Linear Regression Assignment

aritra.dasray.160

October 2023

Contents

1 Introduction

In this project we have a data-set called diabetes.csv. It has 2 columns x and y, where x is an independent Variable and y is a dependent variable. We are trying to plot a best fit line that best describes the relationship between x and across the entire data-set. We aim to do that with the help of a method called Linear Regression.

2 Aims and Objectives

2.1 Aim

- Write a python program that calculates the linear regression parameters of the given data-set and plot a graph of the data-set and the best fit line.
- Validate the best fit line against another plot of linear regression parameter derived from the SciKit-Learn Library.

2.2 Objectives

1. Read the given data-set into the python program.
2. Calculate the Linear Regression Parameters. Use the Parameters to plot the graph of data-points and the best fit line.
3. Calculate new Linear regression Parameters using the SciKit-Learn Library and plot the best fit lines using these parameters.
4. Compare the Programmatically Calculated plot and Library Generated Plot to validate the results.

3 Methodology

Linear regression is a Statistical Method that is used to describe the relationship between an independent variable and one or more dependent variable by fitting a linear equation to the data set.The simplest form of linear regression is a linear equation with one independent variable which takes the form

$$y = \beta.x + \alpha \tag{1}$$

where

- x is the independent variable
- y is the dependent variable
- α is the intercept made by the line
- β is the slope of the line

α and β is called the parameters of the linear regression. Our goal is to optimise α and β such that the equation describes the relationship between all the points in the data-set. This optimization is often done by a method called Ordinary Least Squares (OLS) Method where we try to minimize the sum of squared differences between the observed and predicted values

4 Mathematical Understanding of OLS

Realistically, all data sets have noise in them and as such the we represent this noise in our linear equation by introducing an error term ϵ So our equation now looks like this:

$$y_i = \beta.x_i + \alpha + \epsilon \tag{2}$$

where

- x_i independent variable for the i-th data point.
- y_i dependent variable for the i-th data point.

or

$$\epsilon = y_i - \beta.x_i - \alpha \tag{3}$$

Now, Our objective is to minimize the square of the error term. Thus our Objective Function which is

$$Minimize(y_i - \beta.x_i - \alpha)^2 \tag{4}$$

Calculus is employed to find the values of α and β that minimize the objective function. The minimization process involves taking partial derivatives of the objective function with respect to α and β.

$$\frac{\partial}{\partial \alpha}\left(\sum_{i=1}^{n}(y_i - \alpha - \beta x_i)^2\right) = -2\sum_{i=1}^{n}(y_i - \alpha - \beta x_i) \tag{5}$$

$$\frac{\partial}{\partial \beta}\left(\sum_{i=1}^{n}(y_i - \alpha - \beta x_i)^2\right) = -2\sum_{i=1}^{n}x_i(y_i - \alpha - \beta x_i) \tag{6}$$

These derivatives gives the rate of change of the objective functions with respect to α and β.

The critical points (minima) of the objective function occur when its derivatives are equal to zero. Setting both derivatives to zero gives a system of equations:

$$\sum_{i=1}^{n}(y_i - \alpha - \beta x_i) = 0 \tag{7}$$

$$\sum_{i=1}^{n}x_i(y_i - \alpha - \beta x_i) = 0 \tag{8}$$

On solving these equation we get the values of α and β that minimize the objective function.

Thus we have the value of α and β as

$$\alpha = \bar{y} - \beta\bar{x} \tag{9}$$

$$\beta = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})} \tag{10}$$

where $\bar{x}$ and $\bar{y}$ are the means of x_i and y_i respectively.

Now, we can use the values of α and β to plot a graph for equation 1

5 Code

```
# Declare Library:
import pandas as pd
import numpy as np
import matplotlib.pyplot as plot
from sklearn import linear_model as LR

# read file and store in Variable Data
file_path = <path>

data = pd.read_csv(file_path, header = None)

headers = ['x', 'y']
data.columns = headers

#Take 2 variable and store data as list in those 2 variables
x_columnlist , y_columnlist = list(data['x']), list(data['y'])

#calculate Mean
x_bar = np.mean(x_columnlist)
y_bar = np.mean(y_columnlist)

#calculate slope and intercept
slope = sum((x-x_bar)*(y-y_bar) for x,y in zip(x_columnlist,y_columnlist))/sum((x-x_bar)**2 for x in x_columnlist)
intercept = y_bar - (slope*x_bar)

#create linear regression visualization
plot.style.use('_mpl-gallery')

#setting up the Visualization
fig, ax = plot.subplots()
fig.set_size_inches(8,6)
ax.set_xlim(-0.150, 0.200)
ax.set_ylim(0, 400)
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_title('Linear regression')

#Visualizing the datapoints in diabetes.csv
```

```
ax.scatter(x_columnlist, y_columnlist, label = 'data
    point', color = 'green')

#Visualizing the Linear Regression Line for the data

##Creating plotting dataset
x_values = np.linspace(-0.100, 0.150, 250)
y_values = slope * x_values + intercept

##Plotting Linear regression line
ax.plot(x_values, y_values, label='calculated
    regression line', color='red' )

#Validating Result against predefined Linear
    Regression Library in SciKit-Learn
X = np.array(x_columnlist).reshape(-1,1) ##Input
    pameter in Linear Regression function
model = LR.LinearRegression() ##Creating LR Model
model.fit(X, y_columnlist) ##fitting Model with data
##Plotting Linear regression Plot from Scikit Learn
ax.plot(x_columnlist, model.predict(X),
    label='Validation line', color = 'blue')
plot.show()
```

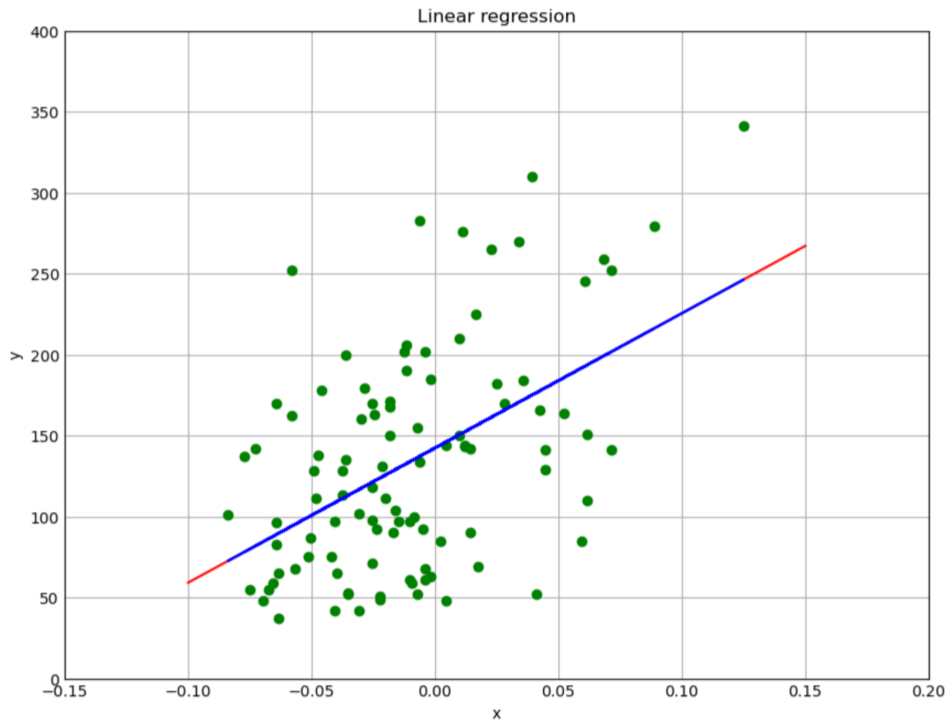

Figure 1: Linear Regression Visualization

6 Result, Validation and Discussion

6.1 Result

Here the Green Dots represents to the data points in the data-set. The Red Line represents the Programmatically Calculated plot. The Blue Line represents the plot generated by the SciKit-Learn library.

6.2 Validation

As observed the Red Line and Blue Line is overlapping. This means that the results generated by my calculations is accurate as it matches the plot generated by pre-existing Library.

6.3 Discussions and further investigations

There are several other methods available for perform Linear Regression on a given data-set. Since the data-set in this case was fairly small the ordinary least square method can easily plot the required graph. For larger data-sets, stochastic gradient descent method may be preferred.