Summary:
Dataset taken from https://archive.ics.uci.edu/ml/datasets/Appliances+energy+prediction was tabulated by ZigBee wireless sensor network which recorded the readings for the following parameters for every ten minutes over the span of 4.5 months - temperate, pressure, humidity, wind speed, visibility, random variables and the amount of electricity used by household.

I have built a multi linear regression model using gradient descend and logistic regression model using SGD to predict the relationship between the various environmental factors and the amount of electricity used.

1. Data Description:

Parameter	Description
Number of rows x columns	19735 x 29
List of columns	Date, Dependent : Appliances, lights ; Room 1 : T1, RH_1 ; Room 2 : T2, RH_2 ; Room 3 : T3, RH_3 Room 4 : T4, RH_4 ; Room 5 : T5, RH_5 ; Room 6 : T6, RH_6 ; Room 7 : T7, RH_7 ; Room 8 : T8, RH_8 Room 9 : T9, RH_9 ; Weather station : T_out, Press_mm_hg ; RH_out, Windspeed, Visibility, Tdewpoint; Random : rv1, rv2

2. Missing values
It has been observed that there are no missing values in the dataset and so there is no need to perform any missing value imputation.

3. Data cleaning

- The date column provides no useful information for our regression and so it is removed from the dataframe.
- The following columns are removed from the dataframe : Date, Lights, Appliances, T6, RH_6, T7, RH_7,T8,RH_8, T9,RH_9, Lights.

4. Scaling values

The input factors such as temperature inside the specific room, humidity inside the specific room, pressure outside, humidity outside, wind speed, visibility and dew point are in different scales/standard units and bringing them to a common scale is required so that there is no domination of any feature on regression. The scaler function brings the entire data between 0 and 1.

5. Linear Regression:

(i) Variables

- **Input variables** (Independent) :
    - T1, RH_1, T2, RH_2, T3, RH_3, T4, RH_4, T5, RH_5,T_out, Press_mm_hg, RH_out, Windspeed, Visibility, Tdewpoint. The variables are represented by X1, X2…… X16 and their coefficients are written as β1,β2… β16. β0 is the intercept of the equation.
- **Output variables** (Dependent) : Appliances

(ii) Equation

The linear regression equation is written as,

Y = β0 + β1 * X1 + β2 * X2 + β3 * X3 + β4 * X4+β5 * X5+β6 * X6+β7 * X7+β8 * X8+β9 * X9+β10 * X10+ β11 * X11 +β12 * X12+β13 * X13+ β14 * X14 +β15 * X15 + β16 * X16

(iii) Cost function

Cost = $\frac{1* \textit{learning rate}}{2*\textit{Number of rows}} \sum_{i=1}^{\textit{number of rows}} \left(\widehat{y(\iota) - y}\,(\iota)\right)^2$

The beta coefficients which has the least cost function is the best linear regression line. To find the cost we need to differentiate with respect to each beta and subtract it from the beta values for multiple times with random restarts to get the best line.

(iv) Training and validation dataset

The entire dataset is split into training and validation dataset as 70% and 30%. After splitting we have the number of rows in training data set = 13814 and validation dataset = 5921.

(v) Computing cost function with experimentation and random restarts

Experiment 1 and Experiment 2 : For various threshold and various learning rates.

The learning rate and threshold are changed for various values and the results are computed below.

Setting β0, β1, β2…… β16 to be 0 as the initial values and we begin the iteration

Step 1 : Computing the errors for the learning rate 0.6,0.7,0.8 and 0.9 for threshold 5000 to 30,000 steps for 6 iterations with a difference of 5000.

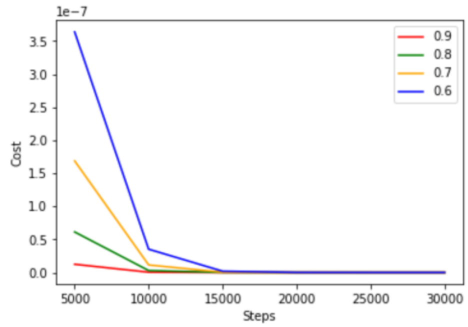
Learning rate	Threshold Steps					
	5000	10000	15000	20000	25000	30000
0.6	3.64E-07	3.52E-08	1.95E-09	6.62E-11	1.11E-12	8.42E-15
0.7	1.69E-07	1.14E-08	4.27E-10	8.41E-12	6.84E-14	2.15E-16
0.8	6.13E-08	2.93E-09	7.37E-11	8.18E-13	3.16E-15	4.03E-18
0.9	1.25E-08	4.25E-10	7.07E-12	4.33E-14	7.77E-17	3.94E-20

Step 2 : Choosing the learning rate and steps for the one which gives the minimum cost i.e., 0.9 and 30,000 steps. The learning rate of 0.9 and β0, β1, β2…… β16 = 0 is implemented for the validation dataset and the results are tabulated below.

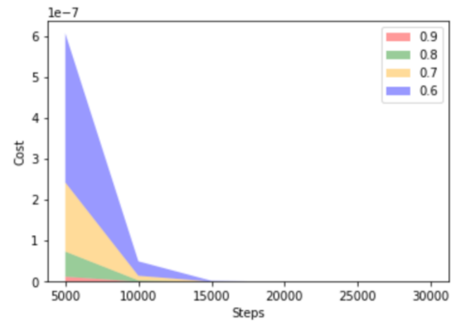
Data	Threshold Steps					
	5000	10000	15000	20000	25000	30000
Training	1.25E-08	4.25E-10	7.07E-12	4.33E-14	7.77E-17	3.94E-20
Validation	5.64E-09	1.66E-10	2.20E-12	9.81E-15	1.19E-17	3.74E-21

Step 3: Line chart and area chart for the training dataset with steps on the x-axis and cost on the y-axis is plotted for various learning rates – training dataset and validation dataset.

Training

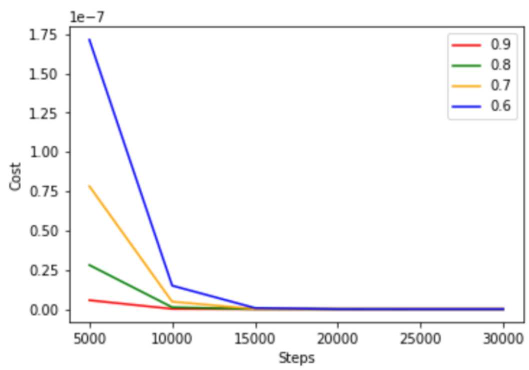


Linear Regression Gradient Descend for training data set

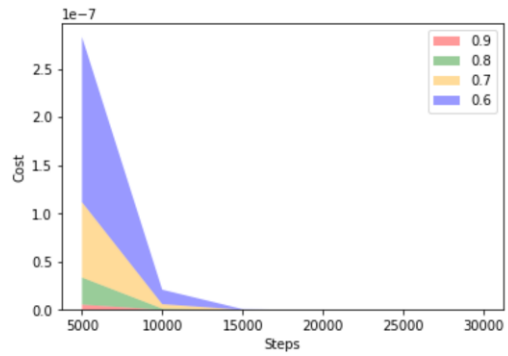


Linear Regression Gradient Descend for training data set

Validation



Linear Regression Gradient Descend for validation data set



Linear Regression Gradient Descend for validation data set

Step 4 : Beta values for the training data set when the learning rate is 0.9 computed.

The beta coefficients obtained from the training data set are now substitued for the validation dataset to compute the Mean square error value 16557.31436129327

Step 5: The values obtained from the python library and gradient descend are compared,

	Gradient descend	**Default**	**Difference**
beta 0	0.25349326	0.253493	1.7E-07
beta 1	9.26E-02	9.26E-02	3.1E-09
beta 2	6.46E-01	6.46E-01	1.6E-08
beta 3	-2.17E-01	-2.17E-01	9E-09
beta 4	-5.04E-01	-5.04E-01	3.3E-08
beta 5	2.10E-01	2.10E-01	3E-09
beta 6	-1.15E-02	-1.15E-02	1.75E-08
beta 7	-6.24E-02	-6.24E-02	3.3E-09
beta 8	-6.67E-02	-6.67E-02	4.7E-09
beta 9	-7.17E-02	-7.17E-02	9E-09
beta 10	2.05E-02	2.05E-02	2.8E-09

beta 11	-1.85E-01	-1.85E-01	3.6E-07
beta 12	1.44E-05	1.44E-05	3.06E-09
beta 13	-1.22E-01	-1.22E-01	1.86E-07
beta 14	2.06E-02	2.06E-02	7.8E-09
beta 15	1.05E-02	1.05E-02	1.8E-09
beta 16	1.07E-01	1.07E-01	2.73E-07

Setting β0, β1, β2…… β16 to be 0.1 as the initial values and we begin the iteration

Step 1 : Computing the errors for the learning rate 0.6,0.7,0.8 and 0.9 for 5000 to 30,000 steps for 6 iterations with a difference of 5000.

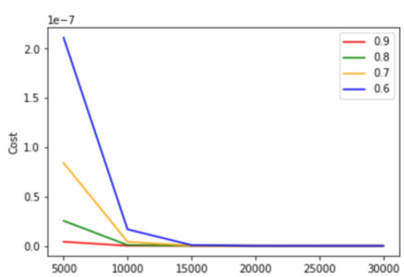
Learning rate	**Steps**					
	5000	**10000**	**15000**	**20000**	**25000**	**30000**
0.6	2.11E-07	1.67E-08	7.54E-10	2.38E-11	3.95E-13	2.99E-15
0.7	8.40E-08	4.09E-09	1.16E-10	2.13E-12	1.72E-14	5.40E-17
0.8	2.54E-08	7.35E-10	1.28E-11	1.33E-13	5.09E-16	6.50E-19
0.9	4.14E-09	6.64E-11	6.74E-13	3.81E-15	6.81E-18	3.46E-21

Step 2 : Choosing the learning rate and steps for the one which gives the minimum cost i.e., 0.9 and 30,000 steps. The learning rate of 0.9 and β0, β1, β2…… β16 = 0.1 is implemented for the validation dataset and the results are tabulated below.

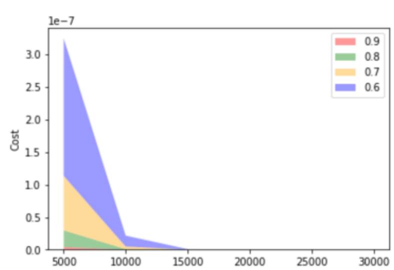
Data	**Steps**					
	5000	**10000**	**15000**	**20000**	**25000**	**30000**
Training	4.14E-09	6.64E-11	6.74E-13	3.81E-15	6.81E-18	3.46E-21
Validation	1.87E-09	2.58E-11	2.08E-13	8.56E-16	1.03E-18	3.25E-22

Step 3: Line chart and area chart for the training dataset with steps on the x-axis and cost on the y-axis is plotted for various learning rates – training dataset and validation dataset.

Training

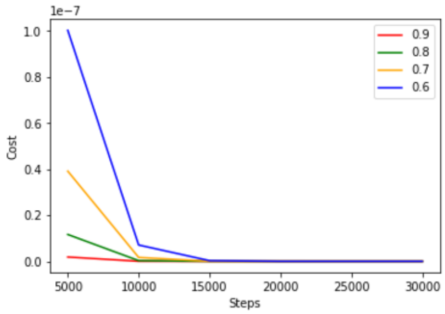


Linear Regression Gradient Descend for training data set

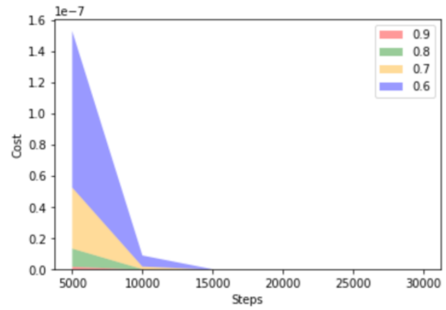


Linear Regression Gradient Descend for training data set

Validation



Linear Regression Gradient Descend for validation data set



Linear Regression Gradient Descend for validation data set

Step 4 : Beta values for the training data set when the learning rate is 0.9 computed.

The beta coefficients obtained from the training data set are now substitued for the validation dataset to computed Mean squared error : 16557.3377051091

Step 5: The values obtained from the python library and gradient descend are compared,

	Gradient descend	Default	Difference
beta 0	0.253493	0.253493	3.8E-07
beta 1	9.26E-02	9.26E-02	4.09E-05
beta 2	6.46E-01	6.46E-01	0.00047
beta 3	-2.17E-01	-2.17E-01	0.00024
beta 4	-5.04E-01	-5.04E-01	5.07E-05
beta 5	2.10E-01	2.10E-01	0.000413
beta 6	-1.15E-02	-1.15E-02	2.86E-05
beta 7	-6.24E-02	-6.24E-02	2.36E-05
beta 8	-6.67E-02	-6.67E-02	4.54E-05
beta 9	-7.17E-02	-7.17E-02	9.2E-06
beta 10	2.05E-02	2.05E-02	1.45E-05
beta 11	-1.85E-01	-1.85E-01	0.000377
beta 12	1.44E-05	1.44E-05	4.73E-08
beta 13	-1.22E-01	-1.22E-01	0.000494
beta 14	2.06E-02	2.06E-02	1.53E-05
beta 15	1.05E-02	1.05E-02	1.92E-05
beta 16	1.07E-01	1.07E-01	0.000238

(vi) Summary of experiments

Among the various scenarios done for the linear regression, the cost is less with beta set to 0.1 initially, learning rate 0.9 and threshold of 30,000 steps.

Y = 0.25349326 + 9.26E-02 * X1 + 6.46E-01 * X2 -2.17E-01 * X3 -5.04E-01 * X4 + 2.10E-01 * X5 -1.15E-02 * X6 -6.24E-02 * X7 -6.67E-02 * X8 -7.17E-02 * X9 + 2.05E-02 * X10 -1.85E-01 * X11 + 1.44E-05 * X12 -1.22E-01 * X13 + 2.06E-02 * X14 + 1.05E-02 * X15 + 1.07E-01 * X16

Logistic regression

Step 1: The values in input and output dataframe are scaled in between 0 and 1 using MinMaxScaler() function.

Step 2: The median of the output is 0.04 and so the values below 0.04 is set to be 0 and the values above 0.04 is set to be 1.

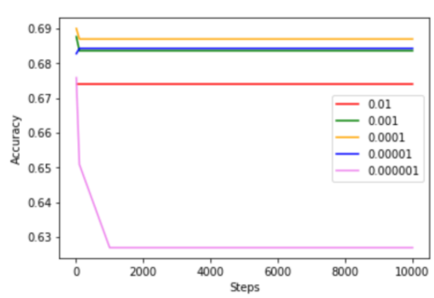
Step 3: 70% of the data is taken as training set and 30% in validation dataset thus having 13814 rows in training data set and 5921 rows in the validation dataset.

Step 4: SGDClassifier is used to perform logistic regression and the various experiments for alpha and iterations are shown below,

Experiment no	Alpha	Threshold (Steps)	Accuracy
1	0.01	10	0.6740415470359736
2	0.01	100	0.6740415470359736
3	0.01	1000	0.6740415470359736
4	0.01	10000	0.6740415470359736
5	0.001	10	0.6877216686370545
6	0.001	100	0.6836682992737714
7	0.001	1000	0.6836682992737714
8	0.001	10000	0.6836682992737714
9	0.0001	10	0.6900861340989698
10	0.0001	100	0.6870461070765074
11	0.0001	1000	0.6870461070765074
12	0.0001	10000	0.6870461070765074
13	0.00001	10	0.6828238473230873
14	0.00001	100	0.6843438608343185
15	0.00001	1000	0.6843438608343185
16	0.00001	10000	0.6843438608343185
17	0.000001	10	0.6758993413274784
18	0.000001	100	0.6509035635872319
19	0.000001	1000	0.6269211281878061
20	0.000001	10000	0.6269211281878061

Step 5: The best accuracy(69.00%) is obtained when the alpha is 0.0001 and 10 iterations.

Step 6: The line chart depicting, each learning rate with steps on x-axis and accuracy on y-axis is plotted below.



Logistic Regression Steps vs Accuracy

Step 7 : Confusion matrix

	Predicted	
Actual	812 (True positive)	1399 (False negative)
	436 (False positive)	3274 (True negative)

Step 8 : Confusion matrix terminologies

Parameters	Formula	Computation
Accuracy	(True positive + True negative) / (True positive + True negative + False Positive + False Negative)	(812 + 3274) / (812 + 3274 + 1399 + 436) = **0.690086**
Sensitivity / True positive rate	True positive/(True positive + False Negative)	812 / (812 + 1399) = **0.3673**
Specificity	True Negative/(True Negative + False Positive)	3274 / (3274 + 436) = **0.8824**
Precision	True Positive / (True positive + False Positive)	812 / (812 + 436) = **0.6506**
False positive rate	False positive / (False positive + True negative)	436 / (436 + 3274) = **0.1175**

Step 9 : AUC score

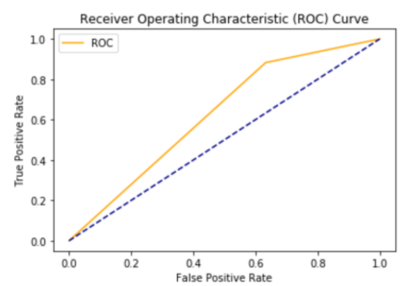
auc = roc_auc_score(Y_Test_logistic,validation_data) = 0.6248672101389645

Step 10: Classification report and ROC curve

```
              precision    recall  f1-score   support

           0       0.65      0.37      0.47      2211
           1       0.70      0.88      0.78      3710

    accuracy                           0.69      5921
   macro avg       0.68      0.62      0.63      5921
weighted avg       0.68      0.69      0.66      5921
```



The beta values from beta0 to beta 16 : 1.74381151, 2.52754395, 8.34048411, 1.04910764, -2.70634032 ,0.25027259,-4.45253276,-0.30169553, -0.54235959 ,-0.62798051 , 1.54883637, -0.62292934, -1.04076414, -1.48840341, 0.52930166, 0.02767399, -0.23102268

Experiment 3 - Random 10 features ::

These are the input variables which are considered to build out linear and logistic regression model, 'T1', 'RH_1', 'T2', 'RH_2', 'T3', 'RH_3', 'T_out', 'Press_mm_hg', 'Windspeed', 'Visibility' which are denoted as X1,X2…..X10

Linear regression

The linear equation line is, Y = β0 + β1 * X1 + β2 * X2 + β3 * X3 + β4 * X4+β5 * X5+β6 * X6+β7 * X7+β8 * X8+β9 * X9+β10 * X10

Step 1: Using 16 variables we found out that when alpha = 0.9 and steps as 30,000 we had the minimum cost. Using the same to build the linear regression function,

Step 2: After running the model the beta coefficients computed are tabulated below,

<u>Training set :</u>

Cost	**Using 16 features** (Previous experiments)	**Using 10 features random**
Training	5.501E-21	3.1269641753475197e-16
Validation	1.94E-13	1.1350815347796348e-10

Step 3: The beta values are as follows,

	Using LOGSITIC function	**Using SGD**	**Difference**
Beta 0	0.197	0.197319	0.000319
Beta 1	0.065	0.064772	0.000228
Beta 2	0.704	0.704338	0.000338
Beta 3	-0.315	-0.31515	0.00015
Beta 4	-0.673	-0.67343	0.00043
Beta 5	0.178	0.178331	0.000331
Beta 6	-0.027	-0.0267	0.000296
Beta 7	-0.021	-0.02116	0.000163
Beta 8	0.001	0.00138	0.00038
Beta 9	0.035	0.034694	0.000306
Beta 10	0.008	0.008282	0.000282

Step 4: Substituting the beta coefficients in validation dataset, the Mean squared error is 3087.6658567015143 which is lesser than what is obtained from 16 variables 16557.3377051091.

Logistic regression

The logistic model is built for alpha 0.001 and threshold of 10,The parameters are computed below,

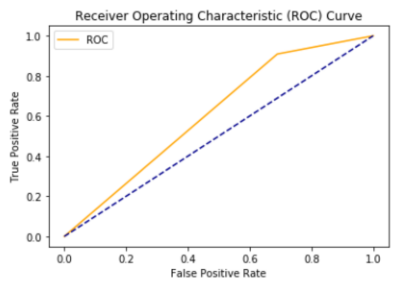
Confusion matrix	**Predicted**	
Actual	686 (True positive)	1525 (False negative)
	337 (False positive)	3373 (True negative)

Parameters	**Formula**	**Computation**
Accuracy	(True positive + True negative) / (True positive + True negative + False Positive + False Negative)	0.6097156340327279

Sensitivity / True positive rate	True positive/(True positive + False Negative)	(686) / (686 + 1525) = 0.31026684758
Specificity	True Negative/(True Negative + False Positive)	3373 / (3373 + 337) = 0.90916442048
Precision	True Positive / (True positive + False Positive)	686 / (686 + 337) = 0.67057673509
False positive rate	False positive / (False positive + True negative)	337 / (337 + 3373) = 0.09083557951

AUC Score : 0.6097156340327279

ROC curve and Classification report




```
              precision    recall  f1-score   support

           0       0.67      0.31      0.42      2211
           1       0.69      0.91      0.78      3710

    accuracy                           0.69      5921
   macro avg       0.68      0.61      0.60      5921
weighted avg       0.68      0.69      0.65      5921
```

Experiment 4 : Best 10 features

Variables considered

The 10 features are chosen only those pertaining to the room – Kitchen, laundry, office room, Teenager room, parent room 1,3,4,8 and 9 since electricity is based primarily on the factors within the room - T1, RH_1, T3, RH_3, T4, RH_4, T8, RH_8 and T9, RH_9. (X1,X2….X10)

Linear regression $Y = \beta_0 + \beta_1 * X_1 + \beta_2 * X_2 + \beta_3 * X_3 + \beta_4 * X_4 + \beta_5 * X_5 + \beta_6 * X_6 + \beta_7 * X_7 + \beta_8 * X_8 + \beta_9 * X_9 + \beta_{10} * X_{10}$

Step 1: Using 16 variables we found out that when alpha = 0.9 and steps as 30,000 we had the minimum cost. Using the same to build the linear regression function,

Step 2: After running the model the beta coefficients computed are tabulated below,
Training set :

Cost	Using 16 features	Using 10 features best
Training	5.501E-21	4.0152586070908055e-32
Validation	1.94E-13	1.096475981992043e-20

Step 3: The beta values are as follows,

	Using gradient	Default package	Difference
Beta 0	0.050489	0.059	0.008511
Beta 1	-0.08933	-0.055	0.034329
Beta 2	0.228513	0.241	0.012487
Beta 3	0.265008	0.286	0.020992
Beta 4	0.081864	0.068	0.013864

Beta 5	0.042509	0.033	0.009509
Beta 6	0.042718	0.011	0.031718
Beta 7	0.098382	0.051	0.047382
Beta 8	-0.23556	-0.206	0.029563
Beta 9	-0.2533	-0.251	0.002303
Beta 10	-0.04184	-0.05	0.00816

Step 4: Substituting the beta coefficients in validation dataset, the Mean squared error is 118.34527426668004 which is lesser than what is obtained from 16 variables 16557.3377051091.

Logistic regression

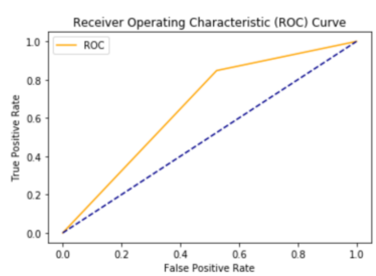
The logistic model is built for alpha 0.001 and maximum iterations are 10. The parameters are tabulated,

	Predicted	
Actual	1053 (True positive)	1158 (False negative)
	565 (False positive)	3145 (True negative)

Parameters	Formula	Computation
Accuracy	(True positive + True negative) / (True positive + True negative + False Positive + False Negative)	0.6619819915370464
Sensitivity / True positive rate	True positive/(True positive + False Negative)	1053 / (1053 + 1158) = 0.47625508819
Specificity	True Negative/(True Negative + False Positive)	3145 / (3145 + 565) = 0.84770889487
Precision	True Positive / (True positive + False Positive)	1053 / (1053 + 565) = 0.65080346106
False positive rate	False positive / (False positive + True negative)	565 / (565 +3145) = 0.15229110512

AUC Score : 0.6619819915370464

ROC curve and Classification report




```
              precision    recall  f1-score   support

           0       0.65      0.48      0.55      2211
           1       0.73      0.85      0.78      3710

    accuracy                           0.71      5921
   macro avg       0.69      0.66      0.67      5921
weighted avg       0.70      0.71      0.70      5921
```