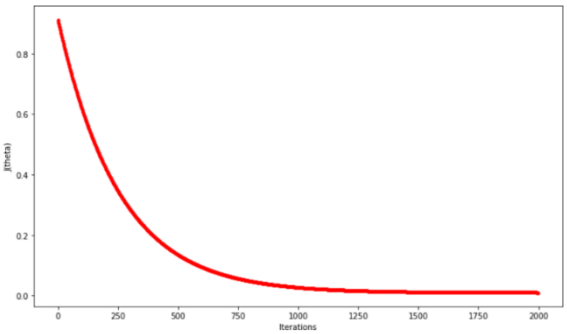


∝=0.1

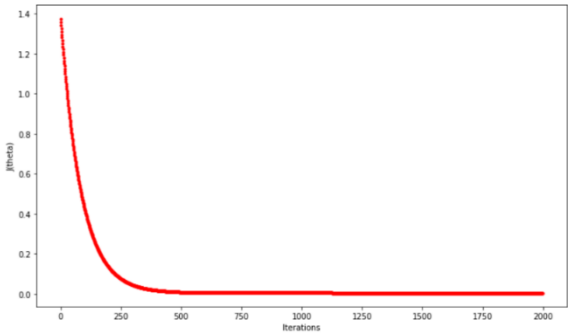


∝= 0.01

Siddhartha Shankar
https://github.com/sid0712
https://www.linkedin.com/in/sid-business/
siddhartha.shankar@icloud.com, 469-509-4807



$\propto = 0.001$



$\propto = 0.003$

Observations based on the graph between J(theta) and Iterations and the table above:

Larger value of ∝ e.g. 0.1 results in the sudden drop in the curve which suggests that it might miss the optimal minima and jump up. Additionally, no. of iterations is relatively larger for ∝ = 0.1. However, the error and cost function are the least in this case, which indicates we need to change convergence threshold value.

As the value of ∝ decreases, the curve slowly moves towards zero rather than suddenly dropping as the cost function approaches its minimum value. Hence, learning rate (∝ = 0.001) is the optimal value provided fixed convergence threshold of 0.0000001 and Iterations = 2000.

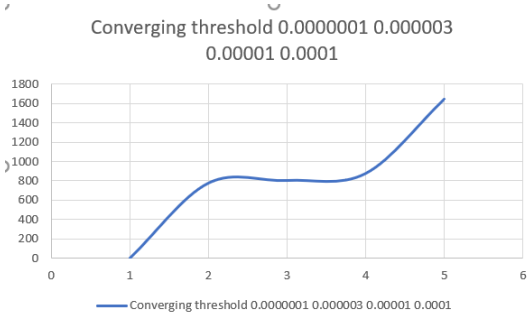
Hence, Optimal values of thetas for learning rate (∝ = 0.001) are :

```
[[0.05967137 0.00153778 0.0028795  0.00148045 0.00276893 0.00159391
  0.00276011 0.00147768 0.00273767 0.00138088 0.00358397 0.00143173
  0.00241155 0.00064471 0.05263513 0.00506897 0.00032471 0.00028761]]
```

Experimentation 2

Experimenting with thresholds for convergence for linear regression and describing error results for train and test sets as a function of threshold.
For Learning rate, (∝ = 0.001)

Converging threshold	cost	Iterations	MSE Train	MSE Test
0.0000001	0.006631	1487	172.45911	78.52625
0.000003	0.007387	572	192.34223	87.48112
0.00001	0.00917	263	240.75897	108.6567
0.0001	0.01373	1	365.23942	162.6253



As per the above table, it is evident that as the value of converging threshold decreases no. of iterations increases. Also, Mean Square Error for train and test data decreases with decrease in converging threshold value. Additionally, cost function is minimum for the lowest converging threshold.

Thus, converging threshold = 0.0000001 is the optimal value as it gives the lowest value of MSE.

Optimal values of thetas for learning rate (∝ = 0.001), converging threshold = 0.0000001 are :

Siddhartha Shankar
https://github.com/sid0712
https://www.linkedin.com/in/sid-business/
siddhartha.shankar@icloud.com, 469-509-4807

```
[[0.05967137 0.00153778 0.0028795  0.00148045 0.00276893 0.00159391
  0.00276011 0.00147768 0.00273767 0.00138088 0.00358397 0.00143173
  0.00241155 0.00064471 0.05263513 0.00506897 0.00032471 0.00028761]]
```

Experimentation 3

Picking ten features randomly and retraining your models only on these ten features. Comparing train and test error results with the original model.

The Linear Regression Model II with 10 random variables to predict the energy consumption:

$$\text{Appliances, y} = \beta 0 + \beta_1 * \text{T1} + \beta_2 * \text{T2} + \beta_3 * \text{T3} + \beta_4 * \text{T4} + \beta_5 * \text{T5} + \beta_6 * \text{T7} + \beta_7 * \text{T8} + \beta_8 * \text{T\_out} + \beta_9 * \text{Press\_mm\_hg} + \beta_{10} * \text{Tdewpoint}$$

MODEL 2: Randomly variable selected model

Parameters	Original Model	Model with 10 random variables
cost	0.006631	0.006805
Iterations	1487	1510
MSE Train	172.45911	175.37497
MSE Test	78.52625	80.59183

Model II with 10 random features has greater value of MSE for both training and test dataset as compared to the Original Model I. There is 1.66% increase in MSE for training dataset and 2.56% increase for test dataset in randomly selected model compared to the original model. In addition to that, there is increase in cost function and no. of iterations as well.

Experimentation 4

Picking ten features that are best suited and retraining the models only on these ten features. Comparing train and test error results with the original and randomly selected model.

The Linear Regression Model III with 10 best suited variables to predict the energy consumption:

$$\text{Appliances, y} = \beta 0 + \beta_1 * \text{T1} + \beta_2 * \text{RH\_1} + \beta_3 * \text{T2} + \beta_4 * \text{RH\_2} + \beta_5 * \text{T3} + \beta_6 * \text{RH\_3} + \beta_7 * \text{T5} + \beta_8 * \text{RH\_5} + \beta_9 * \text{T\_out} + \beta_{10} * \text{RH\_out}$$

MODEL 3: Best suited variable selected model

Parameters	Original Model	Model with 10 random variables	Model with 10 best variables
cost	0.006631	0.006805	0.02688
Iterations	1487	1510	2000
MSE Train	172.45911	175.37497	774.73189
MSE Test	78.52625	80.59183	318.31814

Table 4.1

Converging threshold	Iterations	cost	MSE Train	MSE Test
0.0000001	2000	0.02688	774.7319	318.3181
0.000003	1561	0.02777	801.2273	328.8629
0.00001	1120	0.03026	873.3371	358.3648
0.0001	410	0.0576	1640.034	682.0929

Table 4.2

Siddhartha Shankar
https://github.com/sid0712
https://www.linkedin.com/in/sid-business/
siddhartha.shankar@icloud.com, 469-509-4807

Based on the tables above, cost function and MSE of model III for training and test datasets both are greater than the randomly selected model II and the original model I. Moreover, in table 4.1, no. of optimum iterations decreases while cost and MSE (train & test) both increases with increase in converging threshold.

As expected, using lesser number of features perform poorer as compared to using the more features. The reason for such an output is the lesser feature is not able to capture the variance in the dependent variable as good as the model with full features does.

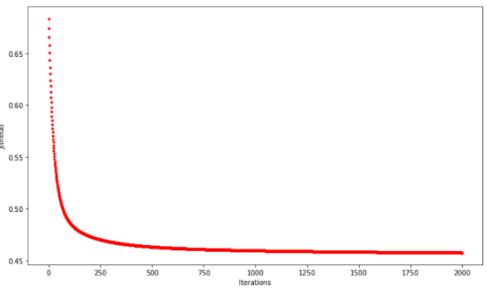
Logistic regression

Experimentation 1

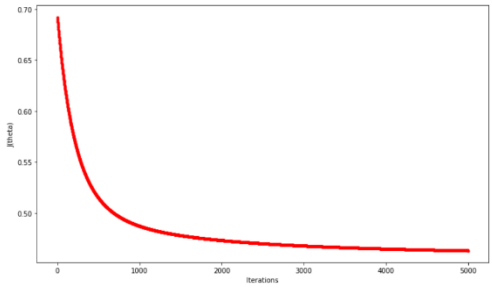
Experimenting with parameters for logistic regression (e.g. learning rate ∝, iterations, convergence threshold) and the error/accuracy for train and test sets is plotted. Optimal parameter values are reported.

Parameters input values:

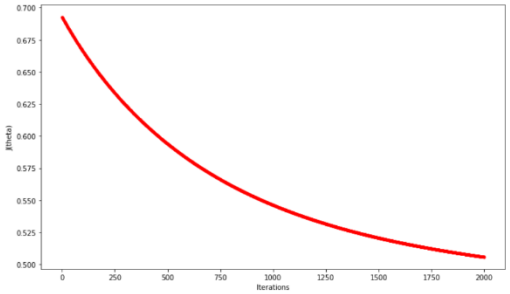
Theta = 0
Iterations = 2000
Convergence threshold = 0.000001
Learning rate, ∝ = 0.1, 0.01, 0.003



∝= 0.1



∝= 0.01



∝= 0.003

Learning rate, ∝	cost	Iterations	Baseline model Accuracy	Test data Accuracy
0.1	0.45062	2799	78.38%	80.96%
0.01	0.46977	5000	78.38%	79.97%
0.003	0.50342	5000	78.38%	78.82%

Observations based on the graph between J(theta) and Iterations and the table above:

Larger value of ∝ e.g. 0.1 results in the sudden drop in the curve which suggests that it might miss the optimal minima and jump up. However, no. of iterations and cost function is smaller larger for ∝ = 0.1.

As the value of ∝ decreases, the curve slowly moves towards zero rather than suddenly dropping as the cost function approaches its minimum value. At ∝ = 0.003, cost function is highest.

Learning rate, ∝= 0.01 is the optimal value provided fixed convergence threshold of 0. 000001 as it gives 80% prediction accuracy and relatively less cost than $\propto = 0.003$

Siddhartha Shankar
https://github.com/sid0712
https://www.linkedin.com/in/sid-business/
siddhartha.shankar@icloud.com, 469-509-4807

Hence, Optimal values of thetas for learning rate ($\propto$ = 0.01) are :

```
[[-1.25542833  0.0638731   0.38821111  0.1440785  -0.07696506  0.14769887
   0.10259758 -0.07817464 -0.00149167 -0.10655664  0.1059958  -0.09127324
  -0.35364316  0.04811859 -0.068329   -0.26946454  0.07910182 -0.10289467]]
```

Experimentation 3

Picking ten features randomly and retraining your models only on these ten features. Comparing train and test error results with the original model.

The Logistics Regression Model II with 10 random variables to predict the energy consumption:

$$\text{Appliances, y} = \beta 0 + \beta_1 * \text{T1} + \beta_2 * \text{T2} + \beta_3 * \text{T3} + \beta_4 * \text{T4} + \beta_5 * \text{T5} + \beta_6 * \text{T7} + \beta_7 * \text{T8} + \beta_8 * \text{T\_out} + \beta_9 * \text{Press\_mm\_hg} + \beta_{10} * \text{Tdewpoint}$$

MODEL 2: Randomly variable selected model

Logistic Model	cost	Iterations	Baseline model Accuracy	Test data Accuracy
Model I_Original	0.46977	5000	78.38%	81.00%
Model II_Random	0.75491	489	78.38%	78.00%

As expected, Model II with 10 random features has less accuracy as compared to the Original Model I.

Experimentation 4

Picking ten features that are best suited and retraining the models only on these ten features. Comparing train and test error results with the original and randomly selected model.

The Logistic Regression Model III with 10 best suited variables to predict the energy consumption:

$$\text{Appliances, y} = \beta 0 + \beta_1 * \text{T1} + \beta_2 * \text{RH\_1} + \beta_3 * \text{T2} + \beta_4 * \text{RH\_2} + \beta_5 * \text{T3} + \beta_6 * \text{RH\_3} + \beta_7 * \text{T5} + \beta_8 * \text{RH\_5} + \beta_9 * \text{T\_out} + \beta_{10} * \text{RH\_out}$$

MODEL 3: Best suited variable selected model

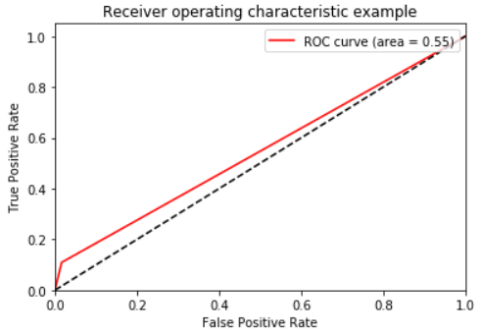
Logistic Model	cost	Iterations	Baseline model Accuracy	Test data Accuracy
Model I_Original	0.46977	5000	78.38%	81.00%
Model II_Random	0.76231	1953	78.38%	78.10%
Model III_best	0.49889	5000	78.38%	77.71%

Based on the table above, accuracy of model III is lower than the randomly selected model II and the original model I. However, cost is lower than the Model II.

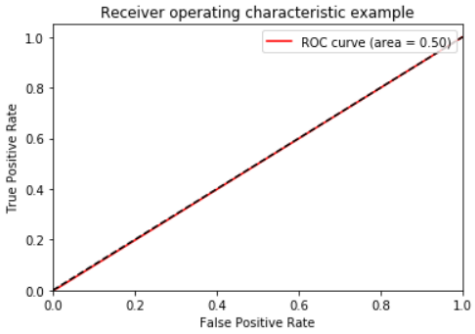
As expected, using lesser number of features perform poorer as compared to using the more features. The reason for such an output is the lesser feature is not able to capture the variance in the dependent variable as good as the model with full features does. Hence, model with a greater number of features performs best.

Siddhartha Shankar
https://github.com/sid0712
https://www.linkedin.com/in/sid-business/
siddhartha.shankar@icloud.com, 469-509-4807

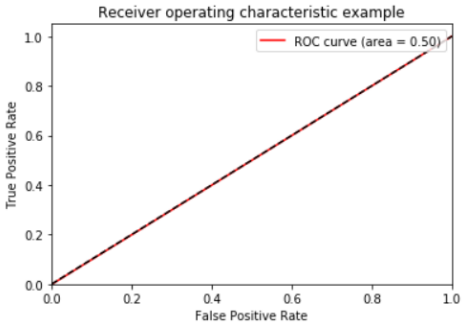
ROC Curve and AUC for Model I, II and III



Model I



Model II



Model III

Model Evaluation for various classifier models to come up with best classifier

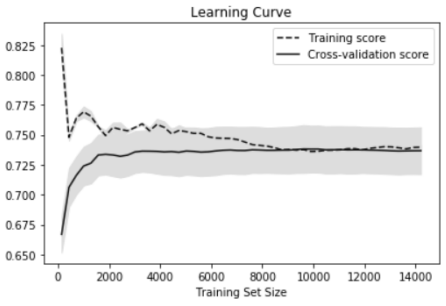
Model Statistics:

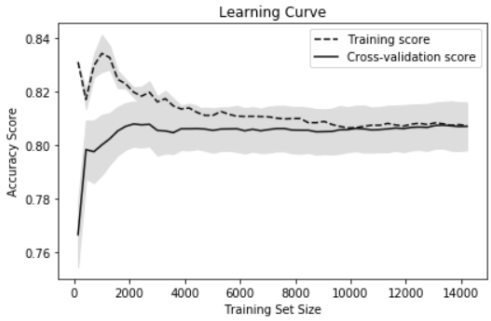
1. Logistic Regression

Classification Report & Confusion matrix

```
[[3020   92]
 [ 655  180]]
              precision    recall  f1-score   support

           0       0.82      0.97      0.89      3112
           1       0.66      0.22      0.33       835

    accuracy                           0.81      3947
   macro avg       0.74      0.59      0.61      3947
weighted avg       0.79      0.81      0.77      3947

Accuracy Score: 81.10000000000001
Cross-Validation Score: 80.0
```

Learning curves to evaluate model performance





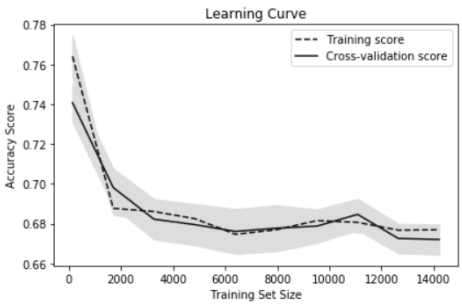
Model Accuracy on test dataset = 81.10%, Cross validation (CV) score = 80.0%, AUC = 0.74

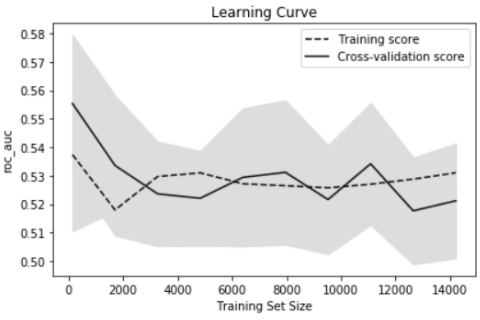
2. Support vector Machines: (kernel = sigmoid)

Classification Report & Confusion matrix

```
[[2489  623]
 [ 615  220]]
              precision    recall  f1-score   support

           0       0.80      0.80      0.80      3112
           1       0.26      0.26      0.26       835

    accuracy                           0.69      3947
   macro avg       0.53      0.53      0.53      3947
weighted avg       0.69      0.69      0.69      3947

Accuracy Score: 68.60000000000001
Cross-Validation Score: 78.60000000000001
```

Learning curves to evaluate model performance





Model Accuracy on test dataset = 68.60%, Cross validation (CV) score = 78.60%

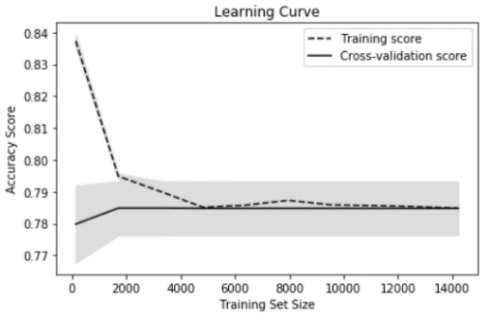
Siddhartha Shankar
https://github.com/sid0712
https://www.linkedin.com/in/sid-business/
siddhartha.shankar@icloud.com, 469-509-4807

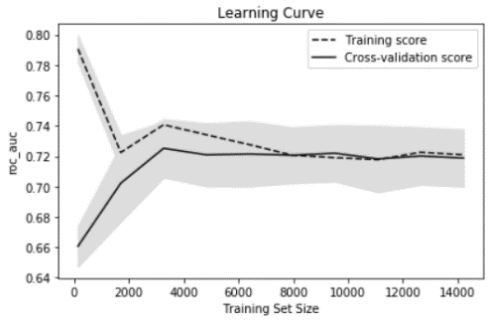
3. Support vector Machines: (kernel= linear)

Classification Report & Confusion matrix

```
[[3112    0]
 [ 835    0]]
              precision    recall  f1-score   support

           0       0.79      1.00      0.88      3112
           1       0.00      0.00      0.00       835

    accuracy                           0.79      3947
   macro avg       0.39      0.50      0.44      3947
weighted avg       0.62      0.79      0.70      3947

Accuracy Score: 78.8
Cross-Validation Score: 78.5
```

Learning curves to evaluate model performance





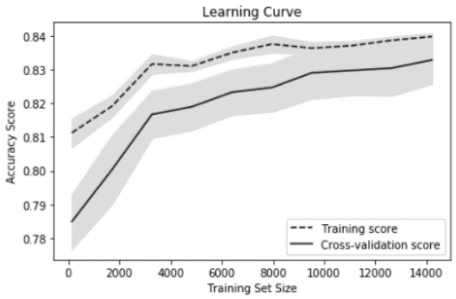
Model Accuracy on test dataset = 78.80%, Cross validation (CV) score = 78.50%

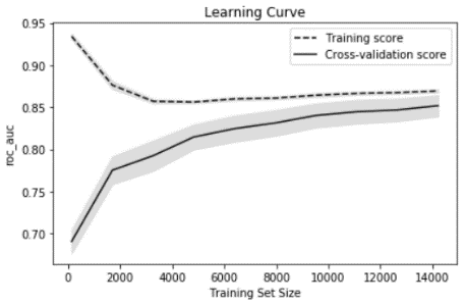
4. Support vector Machines: (kernel= Gaussian 'rbf')

Classification Report & Confusion matrix

```
[[3029   83]
 [ 549  286]]
              precision    recall  f1-score   support

           0       0.85      0.97      0.91      3112
           1       0.78      0.34      0.48       835

    accuracy                           0.84      3947
   macro avg       0.81      0.66      0.69      3947
weighted avg       0.83      0.84      0.81      3947

Accuracy Score: 84.0
Cross-Validation Score: 83.3
```

Learning curves to evaluate model performance





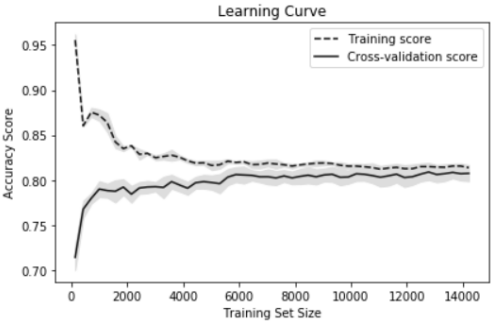
Model Accuracy on test dataset = 84.0%, Cross validation (CV) score = 83.30%

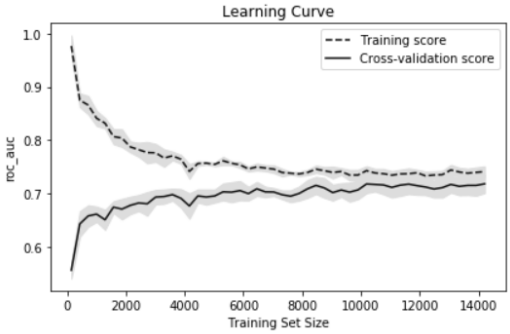
5. Decision Tree Classifier

Classification Report & Confusion matrix

```
[[3063   49]
 [ 676  159]]
              precision    recall  f1-score   support

           0       0.82      0.98      0.89      3112
           1       0.76      0.19      0.30       835

    accuracy                           0.82      3947
   macro avg       0.79      0.59      0.60      3947
weighted avg       0.81      0.82      0.77      3947

Accuracy Score: 81.6
Cross-Validation Score: 80.80000000000001
```

Learning curves to evaluate model performance





Model Accuracy on test dataset = 84.0%. Cross validation (CV) score = 83.30% and AUC = 0.73

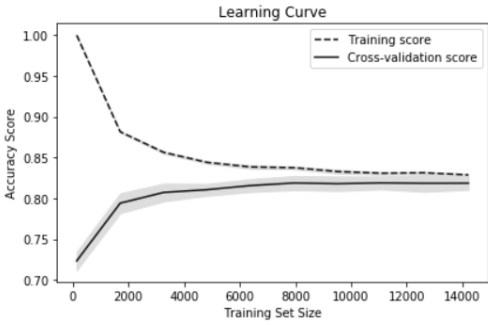
Siddhartha Shankar
https://github.com/sid0712
https://www.linkedin.com/in/sid-business/
siddhartha.shankar@icloud.com, 469-509-4807

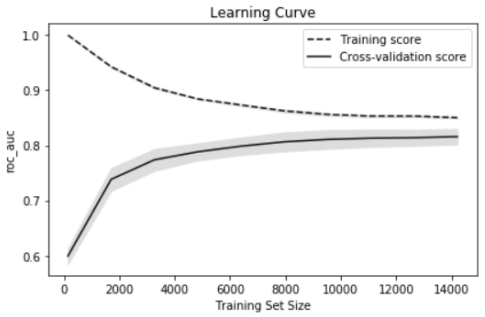
6. AdaBoost Classifier

Classification Report & Confusion matrix

```
[[2943  169]
 [ 396  439]]
              precision    recall  f1-score   support

           0       0.88      0.95      0.91      3112
           1       0.72      0.53      0.61       835

    accuracy                           0.86      3947
   macro avg       0.80      0.74      0.76      3947
weighted avg       0.85      0.86      0.85      3947

Accuracy Score: 85.7
Cross-Validation Score: 85.3
```

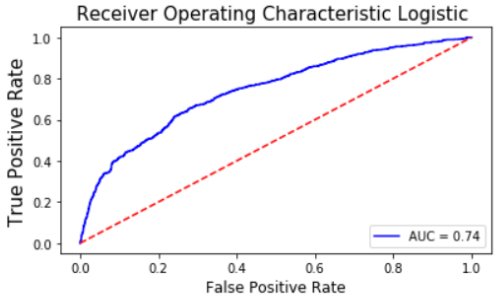
Learning curves to evaluate model performance

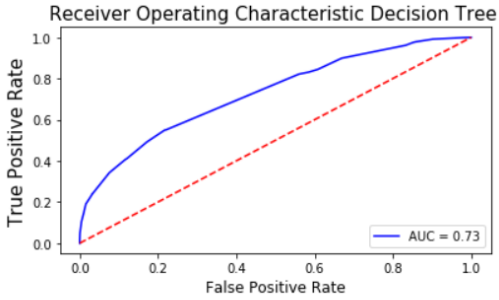


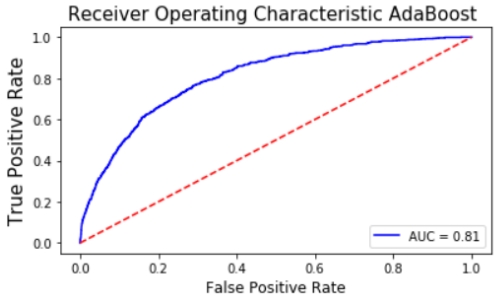


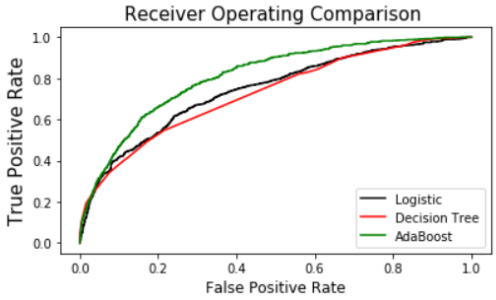
Model Accuracy on test dataset = 85.7%. Cross validation (CV) score = 85.30% and AUC = 0.81

ROC (Receiver Operating Characteristic) and AUC comparison between models









Models	Accuracy
AdaBoost Classifier	85.3
SVM: kernel=Gaussian	83.3
Decision Tree Classifier	80.8
Logistic Model	80.0
SVM: kernel=sigmoid	78.6
SVM: kernel=linear	78.5

Cross-Validation Accuracy based on Accuracy

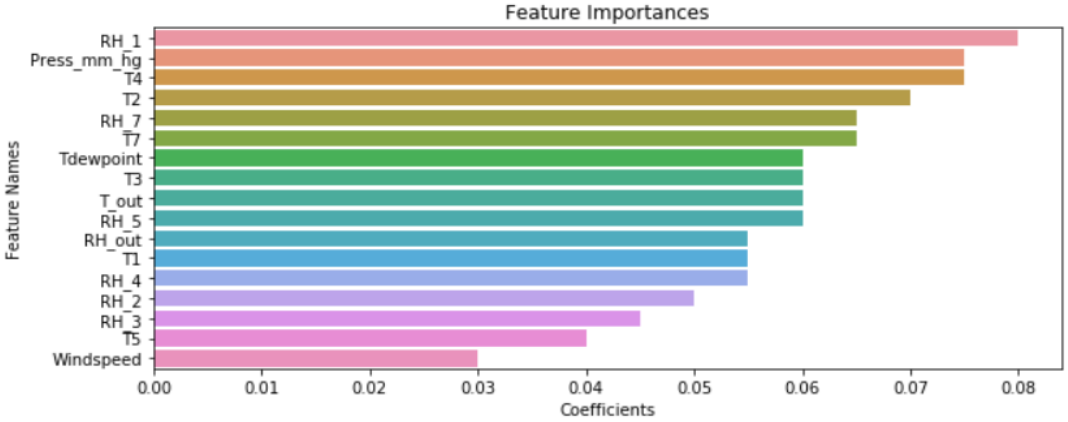
Cross- Validation score helps us to validate the model with cross-validation score. Higher the Cross-Validation score, better the model is in performing classification.

Based on ROC curve, AUC and Cross Validation (CV) score, AdaBoost Classifier performs best for the Energy Consumption dataset.

Siddhartha Shankar
https://github.com/sid0712
https://www.linkedin.com/in/sid-business/
siddhartha.shankar@icloud.com, 469-509-4807

Conclusion/Inference

As per AdaBoost classifier, important features to predict high and low energy consumption is as per the figure below. Humidity is the most significant factor.



- Based on the analysis, it can be concluded that model with higher number of features performs well comparison to model with fewer number of features. This is because higher number of variables explains much of the variation in target variable. However, prediction can be improved if we have access to demographic data such as family income, family size, geographical location etc.
- As there are no features having a linear relationship with the target variable. Linear Regression do not perform well with this data. Moreover, prediction can be improved with time series where we would have predicted future consumption based on past consumption.
- As per the correlation matrix, 'Humidity' attribute is positively correlated with target variable and have significantly higher values. Hence, 'Humidity' explains a lot of variation in energy consumption by Appliances.
- As per the Density plot and histogram, it is evident that target variable 'Appliances' is not normally distributed. Hence, using other functional form of target variable may predict better energy consumption by Appliances.
- The energy consumption is generally higher during weekends. So, creating binary variable such as weekdays and weekends will provide better insights and can have significant effect on prediction.