Machine Learning

Spring 2016

Submitted to

Gady Agam

By

Harsh Parikh
A20338453

Problem Statement

In this assignment, I am trying to implement Single Variate and Multi-Variate Parametric Regression. I am using different data sets, polynomial values, ratio of training and testing data, trying higher dimensions and iterative & kernel methods. And analyzing differences by comparing different results.

Proposed Solution

First of all, I am scrapping the data from website and making it useful as per I want. I have implemented all methods from scratch in Python and got the results by doing Matrix Manipulations and found Training and Testing errors using iterative method. I have used MSE (Mean Squared Error) to measure errors for all datasets. For multi-variate, I have used Gradient Descent to implement iterative solution. At the end, I am making graphs of both datasets for each file.

At last, I am running **sklearn and scipy** Library to check the validity of results.

Implementation Details

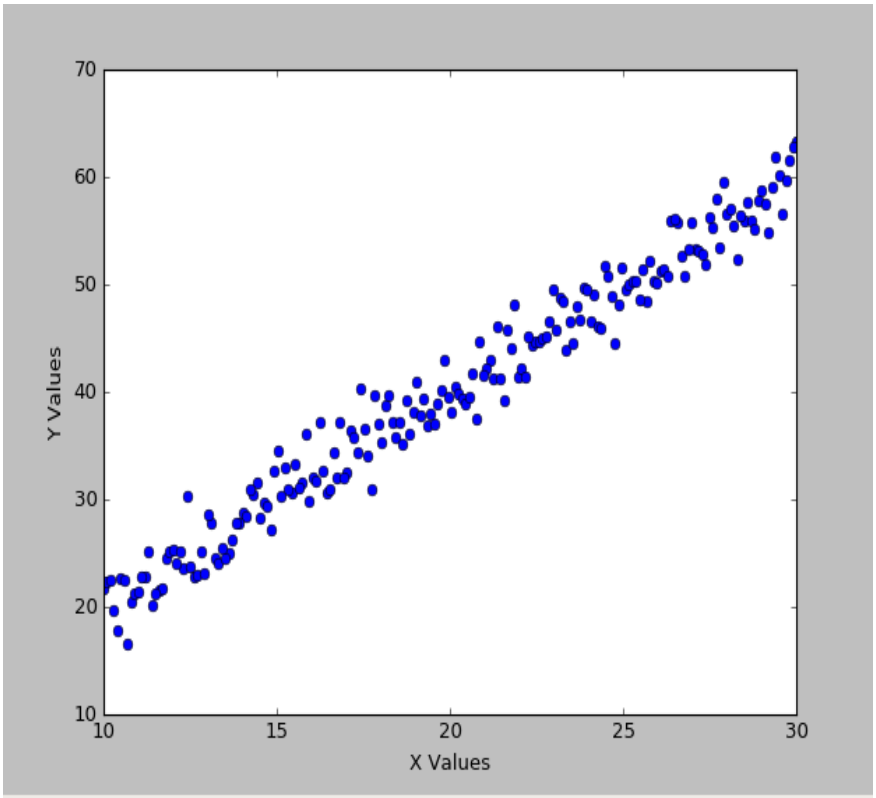
For implementation, I encapsulated all the work in a modular way I.e made methods for every functionality. While doing matrix manipulations, I was facing problems of complexity because I was looping (n^2) times then I resolved this problem by using **numpy** array and all manipulations have become very easy. Also faced the problem with Gaussian Kernel method for dual regression.

While running program, a user will be asked to enter File Number, Polynomial Value and number of Folds to enter. And based on that, all values and graphs will be populated. Folds are actually for dividing datasets into training and testing parts.

Results and Discussion

Loading and plotting Data:

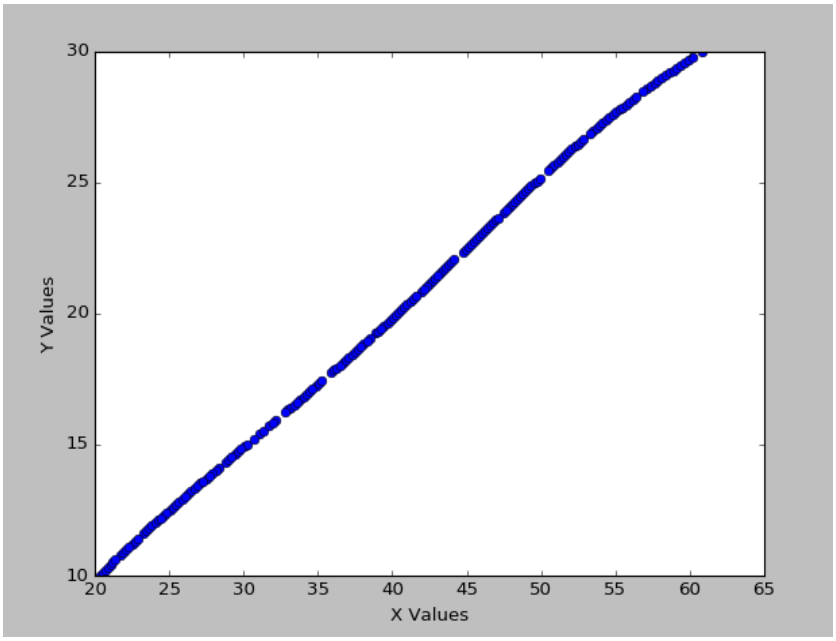
This graph is for **single variable dataset 1. (polynomial = 1)**



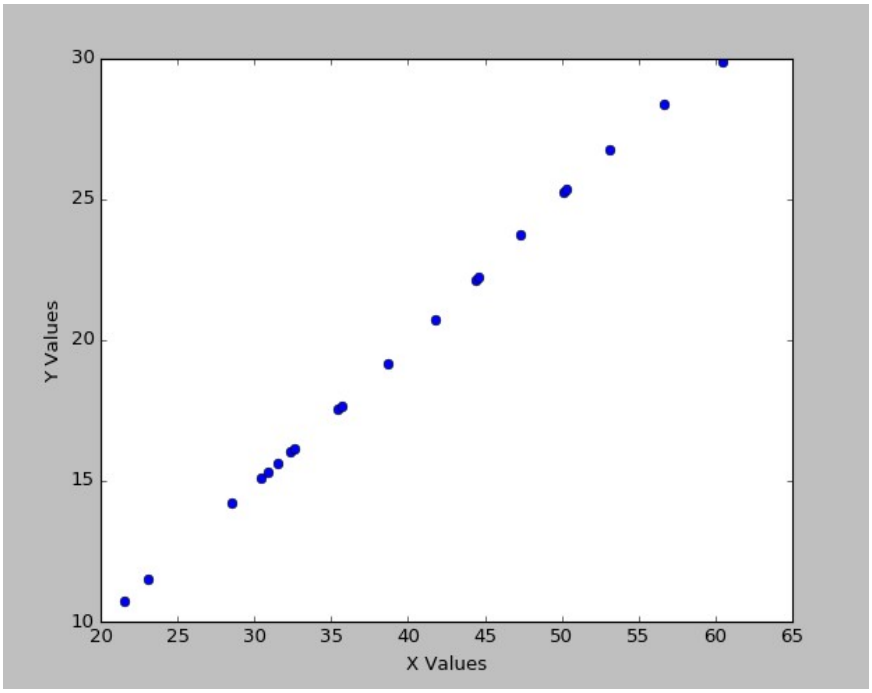
Single-Variate Linear & Polynomial Model Results:

These graphs are for single variable training and testing errors for **10 folds and 4 polynomial for first Dataset.**

This is Training error plot based on above mentioned parameters.



This is Testing Error Plot based on above mentioned parameters.



1. Polynomial =1, fold =10

Training Error: 4.227680
Testing Error: 4.325595

2. Polynomial =4, fold =10

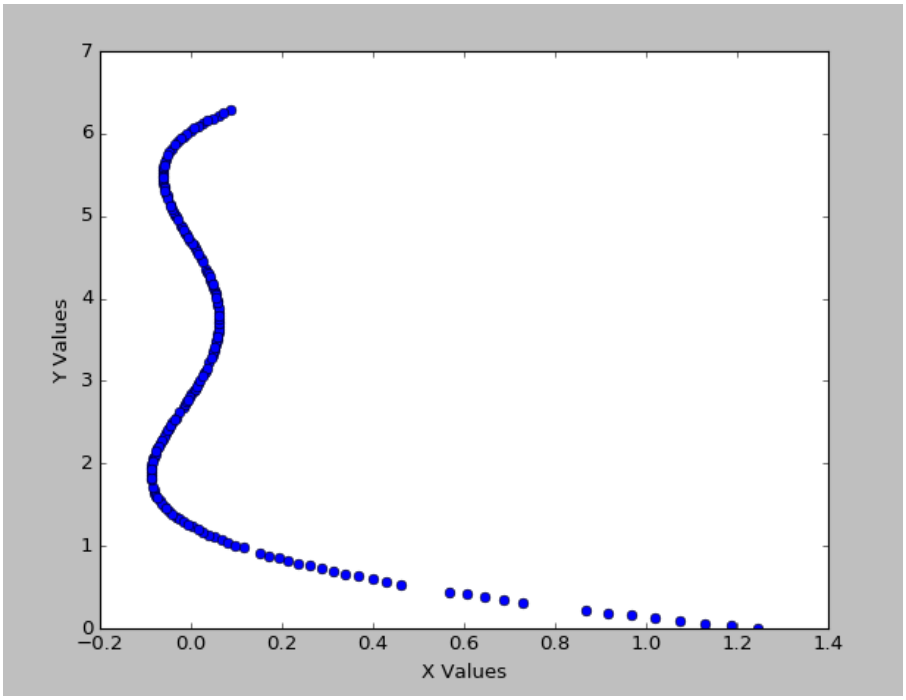
Training Error: 4.117312
Testing Error: 4.209009

3. Polynomial =1, fold =20

Training Error: 4.230768
Testing Error: 4.352882

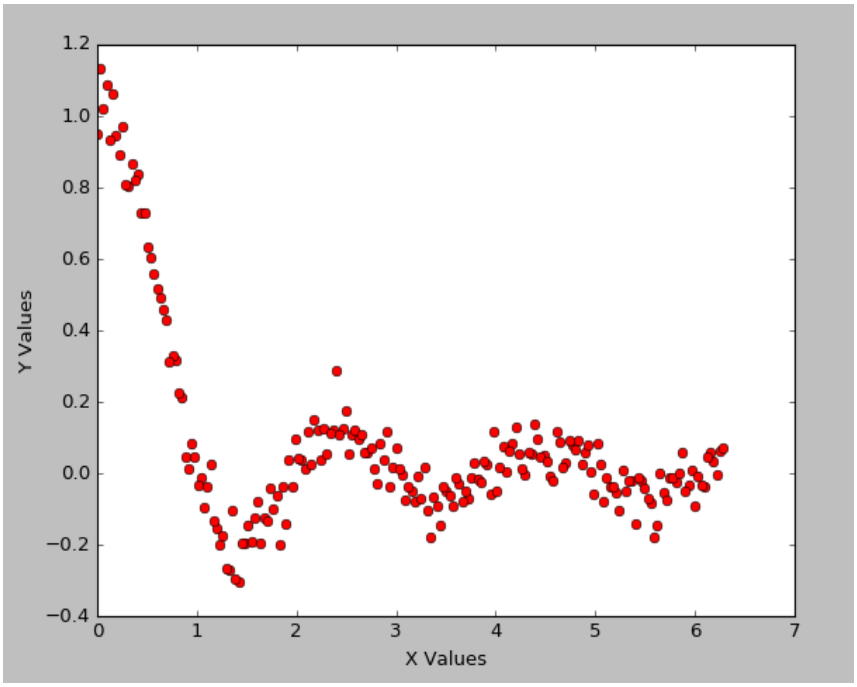
Here from this result, we can conclude that if we increase polynomial values then we get less errors because we get more nearer results if we increase degrees. Hence, we get values nearer to the actual ones. Moreover, if we increase folds then we get more errors because our training set will be smaller. Hence it leads to more errors.

Now, **for Dataset 2, Polynomial =4, fold =10**



Training Error: 0.011476
Testing Error: 0.011952

for Dataset 2, Polynomial =5 , fold =20



Training Error: 0.011064
Testing Error: 0.011973

Now, if reduce the folds from 90 to 50% then testing error would be

Type	**Testing Error**
File 1,Fold = 10 (90%)	4.325595
File 1, Fold = 5 (80%)	4.331284
File 1, Fold = 2 (50%)	4.353980

Parametric Regression A20338453

Multi-Variate Linear Regression Model Results:

Type	Training Error	Testing Error
File 1,Fold = 10	5.46336	5.484445
File 2, Fold = 20	0.011521	0.011578
File 4, Fold = 10	0.004189	0.004325

Now, for Explicit and Iterative solution, I have got very nearer Theta values.

1. For File 4, Fold 10, Theta Values for different Methods:

Explicit Solution

[[0.99017307]
[0.98348301]
[0.98343765]
[0.98341617]
[0.98344268]
[0.98342768]]

Iterative Solution

[[1.00960446e-02]
[5.01413180e-05]
[2.22473754e-04]
[4.74958981e-05]
[2.90411059e-04]
[-2.49146444e-05]]

2. For File 1, Fold 10, Theta Values for different Methods:

Explicit Solution

[[0.99997347]
[0.9938047]
[0.98894763]]

Iterative Solution

[[0.99999973]
[0.99991557]
[0.99984899]]

Kernel Method:

Time performance of kernel was too fast compare to previous solutions because here we are ignoring the unknown values. But accuracy was lower than the Iterative one.