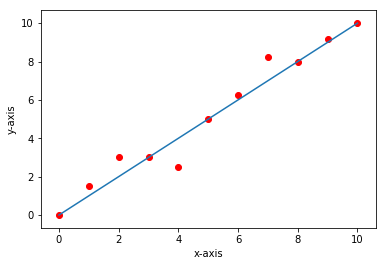
Linear Regression for Newbies



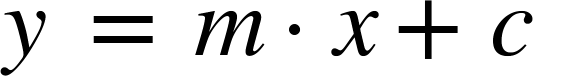
**The Theory**

Linear Regression is the process of fitting a line to the dataset.

**Single Variable Linear Regression**

**The Mathematics**

The equation of Line is



Where,

y = dependent variable

X = independent variable

C = intercept

The algorithm is trying to fit a line to the data by adjusting the values of m and c. Its Objective is to attain to a value of m such that for any given value of x it would be properly predicting the value of y.

There are various ways in which we can attain the values of m and c

1. Statistical approach
2. Iterative approach

**The Dataset**

Dataset consists of two columns namely X and y

Where

For Auto Insurance in Sweden dataset

X = number of claims

Y = total payment for all the claims in thousands of Swedish Kronor

The data is taken from Swedish Committee on Analysis of Risk Premium in Motor Insurance

[Link to All Datasets](http://college.cengage.com/mathematics/brase/understandable_statistics/7e/students/datasets/slr/frames/frame.html)

**The Code**

The Code was written in three phases

1. Data preprocessing phase
2. Training
3. Prediction and plotting

The data preprocessing phase

|  |
| --- |
| **#Imports**  **import** numpy **as** np **import** pandas **as** pd **import** matplotlib.pyplot **as** plt |

Numpy import for array processing, python doesn’t have built in array support. The feature of working with native arrays can be used in python with the help of numpy library.

Pandas is a library of python used for working with tables, on importing the data, mostly data will be of table format, for ease manipulation of tables pandas library is imported

Matplotlib is a library of python used to plot graphs, for the purpose of visualizing the results we would be plotting the results with the help of matplotlib library.

|  |
| --- |
| #Reading the dataset from data dataset = pd.read\_excel(r'data\\slr09.xls') |

In this line of code using the read\_excel method of pandas library, the dataset have been imported from data folder and stored in dataset variable.

On visualising the dataset , it contains of two columns X and Y where X is dependent variable and Y is Independent Variable



X is an independent varible

Y is dependent variable Inference

For x-value of 7.6 ,157 y-value

for x-value of 7.1 ,174 y-value

And goes on

|  |
| --- |
| #Creating Dependent and Independent variables dataset = dataset.sort\_values('X')  X = dataset['X'].values  y = dataset['Y'].values |

Sorting is done for better visualization of data. The X Column from the dataset is extracted into an X variable of type numpy, similarly the y variable.



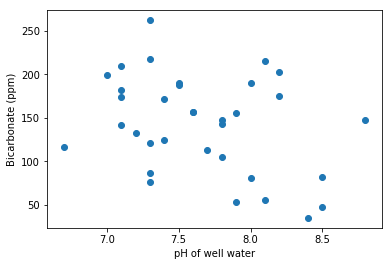
On input 10 it would result in a pandas Series object

So, values attribute is used to attain an numpy array

|  |
| --- |
| #Visualizing the data  plt.scatter(X,y)  plt.xlabel('pH of well water')  plt.ylabel('Bicarbonate (ppm)')  plt.show() |

The step is to just see how the dataset is

On visualization the data would appear something like this



Each point on the plot is a data point showing the respective ph value on x-axis and bicarbonate value on y-axis

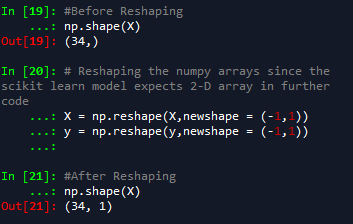
Need to change

|  |
| --- |
| # Splitting the data into training set and test set  **from** sklearn.model\_selection **import** train\_test\_split  X\_test,X\_train,y\_test,y\_train = train\_test\_split(X,y, test\_size = 0.8) |

We are splitting the whole dataset into training and test set where training set is used for fitting the line to data and test set is used to check how good the line if for the data.

|  |
| --- |
| # Reshaping the numpy arrays since the scikit learn model expects 2-D array in further code X\_train = np.reshape(X\_train,newshape = (-1,1))  y\_train = np.reshape(y\_train,newshape = (-1,1))  X\_test = np.reshape(X\_test,newshape = (-1,1))  y\_test = np.reshape(y\_test,newshape = (-1,1)) |

In further the scikit learn model would be expecting a 2-D array of shape (length,1)



The code was just to convert a single dimensional array into a 2-D array where each element is an array

The Training phase

|  |
| --- |
| #Importing the linear model from sklearn framework **from** sklearn.linear\_model **import** LinearRegression lr = LinearRegression() lr.fit(X = X\_train, y = y\_train) |

From scikit learn Library LinearRegression is imported. Lr is an object of LinearRegression.

The process of training is done in the fit method, our dependent and independent variable are fed into to the fit method in which it would try to fit a line to the data provided.

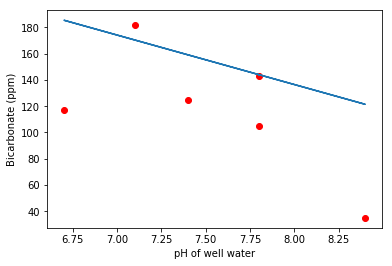
The Prediction phase

|  |
| --- |
| #Predicting the results  y\_pred = lr.predict(X\_test) |

By the trained linear regression model we are trying to predict the values of test data. Y\_pred variable contains all the predicted bicarbonate values of the test ph values.

|  |
| --- |
| #Visualizing the results  plt.scatter(X\_test,y\_test,c='red')  plt.plot(X\_test,y\_pred)  plt.xlabel('pH of well water')  plt.ylabel('Bicarbonate (ppm)')  plt.show() |

As we have predicted the bicarbonate values for a set of ph values we are visualizing the results to check how good did our line fit for our predictions.



The plot shows the red points are the data points are actual ph values where the blue line is the predictions