Machine Learning

Dr. Shahid Mahmood Awan
Assistant Professor
School of Systems and Technology, University of Management and Technology
shahid.awan@umt.edu.pk

Umer Saeed **(MS Data Science, BSc Telecommunication Engineering)**
Sr. RF Optimization & Planning Engineer
f2017313017@umt.edu.pk

Machine Learning

Linear Regression

UNIVERSITY OF MANAGEMENT AND TECHNOLOGY

UMT

Machine Learning

Data Source

Energy Efficiency Data Set

- Energy Efficiency Data Set selected for Task-1 from the following URL; https://archive.ics.uci.edu/ml/datasets/Energy+efficiency

- **Source:**
- The dataset was created by Angeliki Xifara and was processed by Athanasios Tsanas.

Energy Efficiency Data Set

- **Data Set Information:**

- We perform energy analysis using 12 different building shapes simulated in Ecotect.

- The buildings differ with respect to the glazing area, the glazing area distribution, and the orientation, amongst other parameters.
-
- We simulate various settings as functions of the afore-mentioned characteristics to obtain 768 building shapes.

- The dataset comprises 768 samples and 8 features, aiming to predict two real valued responses. It can also be used as a multi-class classification problem if the response is rounded to the nearest integer.

Energy Efficiency Data Set

- **Attribute Information:**

- The dataset contains eight attributes (or features, denoted by X1...X8) and two responses (or outcomes, denoted by Y1 and Y2). The aim is to use the eight features to predict each of the two responses.

Attributes/Features/Variables
X1 Relative Compactness
X2 Surface Area
X3 Wall Area
X4 Roof Area
X5 Overall Height
X6 Orientation
X7 Glazing Area
X8 Glazing Area Distribution

Outcome/Responses
Y1 Heating Load
Y2 Cooling Load

Energy Efficiency Data Set

Item	Variable	Discrete Values	Range
1	X1 Relative Compactness	12	[0.62 , 0.98]
2	X2 Surface Area	12	[514.5 , 808.5]
3	X3 Wall Area	7	[245 , 416.5]
4	X4 Roof Area	4	[110.25 , 220.5]
5	X5 Overall Height	2	[3.5 , 7]
6	X6 Orientation	4	[2 ,5]
7	X7 Glazing Area	4	[0 ,0.4]
8	X8 Glazing Area Distribution	6	[0 , 5]

- **Table 1: Input Variables (Attributs)**

Item	Outcome	Mean	Range
1	Y1 Heating Load	22.30720052	[6.01 , 43.1]
2	Y2 Cooling Load	24.58776042	[10.9, 48.03]

- **Table 2: Output Variables (Responses)**

Machine Learning

Linear Regression with one variable

Linear Regression with One Variable

- The first step is to import the required Libraries

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import scipy.stats as stats
```

- The second step is to load the dataset (**Energy Efficiency Data Set)**

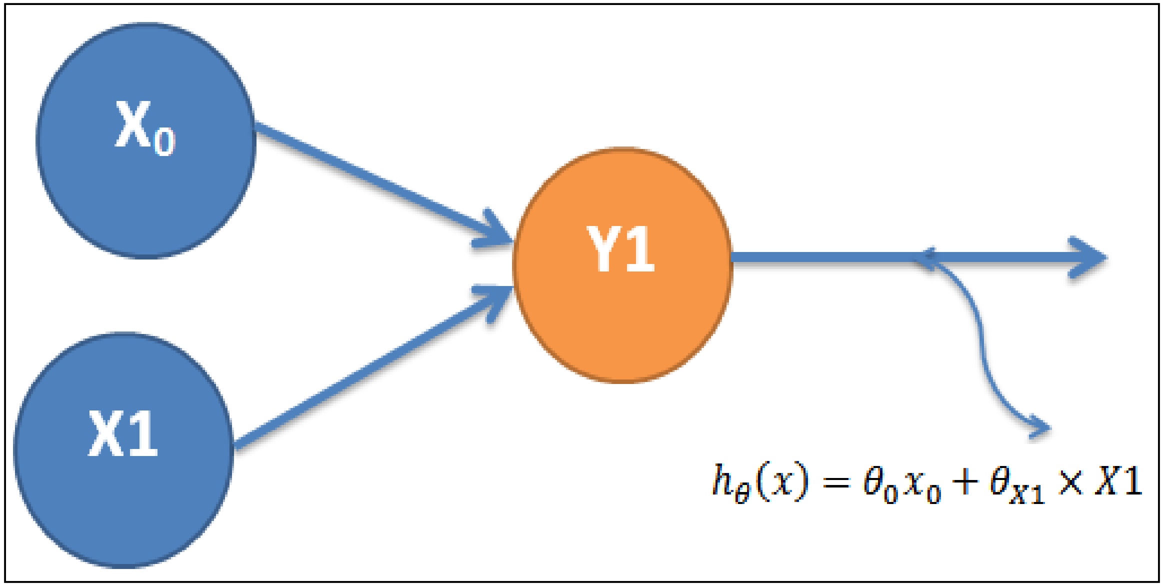
```
enb=pd.read_csv("ENB2012_data.csv")
```

- The data will be loaded using Python Pandas, a data analysis module. It will be loaded into a structure known as a Panda Data Frame, which allows for each manipulation of the rows and columns.

Machine Learning

Linear Regression with One Variable(Outcome Heating Loading)

Linear Regression with One Variable



- Create two arrays: x (X1 Relative Compactness) and y (Y1 Heating Load). Intuitively we'd expect to find some correlation between the two variables.

Linear Regression with One Variable

- In next step, Splitting the dataset into the features set and Outcome set

```
x = enb.iloc[:, 0].values

x=x.reshape(len(x),1)

y = enb.iloc[:, -2].values

y=y.reshape(len(y),1)
```

- The data will be split into a training and test set. Once we have the test data, we can find a best fit line and make predictions.

```
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.1, random_state= 0)
```

Linear Regression with One Variable

- Fitting Simple Linear Regression to the Training set

```
from sklearn.linear_model import LinearRegression

regressor = LinearRegression()

regressor.fit(x_train, y_train)
```

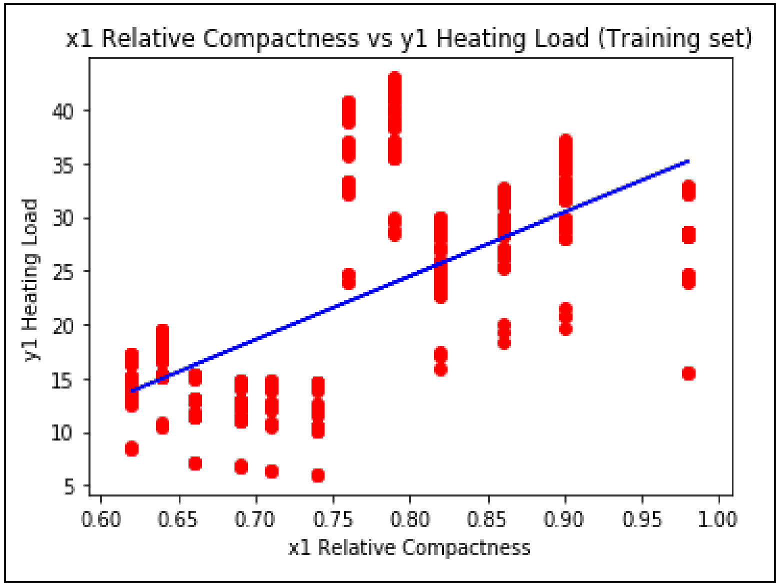
- Predicting the Test set results

```
y_pred = regressor.predict(x_test)
```

Linear Regression with One Variable

- Visualising the Training set results

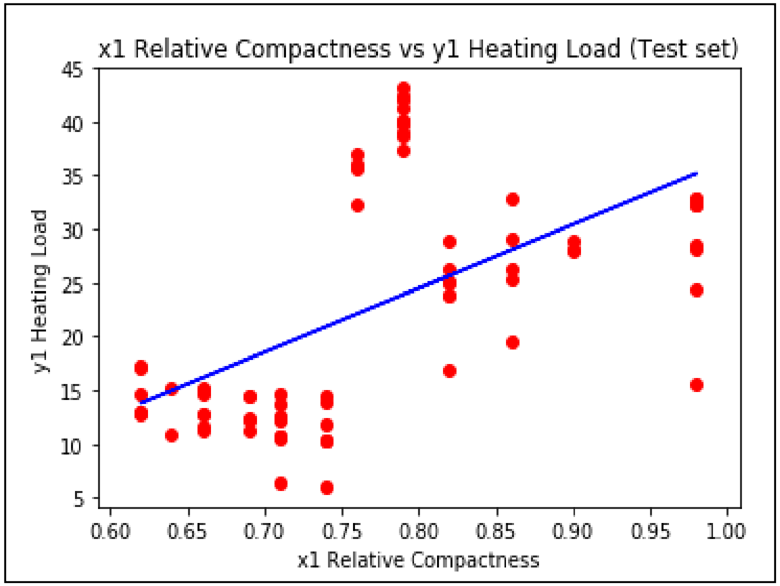
```
plt.scatter(x_train, y_train, color = 'red')
plt.plot(x_train, regressor.predict(x_train), color = 'blue')
plt.title('x1 Relative Compactness vs y1 Heating Load (Training set)')
plt.xlabel('x1 Relative Compactness')
plt.ylabel('y1 Heating Load')
plt.show()
```



Linear Regression with One Variable

- Visualising the Test set results

```
plt.scatter(x_test, y_test, color = 'red')
plt.plot(x_train, regressor.predict(x_train), color = 'blue')
plt.title('x1 Relative Compactness vs y1 Heating Load (Test set)')
plt.xlabel('x1 Relative Compactness')
plt.ylabel('y1 Heating Load')
plt.show()
```



Linear Regression with One Variable

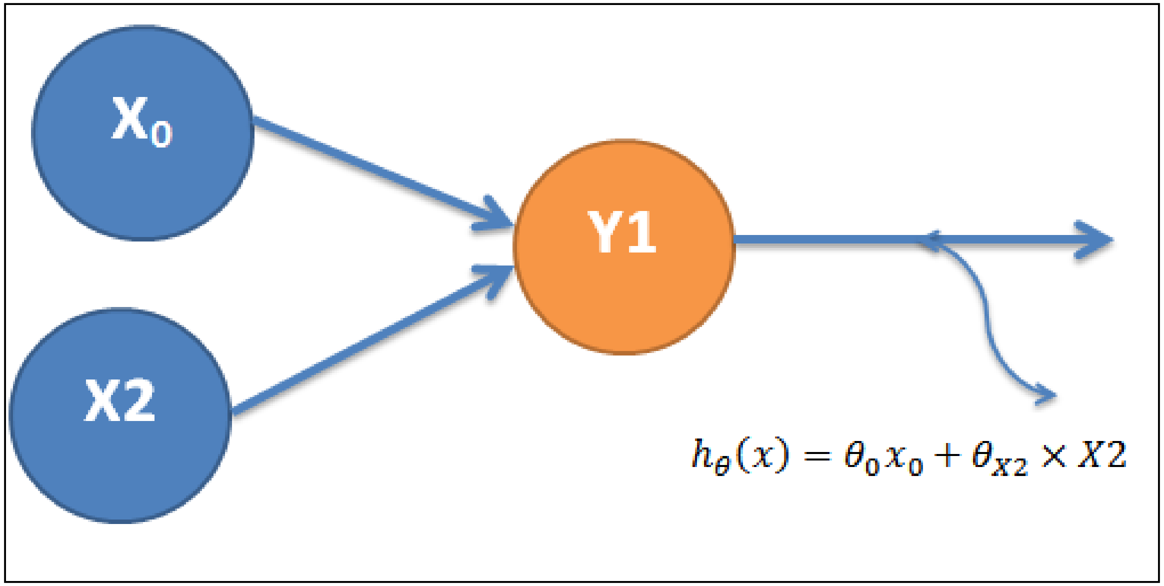
- Building the optimal model using Backward Elimination

```
import statsmodels.formula.api as sm

x= np.append(arr=np.ones((768,1)).astype(int),values=x,axis=1)

x_opt = x[:,[0,1]]

regressor_OSL = sm.OLS(endog=y, exog=x_opt).fit()

regressor_OSL.summary()
```

Linear Regression with One Variable

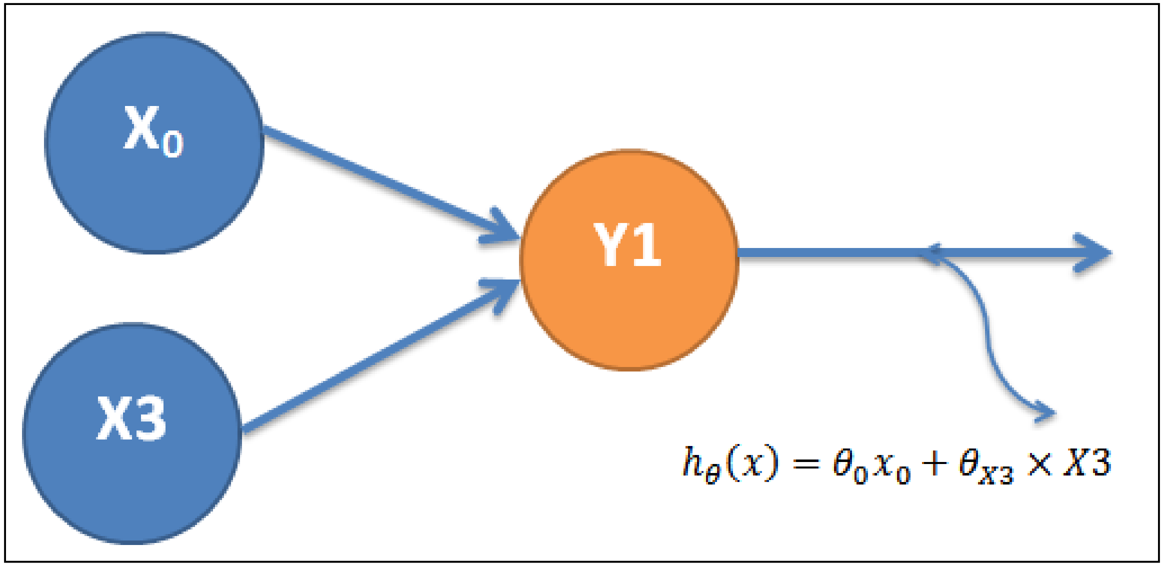
```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.387
Model:                            OLS   Adj. R-squared:                  0.386
Method:                 Least Squares   F-statistic:                     484.0
Date:                Sun, 22 Apr 2018   Prob (F-statistic):           1.59e-83
Time:                        12:35:23   Log-Likelihood:                -2676.5
No. Observations:                 768   AIC:                             5357.
Df Residuals:                     766   BIC:                             5366.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        -23.0530      2.081    -11.076      0.000     -27.139     -18.967
x1            59.3590      2.698     22.001      0.000      54.063      64.655
==============================================================================
Omnibus:                       53.989   Durbin-Watson:                   0.305
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               64.327
Skew:                           0.708   Prob(JB):                     1.08e-14
Kurtosis:                       3.059   Cond. No.                         15.0
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

Linear Regression with One Variable



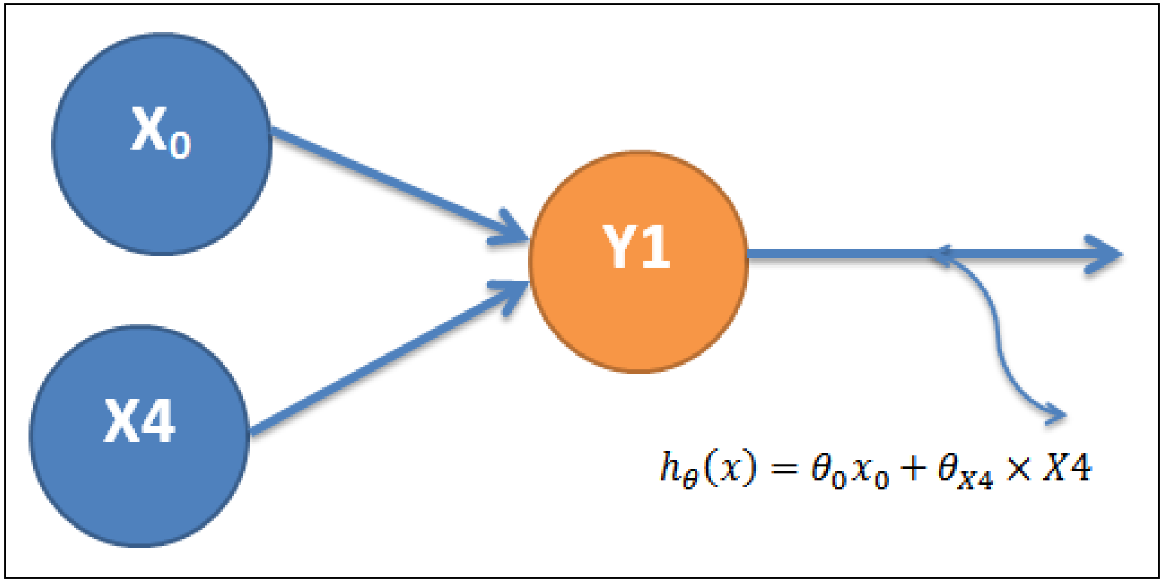
- Create two arrays: x (X2 Surface Area) and y (Y1 Heating Load). Intuitively we'd expect to find some correlation between the two variables.

Linear Regression with One Variable



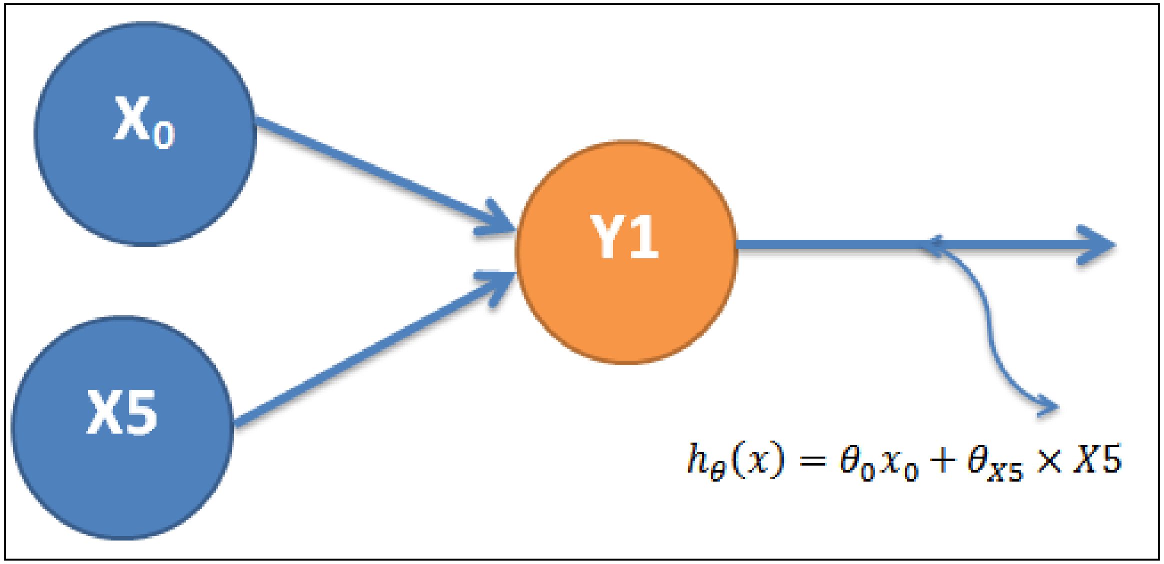
- Create two arrays: x (X3 Wall Area) and y (Y1 Heating Load). Intuitively we'd expect to find some correlation between the two variables.

Linear Regression with One Variable



- Create two arrays: x (X4 Roof Area) and y (Y1 Heating Load). Intuitively we'd expect to find some correlation between the two variables.

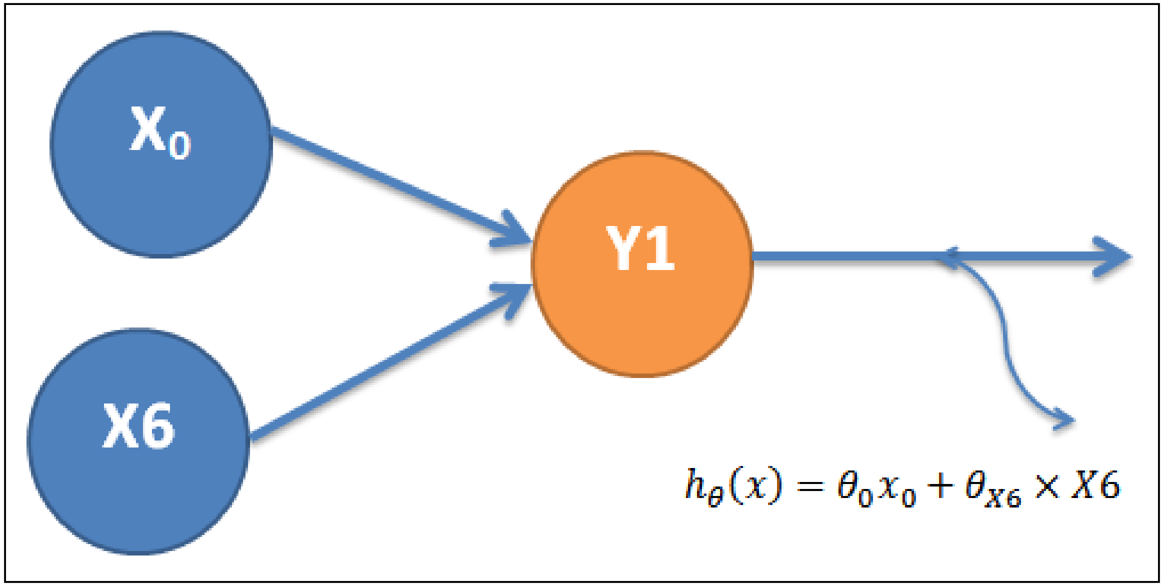
Linear Regression with One Variable



$h_\theta(x) = \theta_0 x_0 + \theta_{X5} \times X5$

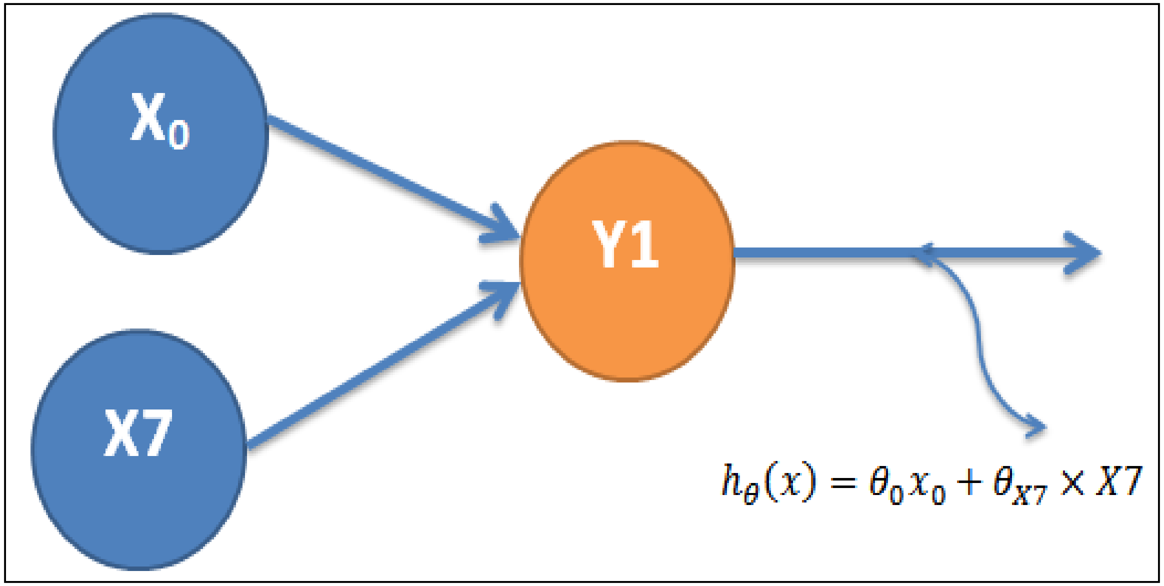
- Create two arrays: x (X5 Overall Height) and y (Y1 Heating Load). Intuitively we'd expect to find some correlation between the two variables.

Linear Regression with One Variable



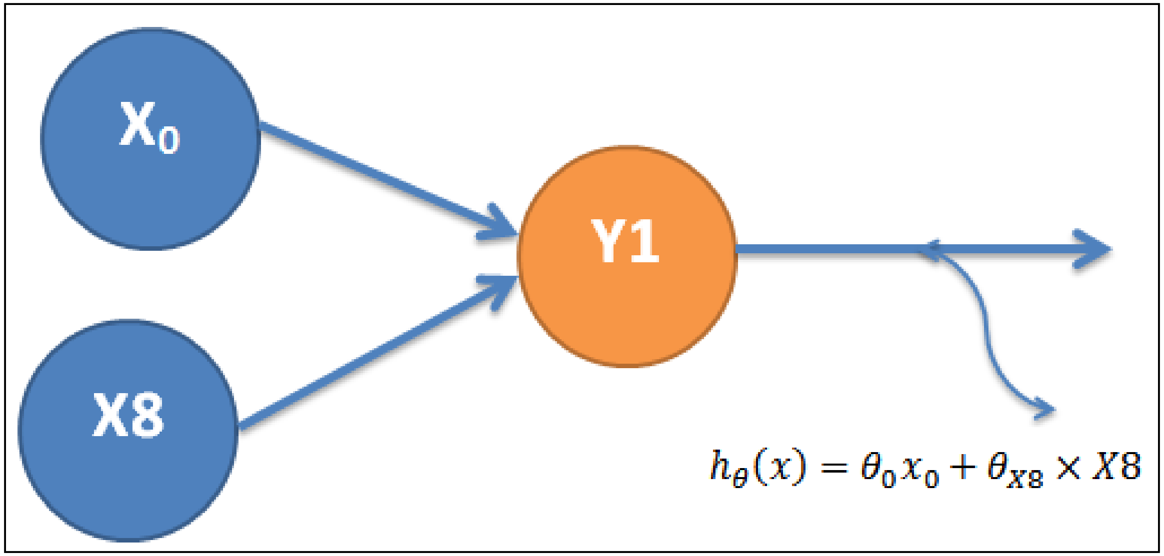
- Create two arrays: x (X6 Orientation) and y (Y1 Heating Load). Intuitively we’d expect to find some correlation between the two variables.

Linear Regression



- Create two arrays: x (X7 Glazing Area) and y (Y1 Heating Load). Intuitively we'd expect to find some correlation between the two variables.

Linear Regression

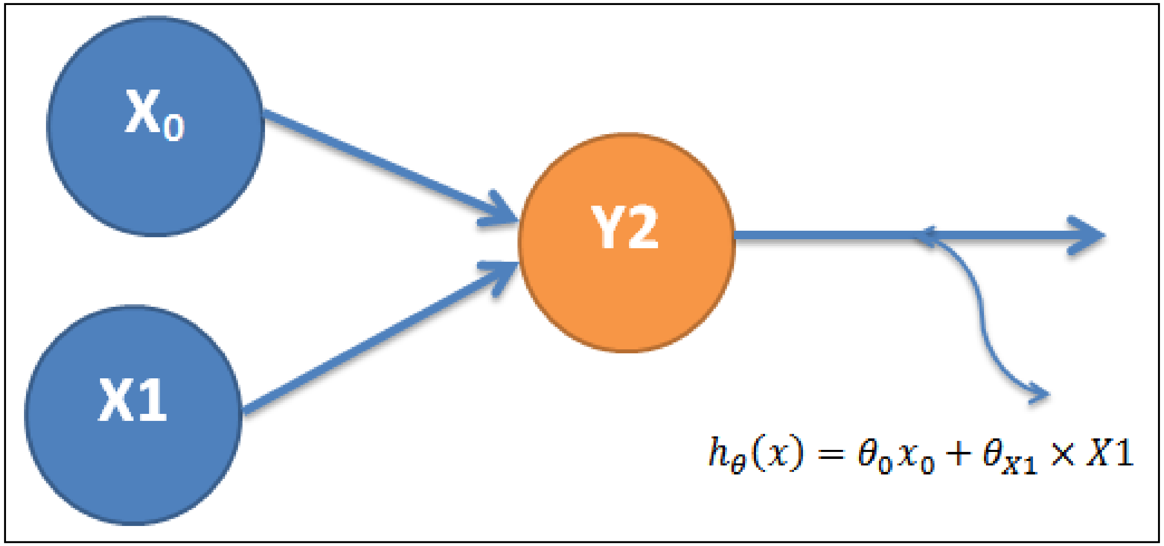


- Create two arrays: x (X8 Glazing Area Distribution) and y (Y1 Heating Load). Intuitively we'd expect to find some correlation between the two variables.

Machine Learning

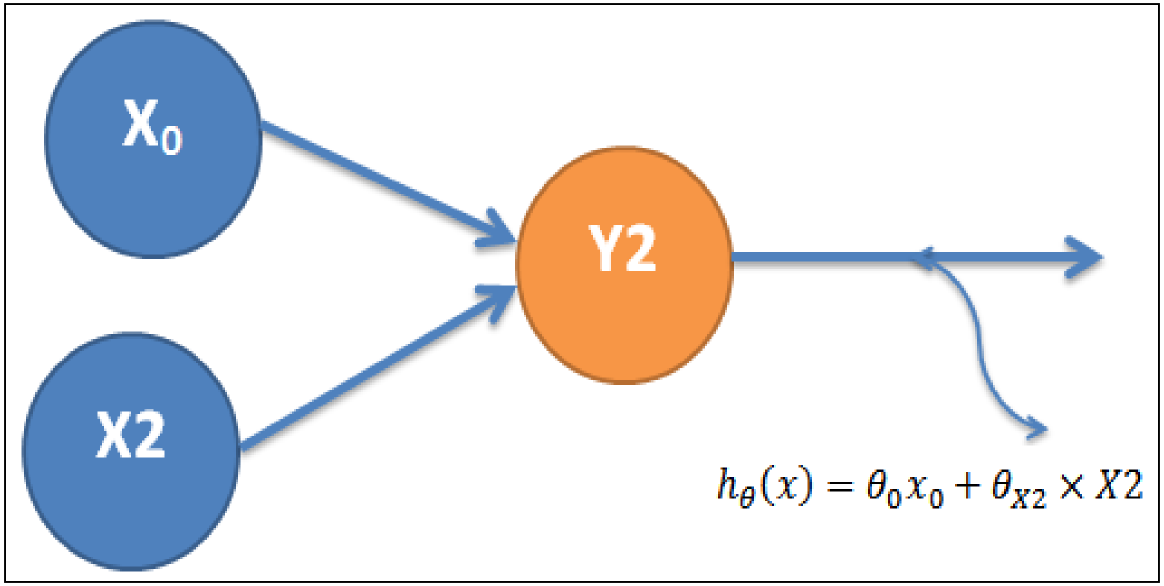
Linear Regression with One Variable(Outcome Cooling Loading)

Linear Regression with One Variable



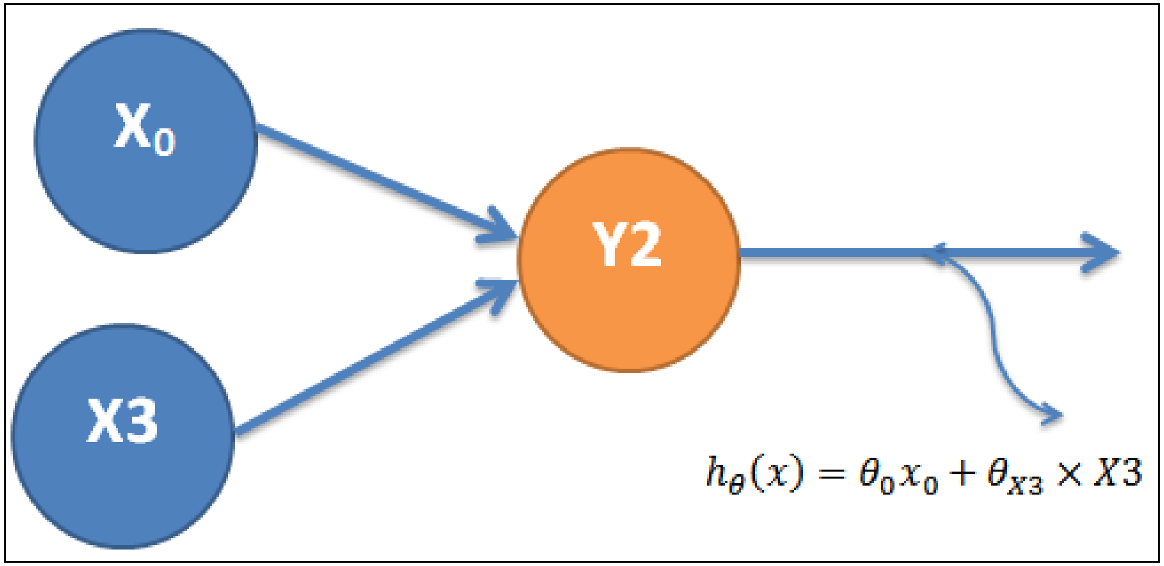
- Create two arrays: x (X1 Relative Compactness) and y (Y2 Cooling Load). Intuitively we'd expect to find some correlation between the two variables.

Linear Regression with One Variable



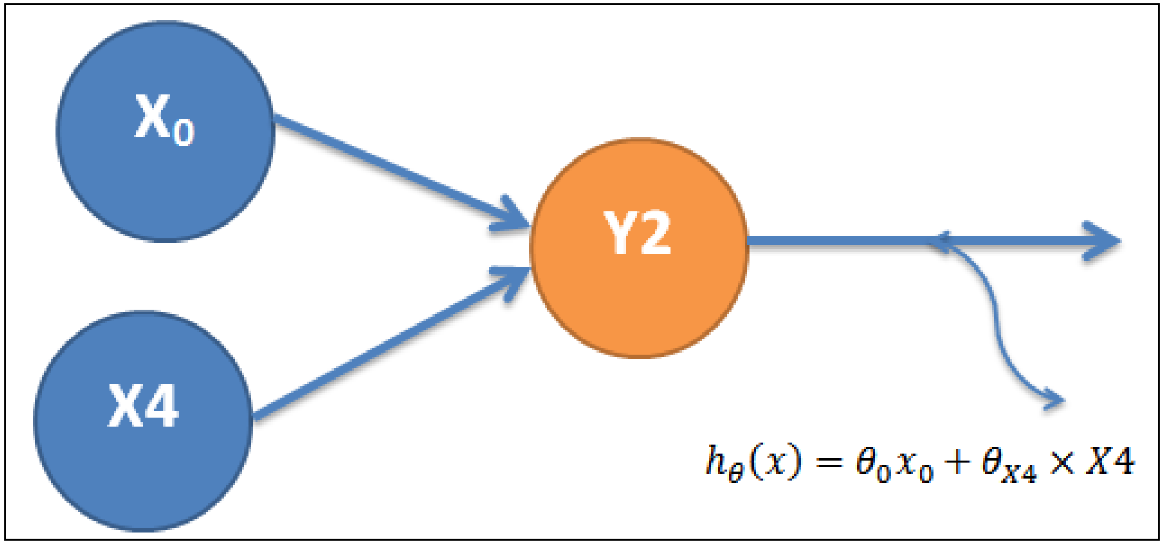
- Create two arrays: x (X2 Surface Area) and y (Y2 Cooling Load). Intuitively we'd expect to find some correlation between the two variables.

Linear Regression with One Variable



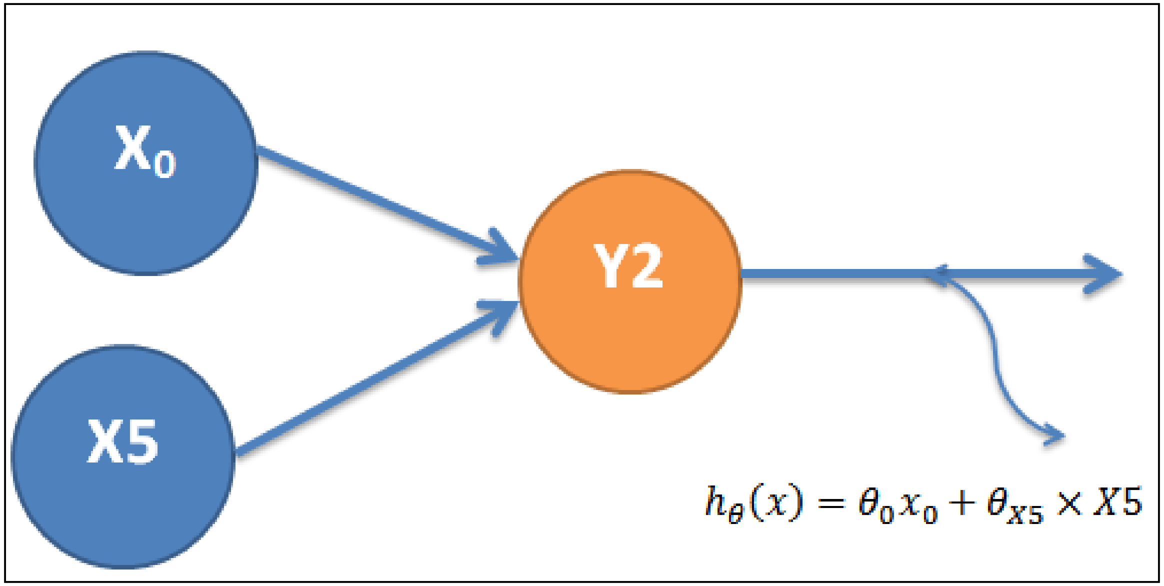
- Create two arrays: x (X3 Wall Area) and y (Y2 Cooling Load). Intuitively we'd expect to find some correlation between the two variables.

Linear Regression with One Variable



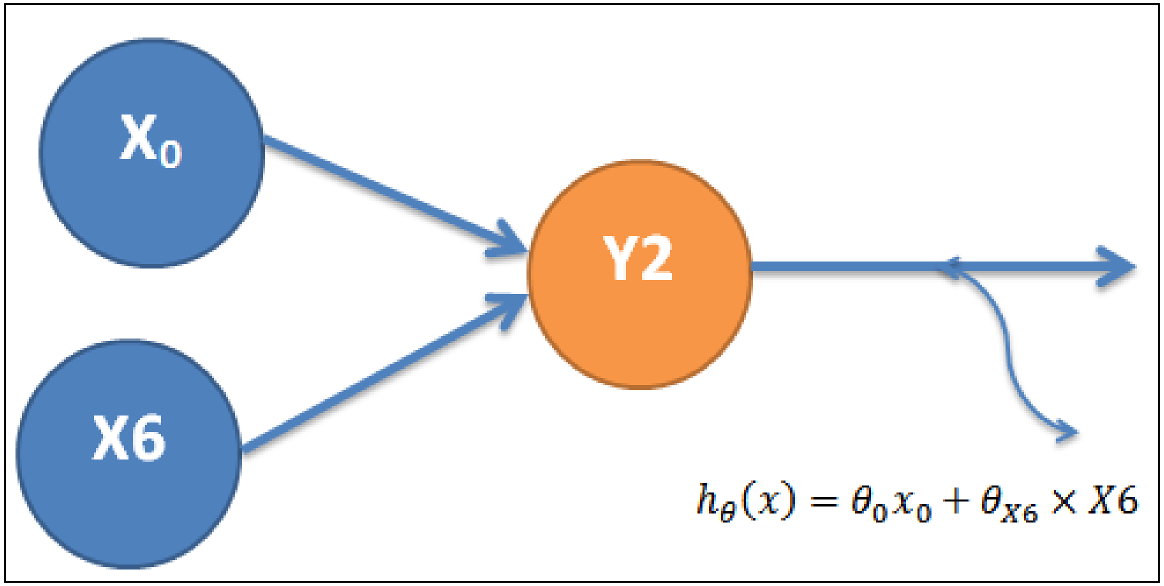
- Create two arrays: x (X4 Roof Area) and y (Y2 Cooling Load). Intuitively we'd expect to find some correlation between the two variables.

Linear Regression with One Variable



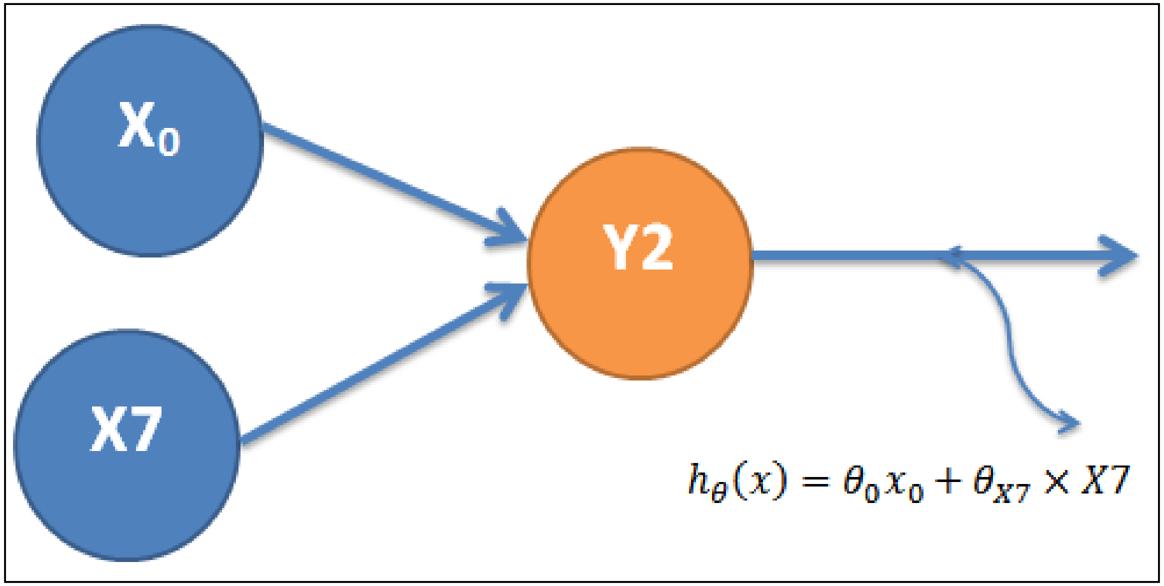
- Create two arrays: x (X5 Overall Height) and y (Y2 Cooling Load). Intuitively we'd expect to find some correlation between the two variables.

Linear Regression with One Variable



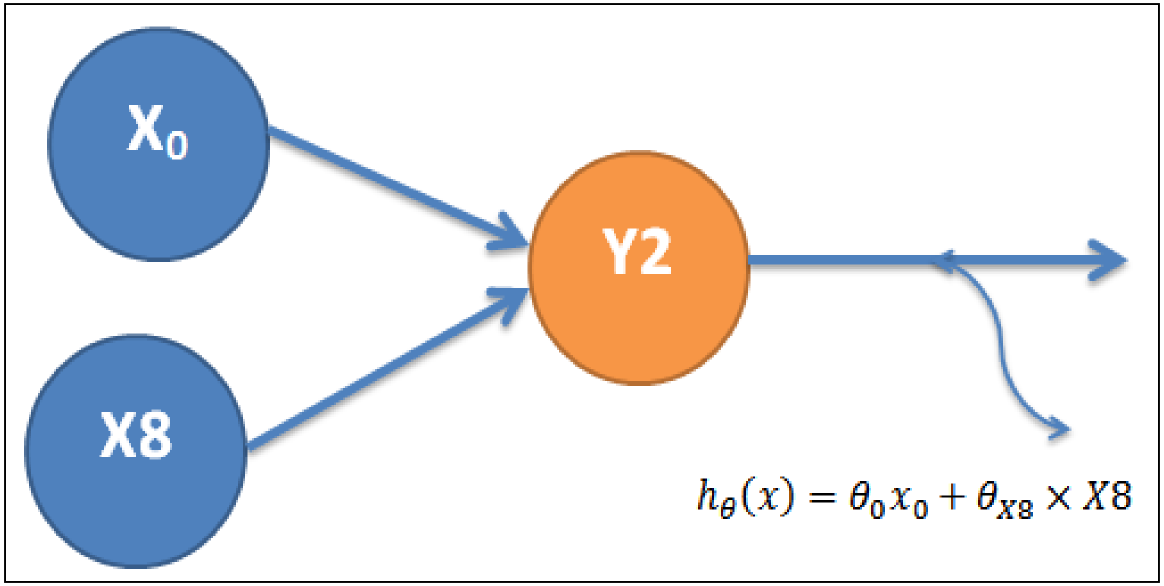
- Create two arrays: x (X6 Orientation) and y (Y2 Cooling Load). Intuitively we'd expect to find some correlation between the two variables.

Linear Regression



- Create two arrays: x (X7 Glazing Area) and y (Y2 Cooling Load). Intuitively we'd expect to find some correlation between the two variables.

Linear Regression



- Create two arrays: x (X8 Glazing Area Distribution) and y (Y2 Cooling Load). Intuitively we'd expect to find some correlation between the two variables.

Machine Learning

Linear Regression with Multiple Variables

Linear Regression with One Variable

- The first step is to import the required Libraries

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import scipy.stats as stats
```

- The second step is to load the dataset (**Energy Efficiency Data Set)**

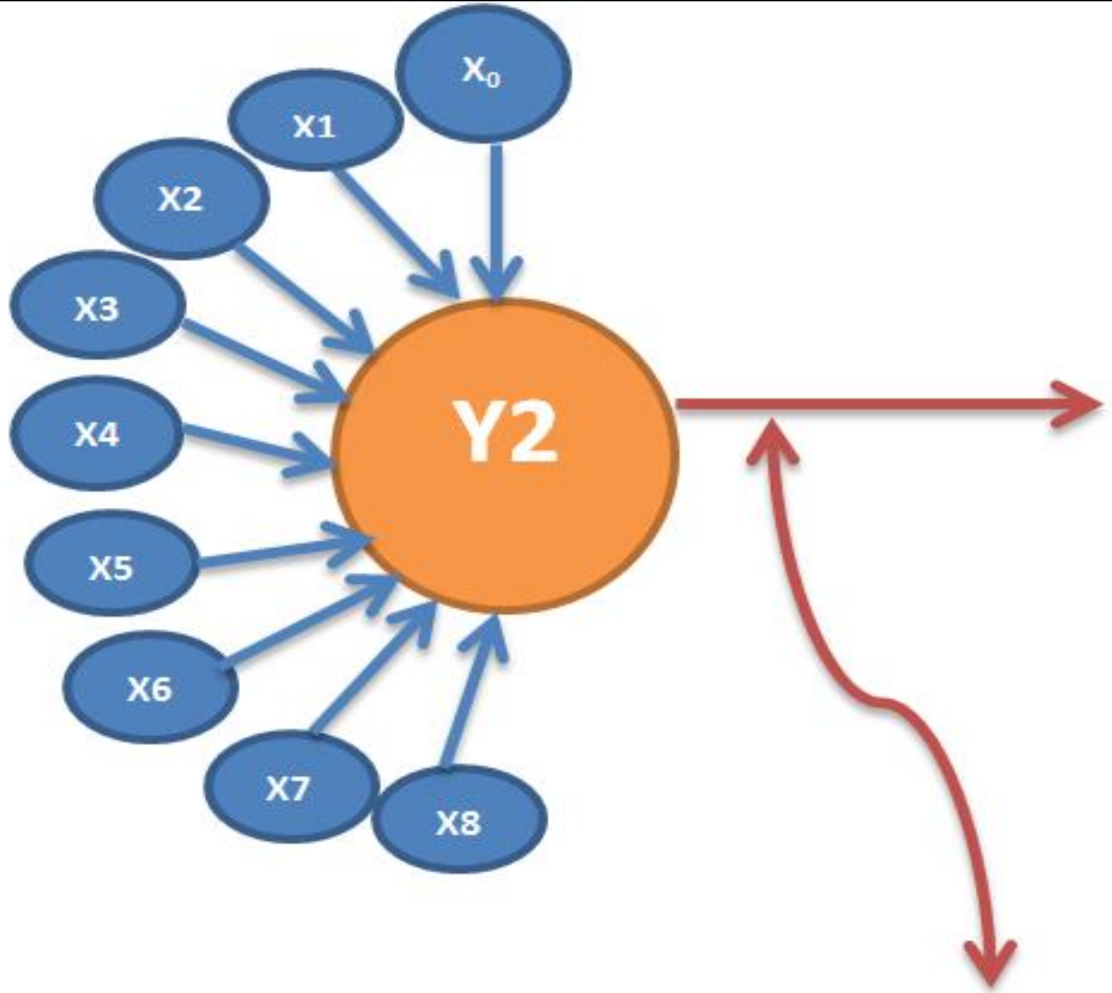
```
enb=pd.read_csv("ENB2012_data.csv")
```

- The data will be loaded using Python Pandas, a data analysis module. It will be loaded into a structure known as a Panda Data Frame, which allows for each manipulation of the rows and columns.

Machine Learning

Linear Regression with Multiple Variables (Outcome Heating Loading)

Linear Regression with Multiple Variables



$$h_\theta(x) = \theta_0 x_0 + \theta_{X1}X1 + \theta_{X2}X2 + \theta_{X3}X3 + \theta_{X4}X4 + \theta_{X5}X5 + \theta_{X6}X6 + \theta_{X7}X7 + \theta_{X8}X8$$

Linear Regression with Multiple Variables

- In next step, Splitting the dataset into the features set and Outcome set

```
x = enb.iloc[:, :8].values

y = enb.iloc[:, 8].values
```

- The data will be split into a training and test set. Once we have the test data, we can find a best fit line and make predictions.

```
from sklearn.cross_validation import train_test_split

x_train, x_test, y_train, y_test = train_test_split(x, y,test_size = 0.1,random_state= 0)
```

Linear Regression with Multiple Variables

- Fitting Multiple Linear Regression to the Training set

```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

- Predicting the Test set results

```
y_pred = regressor.predict(x_test)
```

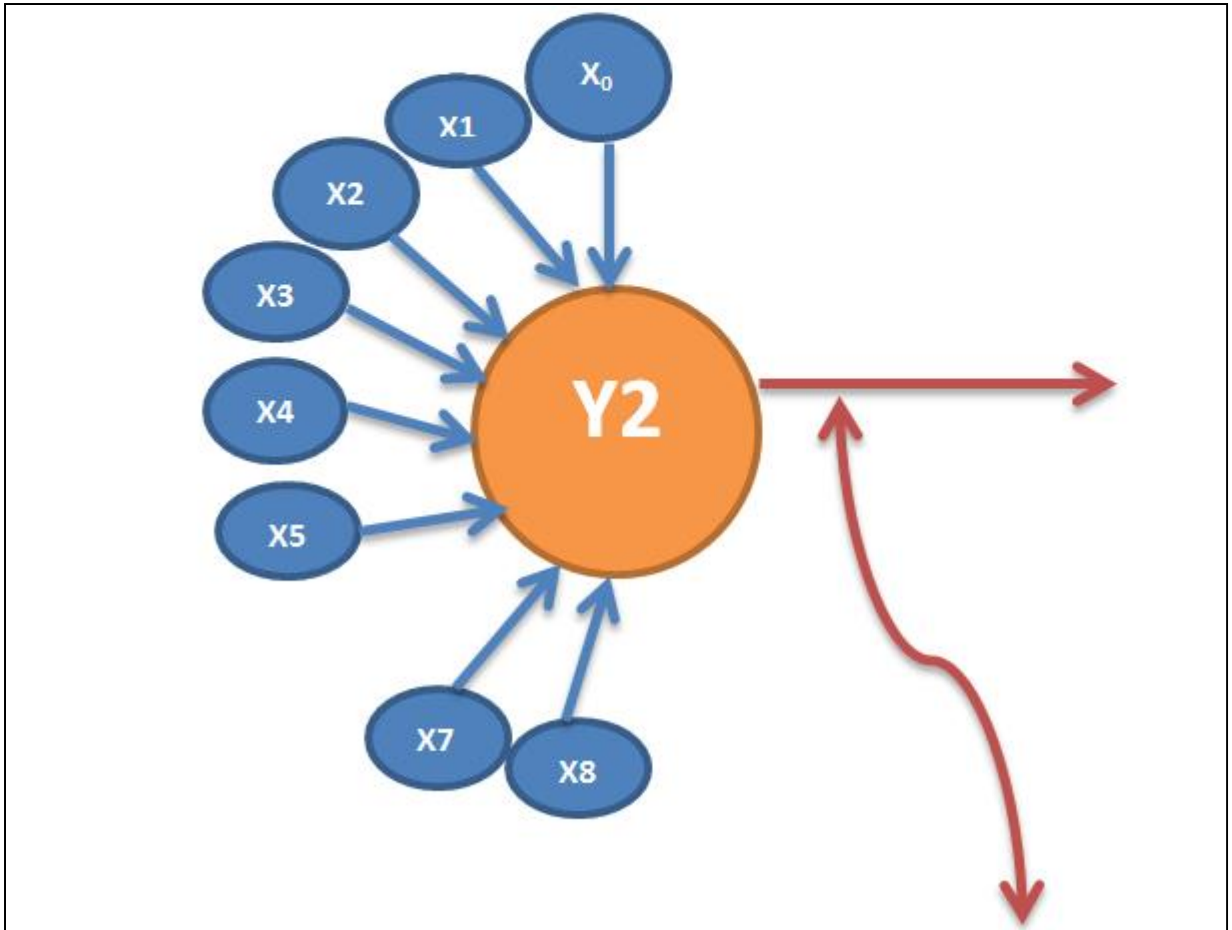
- Building the optimal model using Backward Elimination

```
import statsmodels.formula.api as sm
x= np.append(arr=np.ones((768,1)).astype(int),values=x,axis=1)
x_opt = x[:,[0,1,2,3,4,5,6,7,8]]
regressor_OSL = sm.OLS(endog=y, exog=x_opt).fit()
regressor_OSL.summary()
```

Linear Regression with Multiple Variables

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.916
Model:                            OLS   Adj. R-squared:                  0.915
Method:                 Least Squares   F-statistic:                     1187.
Date:                Sun, 22 Apr 2018   Prob (F-statistic):               0.00
Time:                        13:01:09   Log-Likelihood:                -1912.5
No. Observations:                 768   AIC:                             3841.
Df Residuals:                     760   BIC:                             3878.
Df Model:                           7
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         84.0145     19.034      4.414      0.000      46.650     121.379
x1           -64.7740     10.289     -6.295      0.000     -84.973     -44.575
x2            -0.0626      0.013     -4.670      0.000      -0.089      -0.036
x3             0.0361      0.004      9.386      0.000       0.029       0.044
x4            -0.0494      0.008     -6.569      0.000      -0.064      -0.035
x5             4.1699      0.338     12.337      0.000       3.506       4.833
x6            -0.0233      0.095     -0.246      0.805      -0.209       0.163
x7            19.9327      0.814     24.488      0.000      18.335      21.531
x8             0.2038      0.070      2.914      0.004       0.067       0.341
==============================================================================
Omnibus:                       18.648   Durbin-Watson:                   0.654
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               37.708
Skew:                           0.044   Prob(JB):                     6.48e-09
Kurtosis:                       4.082   Cond. No.                     3.34e+15
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 4.08e-23. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

Linear Regression with Multiple Variables



$$h_\theta(x) = \theta_0 x_0 + \theta_{X1}X1 + \theta_{X2}X2 + \theta_{X3}X3 + \theta_{X4}X4 + \theta_{X5}X5 + \theta_{X7}X7 + \theta_{X8}X8$$

Linear Regression with Multiple Variables

- Building the optimal model using Backward Elimination

```
import statsmodels.formula.api as sm
x= np.append(arr=np.ones((768,1)).astype(int),values=x,axis=1)
x_opt = x[:,[0,1,2,3,4,5,7,8]]
regressor_OSL = sm.OLS(endog=y, exog=x_opt).fit()
regressor_OSL.summary()
```

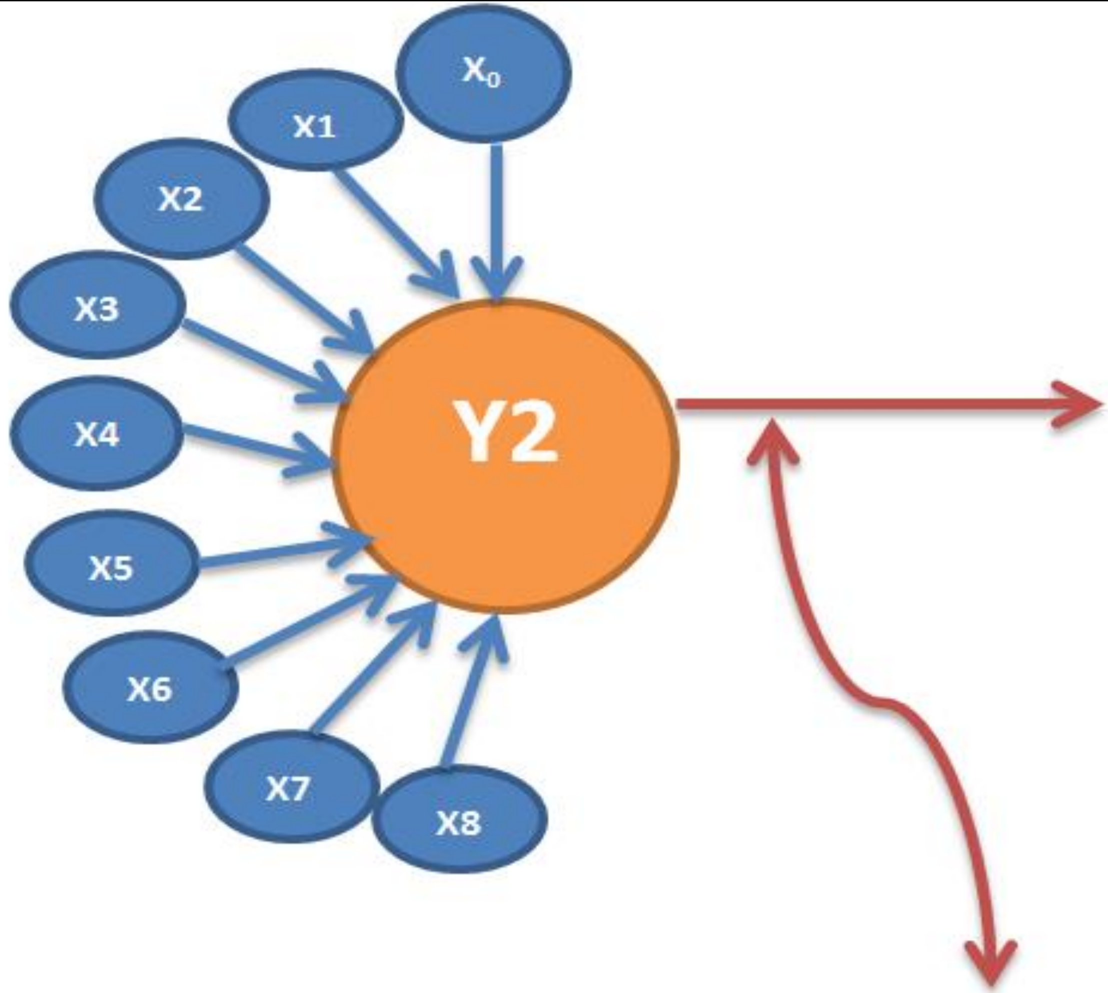
Linear Regression with Multiple Variables

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.916
Model:                            OLS   Adj. R-squared:                  0.916
Method:                 Least Squares   F-statistic:                     1387.
Date:                Sun, 22 Apr 2018   Prob (F-statistic):               0.00
Time:                        13:03:59   Log-Likelihood:                -1912.5
No. Observations:                 768   AIC:                             3839.
Df Residuals:                     761   BIC:                             3871.
Df Model:                           6
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         83.9329     19.019      4.413      0.000      46.597     121.269
x1           -64.7740     10.283     -6.299      0.000     -84.961     -44.587
x2            -0.0626      0.013     -4.673      0.000      -0.089      -0.036
x3             0.0361      0.004      9.392      0.000       0.029       0.044
x4            -0.0494      0.008     -6.573      0.000      -0.064      -0.035
x5             4.1699      0.338     12.345      0.000       3.507       4.833
x6            19.9327      0.813     24.503      0.000      18.336      21.530
x7             0.2038      0.070      2.916      0.004       0.067       0.341
==============================================================================
Omnibus:                       18.654   Durbin-Watson:                   0.654
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               37.740
Skew:                           0.044   Prob(JB):                     6.38e-09
Kurtosis:                       4.082   Cond. No.                     1.26e+16
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 2.85e-24. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

Machine Learning

Linear Regression with Multiple Variables (Outcome Cooling Loading)

Linear Regression with Multiple Variables



$$h_\theta(x) = \theta_0 x_0 + \theta_{X1} X1 + \theta_{X2} X2 + \theta_{X3} X3 + \theta_{X4} X4 + \theta_{X5} X5 + \theta_{X6} X6 + \theta_{X7} X7 + \theta_{X8} X8$$

Linear Regression with Multiple Variables

- In next step, Splitting the dataset into the features set and Outcome set

```
x = enb.iloc[:, :8].values

y = enb.iloc[:, 9].values
```

- The data will be split into a training and test set. Once we have the test data, we can find a best fit line and make predictions.

```
from sklearn.cross_validation import train_test_split

x_train, x_test, y_train, y_test = train_test_split(x, y,test_size = 0.1,random_state= 0)
```

Linear Regression with Multiple Variables

- Fitting Multiple Linear Regression to the Training set

```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

- Predicting the Test set results

```
y_pred = regressor.predict(x_test)
```

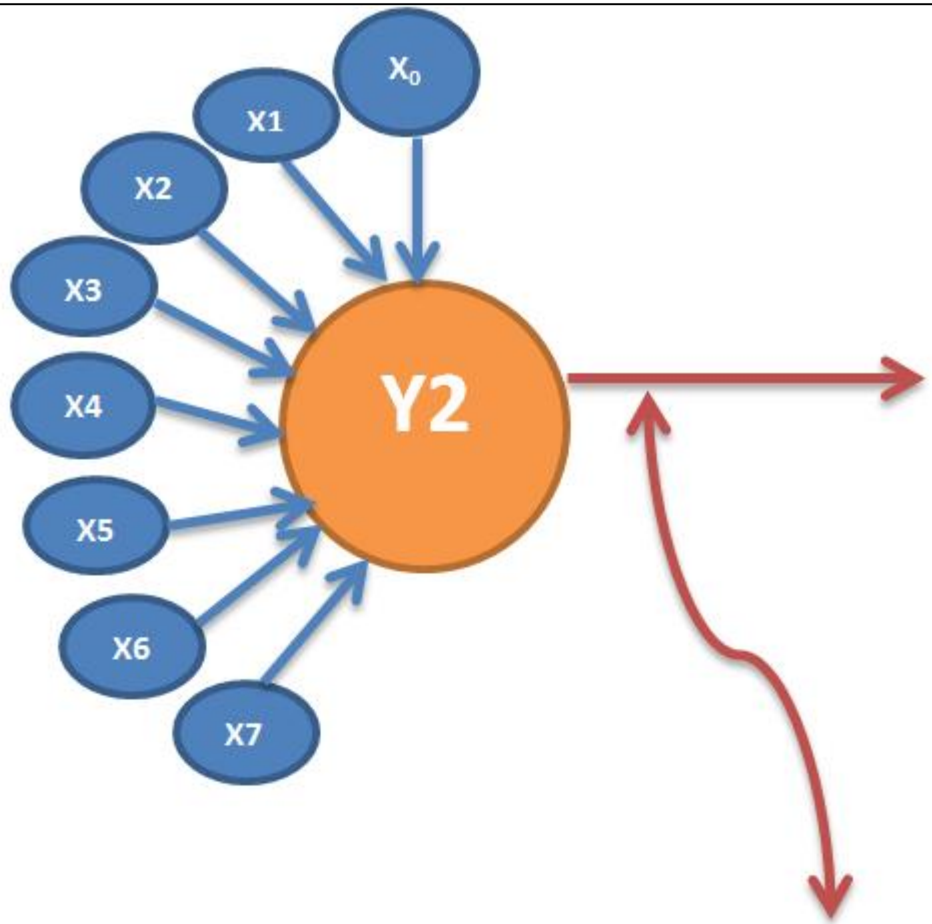
- Building the optimal model using Backward Elimination

```
import statsmodels.formula.api as sm
x= np.append(arr=np.ones((768,1)).astype(int),values=x,axis=1)
x_opt = x[:,[0,1,2,3,4,5,6,7,8]]
regressor_OSL = sm.OLS(endog=y, exog=x_opt).fit()
regressor_OSL.summary()
```

Linear Regression with Multiple Variables

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.888
Model:                            OLS   Adj. R-squared:                  0.887
Method:                 Least Squares   F-statistic:                     859.1
Date:                Sun, 22 Apr 2018   Prob (F-statistic):               0.00
Time:                        13:21:08   Log-Likelihood:                -1979.3
No. Observations:                 768   AIC:                             3975.
Df Residuals:                     760   BIC:                             4012.
Df Model:                           7
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         97.2457     20.765      4.683      0.000      56.483     138.009
x1           -70.7877     11.225     -6.306      0.000     -92.824     -48.751
x2            -0.0661      0.015     -4.519      0.000      -0.095      -0.037
x3             0.0225      0.004      5.365      0.000       0.014       0.031
x4            -0.0443      0.008     -5.404      0.000      -0.060      -0.028
x5             4.2838      0.369     11.618      0.000       3.560       5.008
x6             0.1215      0.103      1.176      0.240      -0.081       0.324
x7            14.7171      0.888     16.573      0.000      12.974      16.460
x8             0.0407      0.076      0.534      0.594      -0.109       0.190
==============================================================================
Omnibus:                      104.668   Durbin-Watson:                   1.094
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              230.547
Skew:                           0.767   Prob(JB):                     8.65e-51
Kurtosis:                       5.203   Cond. No.                     3.34e+15
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 4.08e-23. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

Linear Regression with Multiple Variables



$$h_\theta(x) = \theta_0 x_0 + \theta_{X1} X1 + \theta_{X2} X2 + \theta_{X3} X3 + \theta_{X4} X4 + \theta_{X5} X5 + \theta_{X6} X6 + \theta_{X7} X7$$

Linear Regression with Multiple Variables

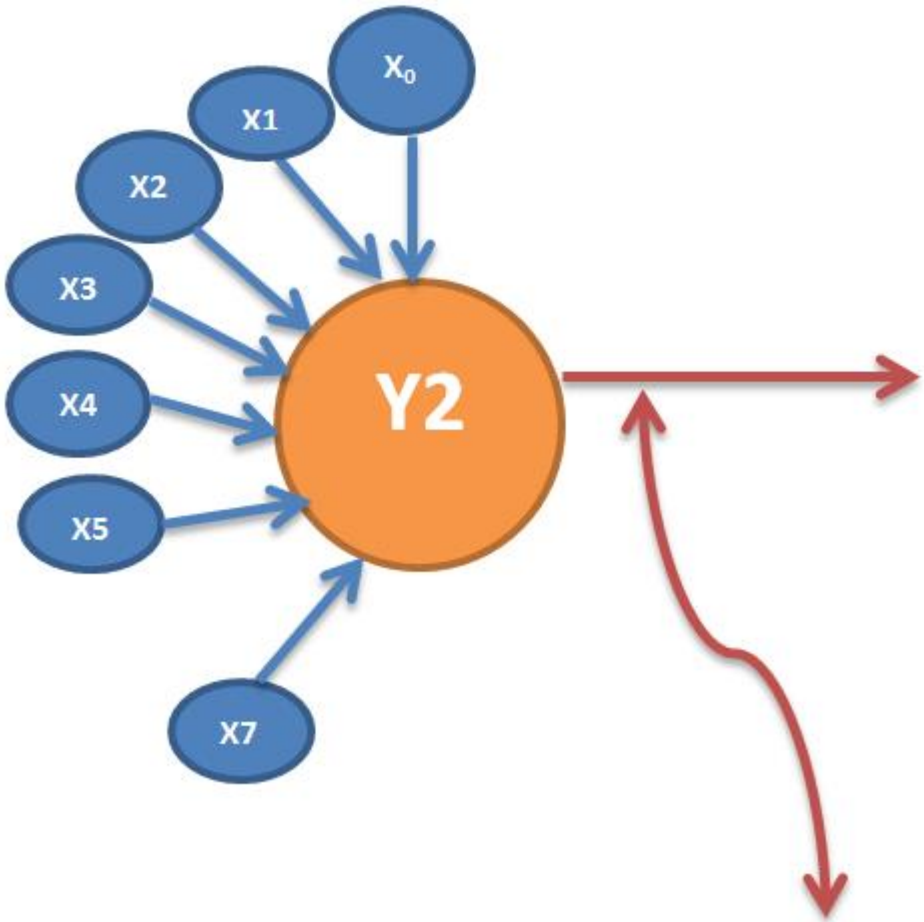
- Building the optimal model using Backward Elimination

```
import statsmodels.formula.api as sm
x= np.append(arr=np.ones((768,1)).astype(int),values=x,axis=1)
x_opt = x[:,[0,1,2,3,4,5,6,7]]
regressor_OSL = sm.OLS(endog=y, exog=x_opt).fit()
regressor_OSL.summary()
```

Linear Regression with Multiple Variables

```
                           OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.888
Model:                            OLS   Adj. R-squared:                  0.887
Method:                 Least Squares   F-statistic:                     1003.
Date:                Sun, 22 Apr 2018   Prob (F-statistic):               0.00
Time:                        13:29:24   Log-Likelihood:                -1979.5
No. Observations:                 768   AIC:                             3973.
Df Residuals:                     761   BIC:                             4005.
Df Model:                           6
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         97.3366     20.754      4.690      0.000      56.594     138.079
x1           -70.7877     11.220     -6.309      0.000     -92.814     -48.762
x2            -0.0661      0.015     -4.521      0.000      -0.095      -0.037
x3             0.0225      0.004      5.367      0.000       0.014       0.031
x4            -0.0443      0.008     -5.407      0.000      -0.060      -0.028
x5             4.2838      0.369     11.623      0.000       3.560       5.007
x6             0.1215      0.103      1.177      0.240      -0.081       0.324
x7            14.8180      0.867     17.086      0.000      13.116      16.520
==============================================================================
Omnibus:                      104.448   Durbin-Watson:                   1.093
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              230.721
Skew:                           0.764   Prob(JB):                     7.93e-51
Kurtosis:                       5.208   Cond. No.                     1.26e+16
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 2.85e-24. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

Linear Regression with Multiple Variables



$$h_\theta(x) = \theta_0 x_0 + \theta_{X1} X1 + \theta_{X2} X2 + \theta_{X3} X3 + \theta_{X4} X4 + \theta_{X5} X5 + \theta_{X7} X7$$

Linear Regression with Multiple Variables

- Building the optimal model using Backward Elimination

```
import statsmodels.formula.api as sm
x= np.append(arr=np.ones((768,1)).astype(int),values=x,axis=1)
x_opt = x[:,[0,1,2,3,4,5,7]]
regressor_OSL = sm.OLS(endog=y, exog=x_opt).fit()
regressor_OSL.summary()
```

Linear Regression with Multiple Variables

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.888
Model:                            OLS   Adj. R-squared:                  0.887
Method:                 Least Squares   F-statistic:                     1203.
Date:                Sun, 22 Apr 2018   Prob (F-statistic):               0.00
Time:                        13:33:40   Log-Likelihood:                -1980.2
No. Observations:                 768   AIC:                             3972.
Df Residuals:                     762   BIC:                             4000.
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         97.7618     20.756      4.710      0.000      57.015     138.508
x1           -70.7877     11.223     -6.307      0.000     -92.819     -48.756
x2            -0.0661      0.015     -4.520      0.000      -0.095      -0.037
x3             0.0225      0.004      5.366      0.000       0.014       0.031
x4            -0.0443      0.008     -5.405      0.000      -0.060      -0.028
x5             4.2838      0.369     11.620      0.000       3.560       5.008
x6            14.8180      0.867     17.082      0.000      13.115      16.521
==============================================================================
Omnibus:                      104.896   Durbin-Watson:                   1.095
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              232.225
Skew:                           0.766   Prob(JB):                     3.74e-51
Kurtosis:                       5.215   Cond. No.                     1.26e+16
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 2.85e-24. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```