# Assignment 1: Project Report - Simple Linear Regression

**Submitted By:** **Shubham Murari**
**Net ID:** **SXM170056**
**Course:** **BUAN 6341.501 – Applied Machine Learning**

**Problem Summary:**

We have been provided with a regression problem for **'Appliances Energy Prediction'.** There are 19735 instances (records) with no missing values. The energy data was logged on a time basis where the Temperature and Humidity Conditions of a house were monitored with the help of sensors. Additionally, the data is merged with the nearest airport's weather station's data. Also, the actual dataset shows that this is multi variate time series problem but for our analysis and specially for this project we are not considering the date time variable.

**Goal:**

Our goal is to implement the Gradient Descent algorithm and calculate the Cost Function value for our model, where we are predicting the Appliances' energy uses which is our target variable. We need to predict it on the basis of different values of Temperature, Humidity, Visibility etc. which is given in our dataset.

**Exploratory Analysis and Feature Engineering:**

Before directly jump into any conclusion or analysis, there are certain steps that I kept in my mind before starting this interesting exercise.

1. The various Temperature and Humidity values that were given to me, are they from same building/area?
2. If yes, then, how these values are related to each other?
3. How significant each value is when it comes to the relation between my target variable and independent variables?
4. Is there any *correlation*?
5. How is the distribution of my data? Is it normal or skewed etc.?

Once I made myself clear with all these details I started with my analysis. But again, let me walk through the way in which I answered all these questions and fix these issues.

**Step 1:**

When I closely observed the summary statistics for various **Temperature and Humidity** values which were given in the data set, I found the following things-

| | Appliances | lights | T1 | RH_1 | T2 | RH_2 | T3 | RH_3 | T4 | RH_4 |
|---|---|---|---|---|---|---|---|---|---|---|
| count | 19735.000000 | 19735.000000 | 19735.000000 | 19735.000000 | 19735.000000 | 19735.000000 | 19735.000000 | 19735.000000 | 19735.000000 | 19735.000000 |
| mean | 97.694958 | 3.801875 | 21.686571 | 40.259739 | 20.341219 | 40.420420 | 22.267611 | 39.242500 | 20.855335 | 39.026904 |
| std | 102.524891 | 7.935988 | 1.606066 | 3.979299 | 2.192974 | 4.069813 | 2.006111 | 3.254576 | 2.042884 | 4.341321 |
| min | 10.000000 | 0.000000 | 16.790000 | 27.023333 | 16.100000 | 20.463333 | 17.200000 | 28.766667 | 15.100000 | 27.660000 |
| 25% | 50.000000 | 0.000000 | 20.760000 | 37.333333 | 18.790000 | 37.900000 | 20.790000 | 36.900000 | 19.530000 | 35.530000 |
| 50% | 60.000000 | 0.000000 | 21.600000 | 39.656667 | 20.000000 | 40.500000 | 22.100000 | 38.530000 | 20.666667 | 38.400000 |
| 75% | 100.000000 | 0.000000 | 22.600000 | 43.066667 | 21.500000 | 43.260000 | 23.290000 | 41.760000 | 22.100000 | 42.156667 |
| max | 1080.000000 | 70.000000 | 26.260000 | 63.360000 | 29.856667 | 56.026667 | 29.236000 | 50.163333 | 26.200000 | 51.090000 |

- **Temperature Values:** Temperature values (T1, T2, T3 etc. other than T6 which is the outside temperature of the building) have almost **Mean Values in the range of 19-21**
- **Humidity Values:** Most of the Humidity values (RH_1, RH_2, RH_3 etc.) have the **Mean Values in the range of 38-40**

**So this paerticular observation gave me an idea that there MIGHT be some correaltion between the temperature and humidity values of different rooms inside the building. To get more clarity on this I opted for the Correlation (Heat Map) Plot of my data set.**

**Step 2:**

As I discussed before, this small observation of summary stats gave me an inkling of correlation between different variables, to get more clarity on that I tried to plot the correlation plot for my data set.

This plot uses a warm-to-cool color spectrum to show how different variables are correlated to each other, but some times to observe these many variables can be very taxing. So for this purpose I implemented the method to fetch correlation matrix to further select the features for my analysis.

I used the following function – ***set the correlation threshold as 0.80***

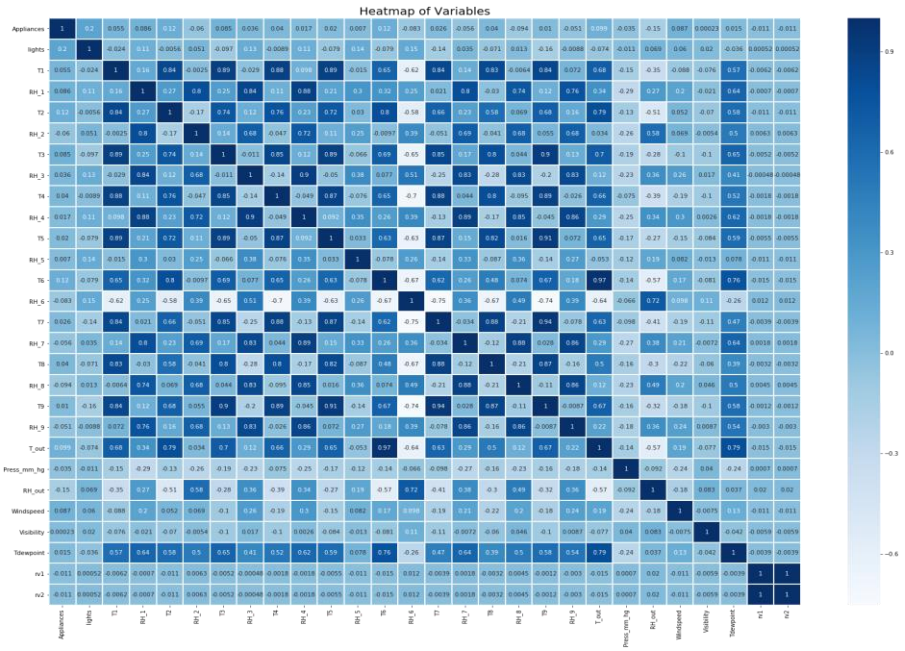
**Further drill down the Correlation Matrix:**

```
corr_matrix = df_Energy.corr().abs()

#the matrix is symmetric so we need to extract upper triangle matrix without diagonal (k = 1)

sol = (corr_matrix.where(np.triu(np.ones(corr_matrix.shape), k=1).astype(np.bool))
                  .stack()
                  .sort_values(ascending=False))
sol = sol.to_frame()
sol.columns=['corr']
sol[sol['corr'] >  0.8]
```

**Heat Map:**



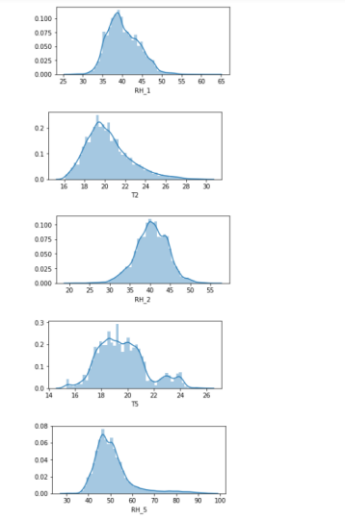
**Correlation Matrix:** (it is just for the reference, whole matrix is available on notebook)

| | | corr |
|---|---|---|
| **rv1** | **rv2** | 1.000000 |
| **T6** | **T_out** | 0.974787 |
| **T7** | **T9** | 0.944776 |
| **T5** | **T9** | 0.911055 |
| **T3** | **T9** | 0.901324 |
| **RH_3** | **RH_4** | 0.898978 |
| **RH_4** | **RH_7** | 0.894301 |
| **T1** | **T3** | 0.892402 |

**Step 3:**

Distribution of the variables – this is also an important aspect, ideally there should be a normal distribution for the regression. I quickly checked it using the *'dist. plot'*

As you can see below most of my values were **normally distributed** for this data-



However, we later normalized the values also, so that we can fir it into our model, normalizing the variables always help to achieve good prediction results or we can say ***optimized cost function value***.

**Feature Selection:**

Once the data is prepared and exploratory analysis is done, I selected 16 Features from my dataset (which were basically filtered on the basis of correlation and distribution)

**Regression Equation:**

**Appliances = 96.934 + 15.622*Lights + 61.415*RH_1 – 13.875*T2 – 39.913*RH_2 – 4.507*T5 +**

**0.6641*RH_5 – 0.999*RH_6 + 11.864*T8 – 26.322*RH_8 – 3.299*T_out –**

**0.386*Press_mm_hg – 2.4457*RH_out + 5.7501*Windspeed + 1.5803*Visibility +**

**4.242*Tdewpoint – 1.142*rv1**

## Experiment 1: (Linear Regression & Logistic Regression)

**Objective-**

Experiment with various Parameters for Linear Regression (e.g. learning rate ∝) and plot the curves for both Train and Test Cost.

**Solution:**

Following features were selected for this experiment including the Target Variable:

```
(['Appliances', 'lights', 'RH_1', 'T2', 'RH_2', 'T5', 'RH_5', 'RH_6',
  'T8', 'RH_8', 'T_out', 'Press_mm_hg', 'RH_out', 'Windspeed',
  'Visibility', 'Tdewpoint', 'rv1'],
```

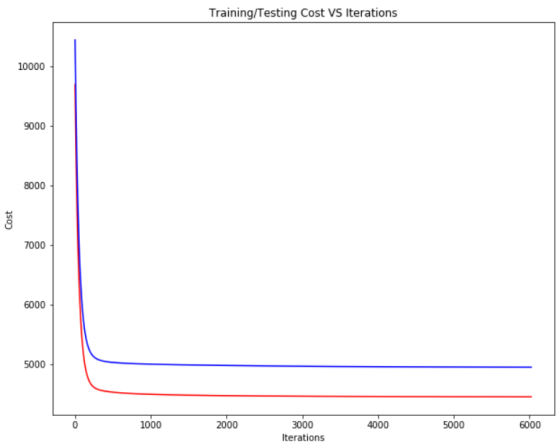
**Step 1:** Data was splitted to Train and Test using the 70/30 split rule.
**Step 2:** Further we performed the feature vectorization and normalization.
**Step 3:** Initilize the value for *beta* as 1.
**Step 4:** Implemented the following functions for ***Cost Function, Gradient Descent and Learning Curves***
**Step 5:** We tried different values of alpha and calculated the Train Cost and Test Cost for this model.

| Learning Rate (∝) | Iterations | Threshold | Train Cost | Test Cost |
|---|---|---|---|---|
| 0.01 | 6022 | 0.001 | 4453.244 | 4949.398 |
| 0.1 | 989 | 0.001 | 4451.761 | 4944.767 |
| 0.001 | 24213 | 0.001 | 4467.126 | 4972.513 |
| 0.5 | 3 | 0.001 | 4882.811 | 5357.399 |

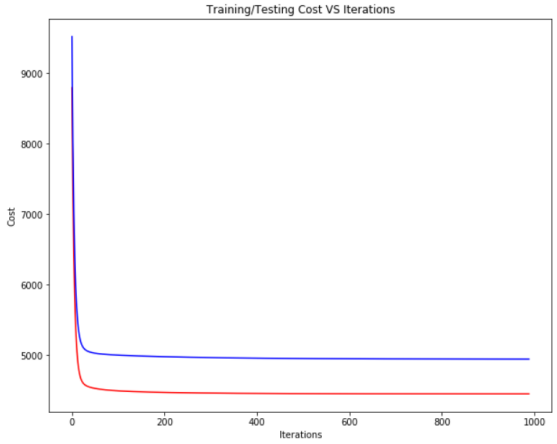
So, for this experiment the maximum number of iterations were kept fixed at 100000 and learning rate were experimented at **0.01, 0.1, 0.001, 0.5**

**Above is the table which shows how Train and Test Cost is getting varied with the Learning Rate (∝).**

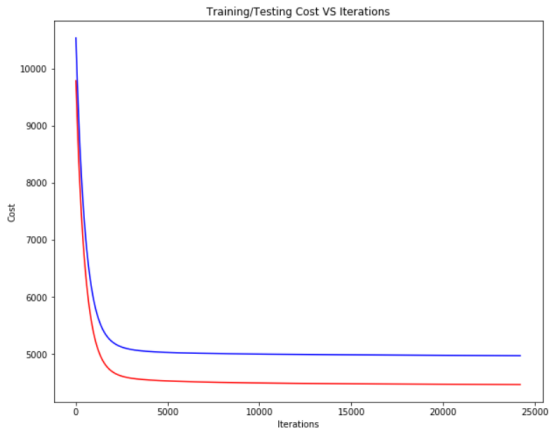
**Below is the plot for Training/Test Cost vs Iterations for different values of (∝) for Linear:**

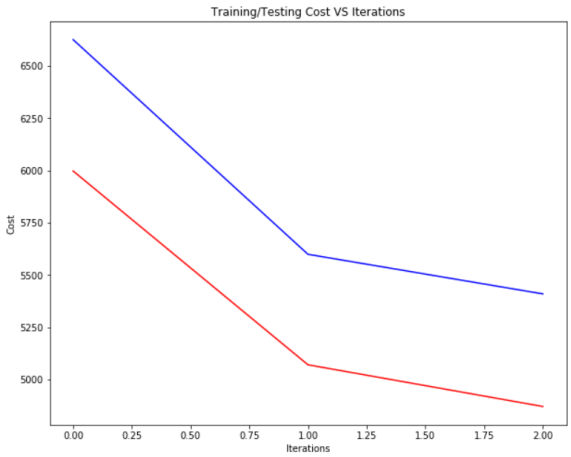



∝ = 0.01




∝ = 0.1




$\propto = 0.001$ $\propto = 0.5$

**Explaination:**

- We can see from the train and test plots and the number of iterations taken by the gradient descent algorithm taking to converge we can say that –
- For smaller learning rate, the number of iterations required is more and in case of learning rate 0.001 the algorithm fails to converge within 20000 iterations.
- For this particular experiment, the best learning rate is 0.1 as the train and the test error are minimum and iterations are also less.

## Part 2: Logistic Regression

We will convert this problem to the Logistic Regression problem, for this purpose I calculated the **Median of Appliances** which came out to be **60.0**.

I used the following logic to convert the problem into the **Binary Classification** problem:

```
In [14]: df['Appliances_Energy'] = np.where(df['Appliances']>= 60, 1, 0)
         df.drop(columns=['Appliances'],axis=1,inplace=True)
```

| | |
|---|---|
| **1:** | **Signifies the Higher Energy Uses** |
| **0:** | **Signifies the Lower Energy Uses** |

Additional function that we implemented was the **Sigmoid Function:**

**Implementing Sigmoid function**

```
def sigmoid(z):
    return 1/ (1 + np.exp(-z))
```

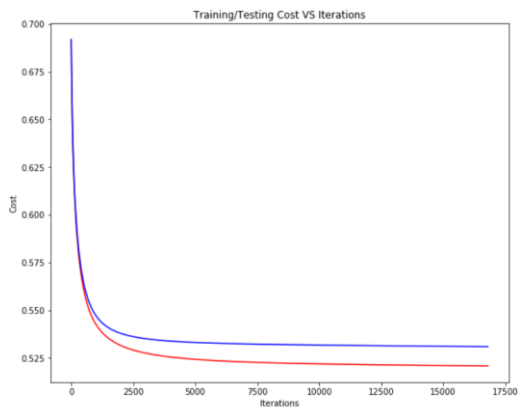
We observed the following values for Learning Rate (alpha) and Cost Functions:

| Learning Rate (∝) | Iterations | Train Cost | Test Cost | Test Accuracy |
|---|---|---|---|---|
| 0.01 | 16803 | 0.52082 | 0.530929 | 73.09576 |
| 0.1 | 4262 | 0.51983 | 0.52995 | 73.1295 |
| 0.5 | 1230 | 0.51974 | 0.52985 | 73.0282 |
| 1.0 | 697 | 0.51973 | 0.52984 | 73.0957 |

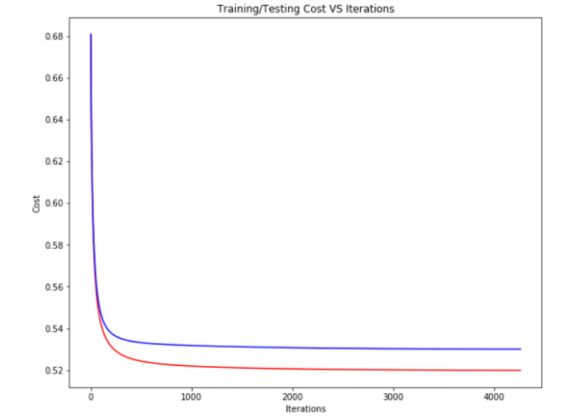
**Explanation:**

- For Experiment 1, the best results are obtained for a value of alpha =1. The train cost and test cost are as low as 0.51973 and 0.52984 respectively.
- Also, it is observed that for a fixed threshold of 0.0000001, the costs decrease as we increase the value of alpha.Accuracy is around **73.1 %** for all values of alpha.

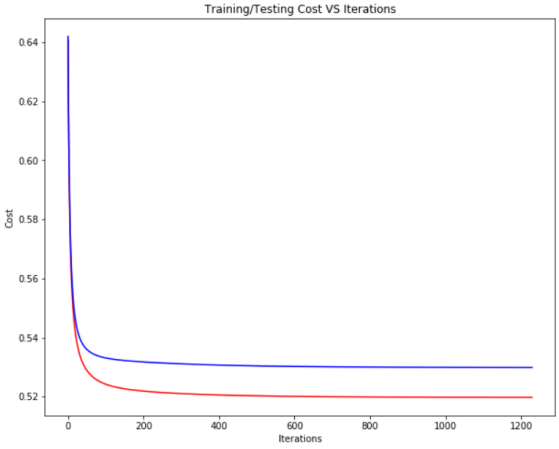
**Below is the plot for Training/Test Cost vs Iterations for different values of (∝) for Logistic:**



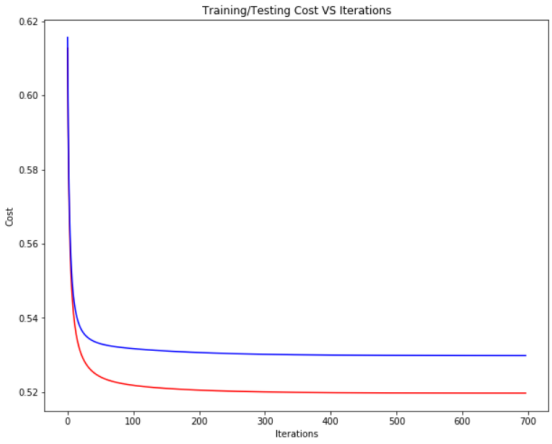

∝ = 0.01




∝ = 0.1




∝ = 0.5




∝ = 1.0

## Experiment 2: (Linear Regression – Change Convergence Threshold)

**Objective:**

Experiment with various thresholds for convergence for linear regression. Plot error results for train and test sets as a function of threshold and describe how varying the threshold affects .

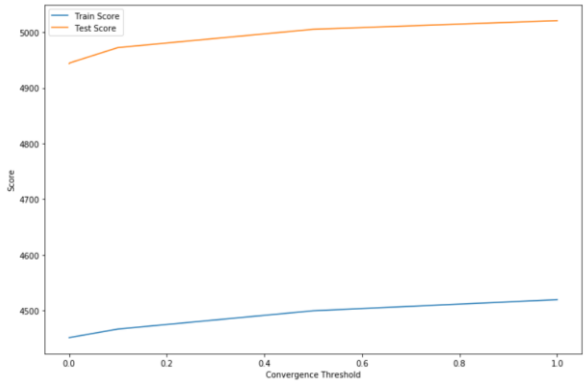
For the same set of features that we considered in Experiment 1 and following the same steps, lets fix the value of alpha and vary the thresholds-

We will get the following observations:

| Learning Rate (∝) | Iterations | (Convergence_Criteria) | Train Cost | Test Cost |
|---|---|---|---|---|
| 0.1 | 1758 | 0.0001 | 4451.533 | 4943.488 |
| 0.1 | 989 | 0.001 | 4451.761 | 4944.767 |
| 0.1 | 602 | 0.01 | 4453.228 | 4949.360 |
| 0.1 | 242 | 0.1 | 4466.999 | 4972.359 |
| 0.1 | 87 | 0.5 | 4499.824 | 5005.192 |
| 0.1 | 58 | 1.0 | 4519.764 | 5020.695 |

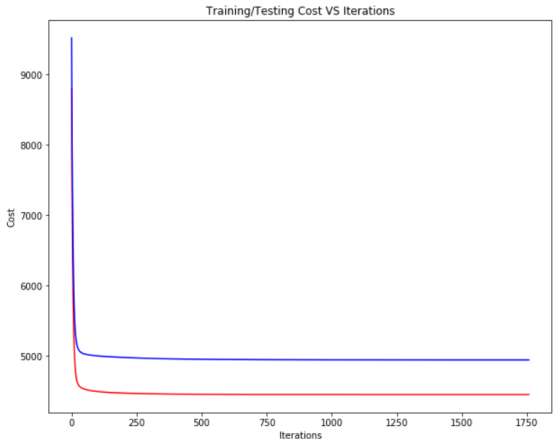
The threshold values chosen for this experimentation are 0.0001, 0.001, 0.01, 0.1, 0.5, 1.0 at the fixed learning rate of alpha 0.1.

**The plot of Train/Test Score vs Convergence Threshold is shown below:**




So this plot clearly shows that for the **lower** values of Convergence Threshold i.e. **0.0001** both the Train Cost and Test Cost are having the minimum values as compare to the other thresholds.

To get the more clarity on this we can also plot the **Train/Test Cost vs Iteration** for best combination i.e **Convergence Criteria and Learning Rate is 0.0001 and 0.1** respectively-




**So the best value of threshold seems to be 0.0001 here.**

## Experiment 3: (Pick 10 Random Features)

Using **np. random. Choice (),** I just picked 10 random variables and follow the same steps as above. Below is the best value for Train Cost and Test Cost in case of Random variable selection.

*10 Randomly Picked variables are:*

```
df_exp3_cols = np.random.choice(df_Energy.columns,10,replace=False)
```

```
df_exp3_cols
```

```
array(['Press_mm_hg', 'RH_out', 'T9', 'lights', 'RH_5', 'T8', 'RH_7',
       'T2', 'RH_9', 'T7'], dtype=object)
```

Again at alpha = 0.1 we are getting the best value for Randomly Picked variables but again these values are greater as compared to the model which uses 16 features.

**Reason:** This is as expected because the features that we picked to predict the Appliances Energy, were picked without any statistical significance, it just random selection. Its highly unlikely that we get the optimized values for cost function in this particular case. Moreover, odds of randomly picking 10 features performing better than the 16 feature dataset is quiet low.

**For Linear Regression**

| Model | Learning Rate (∝) | Iterations | Train Cost | Test Cost |
|---|---|---|---|---|
| Model with 16 Features | 0.1 | 989 | 4451.761 | 4944.767 |
| Model with 10 Random Features | 0.1 | 989 | 4716.215 | 5179.216 |

**For Logistic Regression**

| Model | Learning Rate (∝) | Iterations | Train Cost | Test Cost | Test Accuracy |
|---|---|---|---|---|---|
| Model with 16 Features | 1.0 | 697 | 0.51973 | 0.52984 | 73.0957 |
| Model with 10 Random Features | 1.0 | 527 | 0.58095 | 0.58585 | 69.9881 |

## Experiment 4: (Pick 10 Best Features)

The 10 best features for this particular problem that I selected are:

**T1, T2, T6, RH_1, RH_2, RH_6, T_out, lights, Windspeed, RH_out**

**My Approach behind the selection of 10 best features:**

- Initial approach behind selecting these features was correlation between them. However, there are some features which I believe from the energy consumption's point of view. e.g. **lights** feature might have many 0 values but again its kind of *important* feature when it comes to predicting the value of actual energy consumption.
- Similarly, there are different rooms in the building e.g. I considered mainly the temperature and humidity of the **Kitchen Area, Living Room** and Northside of the Building (outside). Its highly likely that the consumption of electricity in the Living Room and Kitchen will be more due to frequent use of different appliances e.g. *Toaster, Refrigerator, Blender, TV, AC, Heater, Chimneys, Microwaves, Burners, Lamps etc.*
- And **Outside Temperature/Humidity** also plays a critical role in the consumption of devices which comes under HVAC division like (AC and Heater Systems). Based upon the outside humidity and temperature only we use these appliances which again consume high energy as compare to other small appliances.

**For Linear Regression**

| Model | Learning Rate (∝) | Iterations | Train Cost | Test Cost | Performance |
|---|---|---|---|---|---|
| **Model with 16 Features** | 0.1 | 989 | 4451.761 | 4944.767 | Best |
| **Model with 10 Random Features** | 0.1 | 235 | 4716.215 | 5179.216 | Bad |
| **Model with 10 Best Features** | 0.1 | 1935 | 4501.347 | 4945.306 | Good |

**For Logistic Regression**

| Model | Learning Rate ($\propto$) | Iterations | Train Cost | Test Cost | Test Accuracy | Performance |
|---|---|---|---|---|---|---|
| **Model with 16 Features** | 1.0 | 697 | 0.51973 | 0.52984 | 73.0957 | Best |
| **Model with 10 Random Features** | 1.0 | 527 | 0.58095 | 0.58585 | 69.9881 | Bad |
| **Model with 10 Best Features** | 1.0 | 1323 | 0.55235 | 0.55763 | 69.785509 | Good |

**Explanation:**

- This model Performs better than the model where we Randomly picked the 10 features because there was no statistical significance was taken in consideration in case of random selection, the **Training Cost and Test Cost of 10 best Feature Model** is slightly better than the model that had random features.
- But when it comes to the comparison of this model which had **10 best features** with the model that had **16 features**, there is a slight difference in the Training and Testing Cost (as shown in the table above). **Costs are slightly high for model with 10 best Features**. **Because –** it might be the case that smaller feature might be missing some important features due to size constraint or it might not be able to capture the variance in the dependent variable. On the contrary for the model that has 16 features it might be working fine.

**Discussions: Challenges and what more can be done-**

1. As we know that Linear and Logistic Regression are the baseline algorithms, there are advanced ML algorithms like **Random Forest, Decision Trees, KNN** etc. which are expected to give the much more accurate results.
2. We could have implemented **Ridge and Lasso Regression** for better feature selections and more accurate results.
3. Since there was a **constraint in number of features** for the experiment 3 and 4, it was very challenging to pick the best features to train model. This made the model more restricted and the features that we selected might have not able to explain the variance of dependent variable.
4. There were some features like Temperature and Humidity which were not varying much within the building, some of them were highly correlated which might have result in the higher cost values for Random Selection experiment. However, for the remaining experiments we take the correlation factor into consideration.

**********