Understanding Linear Regression and its implementation in Python

This is first part of Machine Learning Blog series. Initially we will also cover basics of Linear Algebra and Geometry. It is expected that Reader has a basic knowledge of Python programming or any object oriented programming can serve the purpose here.

This blog is on Linear Regression right from scratch.

**NOTE:**

You may find the dataset, “Advertising.csv” and code, “Linearregression\_blog.py” for this blog at:

<https://github.com/Thisisretarded/DataSc/tree/master/LinearRegression>

## What is Linear Regression?

Let us consider the following dataset.. It contains information of total sales made by xyz company along with total money spent on marketing team on various advertising platforms like TV, radio, newspaper etc. per month

>>> # importing the libraries

>>> import pandas as pd

>>> import numpy as np

>>> # importing the dataset

>>> dataset = pd.read\_csv("Advertising.csv")

>>> print(dataset.head())

Unnamed: 0 TV radio newspaper sales totalAdvt

0 1 230.1 37.8 69.2 22.1 337.1

1 2 44.5 39.3 45.1 10.4 128.9

2 3 17.2 45.9 69.3 9.3 132.4

3 4 151.5 41.3 58.5 18.5 251.3

4 5 180.8 10.8 58.4 12.9 250.0

>>> # Adding these 3 columns to get total advt budget

>>> dataset[‘Advt\_Budget’] = dataset['TV'] + dataset['radio'] + dataset['newspaper']

>>> df = dataset[['Advt\_Budget', 'sales']]

>>> X=df.iloc[:, 0:1] # X will be independent variable or feature

>>> y=df.iloc[:, 1:] # y will be dependent variable or response

>>> print(df.head())

Advt\_Budget sales

0 337.1 22.1

1 128.9 10.4

2 132.4 9.3

3 251.3 18.5

4 250.0 12.9

Suppose that you are in the analytics team of xyz company and are asked to give some inputs on a marketing plan for next year on the basis of this data.

First we should address the following questions:

1. Is there any relationship between advertising budget and Sales?
   1. This should be our first goal to identify if this relation exist? If not, we may suggest to reduce or minimise the advertising budget.
2. If at all there is a relationship, How strong is that?
   1. Strength of a relationship is that we should be able to predict the sales, given advertising budget of a particular month with a higher accuracy.
   2. If It is nothing more than a random guess, we would say, the relationship is not strong but weak.
3. How accurately can we predict the future sales?
4. Is the relationship linear?
   1. When we plot a graph on a x-y axis coordinates, it should represent straight-line like characteristic. Then the relationship can be treated as a Linear relationship.
   2. If not, then we may not use Linear regression algorithm here.

It turns out that all of the above questions can be answered using Linear regression.

### Simple Linear Regression

It is a very straightforward approach for predicting the dependent variable Y (Sales in this case) from *a single* predictor or *a single* independent variable X(Advertising budget).

It assumes that there is approximately a linear relationship between X and Y . Mathematically this relationship can be represented as:

Y ~ m X + c ------ (i)

It says Y is approximately modeled as m X + c . Or sometimes we will say that we are regressing Y on X . Here X represent the advertising budget and Y represents total sales. Then we cam

Hence we can write the above equation as:

Sales = m x Advt\_Budget + c ------- (ii)

In Practice m and c

Our goal is to find m and c here so that we can calculate Sales by substituting Advt\_Budget in the above equation.

Now we have a Panda’s dataframe with 2 columns namely: ‘Advt\_Budget’ and ‘sales’.

We will fit this dataframe into a class known as Linear regression in sklearn library and that would be known as ‘Training’ of our model.

Before fitting this dataframe, “df”, we will first split this data into 60-40 or 70-30 % ratio and the smaller portion of this data will be used to test our trained model. The bigger portion of this split will be used to train out Model.

Also to split this data frame, we have a method, train\_test\_split, inside sklearn.model\_selection package.

>>> # Splitting the dataset into the Training set and Test set

>>> from sklearn.model\_selection import train\_test\_split

>>> X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 1/4, random\_state = 0, )

**X\_train ----> 75% of ‘Advt\_Budget’ column, will be used for training our model.**

**X\_test ----> 25% of ‘Advt\_Budget’ column, will be used for testing our model’s accuracy.**

**Y\_train ---->75% of ‘sales’ column, will be used for training our model.**

**y\_test ----> 25% of ‘sales’ column, will be used for testing our model’s accuracy.**

Here X is ‘Advt\_Budget’ column, y is ‘sales’ column. test\_size is the ratio in which we need to divide the train and test data. Here ‘test\_size =¼’ means that the size of test data will be 25% of total dataset.

>>> # Fitting Simple Linear Regression to the Training set

>>> from sklearn.linear\_model import LinearRegression # Class for Linear Regression

>>> regressor = LinearRegression()

>>> regressor.fit(X\_train,y\_train) # .fit method to fit the training data in our model

>>> y\_pred = regressor.predict(X\_train) # .predict method to predict the value of sales on training X column

>>> B0 = regressor.coef\_ #After fitting the taining set, coef\_ is the m in eqn (i)

>>> B1 = regressor.intercept\_ # .intercept\_ is c in eqn (i)

>>> # calculating Statistic value

>>> xtrainScore = regressor.score(X\_test, y\_test)\*100 # .score gives the score of test data on the model

>>> print("Eqn is" + " Sales = TV X " + str(B1) +" + " + str(\*B0))

Eqn is Sales = TV X [0.04725206] + [4.575039446059455]

>>>print("Score of prediction: " + str(xtrainScore))

Score of prediction: 81.17280326648255

**LinearRegression ----> class to implement Linear Regression**

**.fit(X\_train,y\_train) ----> method to fit data to train a Linear regression model**

**.predict(X\_train) ----> returns a predicted sequence of y**

**.coef\_ ----> coefficient of eqn (i)**

**.intercept\_ ----> intercept of eqn (i)**

**.score ----> Returns the coefficient of determination R^2 of the prediction**

Here higher the score will lead us to more accurate predictions.

Now lets plot the dataset with predictions

>>> # importing pyplot module from matplotlib package

>>> import matplotlib.pyplot as plt

>>> # Visualising the Training set results

#setting plots and feeding data

>>> plt.scatter(X\_train, y\_train, color = 'red') # Plotting a Scatter plot of trainning data

>>> plt.plot(X\_train, regressor.predict(X\_train), color = 'blue') # Line plot of

>>> plt.grid()

# setting labels and title

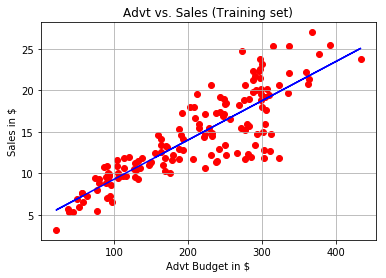
>>> plt.title('Advt vs. Sales (Training set)')

>>> plt.xlabel('Advt Budget in $')

>>> plt.ylabel('Sales in $')

# .show() to show the plot

>>> plt.show()



>>> # Visualising the Test set results

>>> y\_pred = regressor.predict(X\_test)

# Setting plots and feeding data

>>> plt.scatter(X\_test, y\_test, color = 'red')

>>> plt.plot(X\_test, y\_pred, color = 'blue')

>>> plt.grid()

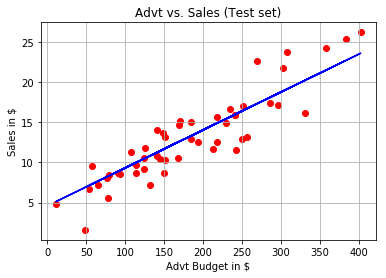
# setting labels and title

>>> plt.title('Advt vs. Sales (Test set)')

>>> plt.xlabel('Advt Budget in $')

>>> plt.ylabel('Sales in $')

>>> plt.show()



### Accessing the Accuracy of the Model

The quality of a linear regression fit is typically assessed using two related quantities: the residual standard error (RSE) and the R2 statistic.

[R-squared](https://statisticsbyjim.com/glossary/r-squared/) is a goodness-of-fit measure for linear regression models. This statistic indicates the percentage of the variance in the dependent variable that the independent variables explain collectively. R-squared measures the strength of the relationship between your model and the dependent variable on a convenient 0 – 100% scale.

After fitting a linear regression model, you need to determine how well the model fits the data. Does it do a good job of explaining changes in the dependent variable? There are several key goodness-of-fit statistics for regression analysis.

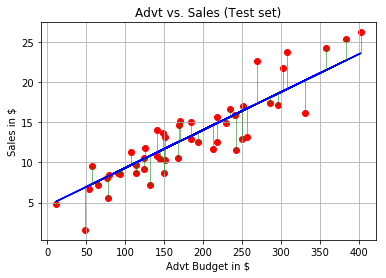
R-Squared is given by ------ (iii)

Let us understand How to calculate RSS first:

Let be the prediction for y based on ith value of X for each value of i.

Then represents the ith residual—this is the difference between the ith observed response value and the ith response value that is predicted by our linear model. We define the residual sum of squares (RSS) as:

----- (iv)



As per the above diagram, RSS for our model fit will be sum of squared values of all the green lines marked.

Now let us understand what is TSS:

------ (v)

Where is the ith response value in

For our dataset as per calculations(using eqn (iv) and (v)), RSS = 265.327 and TSS = 1409.275.

Now since we have RSS and TSS, R-squared value can be calculated using eqn (iii).

Which comes out to be **81.1728** *which comes out to be exactly the* ***same as we got from regressor.score()*** *function of our LinearRegression class.*