CS 584 Machine Learning
Assignment-1
Parametric Regression

Name: Priyank P Shah　　　　　Date:01/19/2016
CWID: A20344797　　　　　Term:Spring'16

1. Problem Statement :

In this project I have developed parametric regression for single variable as well as for multiple features. This project is developed for both linear regression and polynomial regression. Project assignment is required to be done in two parts:

1) Single variable(feature) Regression:

- Compute Training and Testing error from the given data
- Test data with the cross-validation(K-Fold)
- Implement function with linear and polynomial function

2) Multi-variable(feature) Regression:

- Compute Training and Testing error from the given data
- Compute mean square error for for error found in Training and Testing errors.
- Test data with cross-validation(K-Fold)
- Implement function with linear function
- Solved the problem using gradient descent(iterative solution) and compared with explicit solution.

2. Proposed Solution:

- I have developed both the program from scratch by myself, as I am using python for the first time, I have referred notes given by professor.
- While developing code, I have used some basic matrix manipulation functions in order to generate result.
- For Iterative method in Multi-variable method, I have used gradient descent method.

Algorithm:

Single Variable Regression:

- Input: Fold_no(K-Fold), Polynomial Degree
- As per entered K-Fold number divide data into training and testing part and do cross-validation for each fold.
- I have applied dynamic approach in the calculation of matrix A and B due to the polynomial values for the model. Once Matrix is calculated, Theta's value will be calculated.

- Checking training error, testing error from the derived value of Theta.
- Plot graph from generated data(Predicted value of Y) of training and testing data.
- For those datasets which does not have linear data, we need to use polynomial model, function will return mean-square value of error, and hence we can opt best model which fit the data

Multivariate Regression:

- In Multivariate Regression, Data is separated into two part with the input value of K-Fold, those are training dataset and testing datasets.
- From the training data set Theta is calculated, from which training error is measured, and testing error also.
- Training and testing data is used to measured mean-square values of errors.
- To find Training error, I find Predicted value of Y using the Theta and given value of Z metrics and compare it with the given value in the datasets which turns into the training error at the end.
- In order to compare explicit method and iterative method I have used gradient descent for iterative method.

3. Implementation Details:

Instructions:

- Before running program file, put the datasets into the same folder where this two file resides.
- In order to load the different datasets, open particular file and change the file name at the very beginning of the file.
- I have implemented the assignment in two files named as “SingleVar.py” and “MultiVar.py” which contains code for single feature linear/polynomial regression and multivariate regression respectively.
- User will be prompted to input **K-Fold Value** and **polynomial degree value** to calculate further with **SingleVar.py** file.
- Graph will be automatically generated with the output lke training error, testing error.
- User will be prompted to input **K-Fold value** to calculate further with **MultiVar.py**

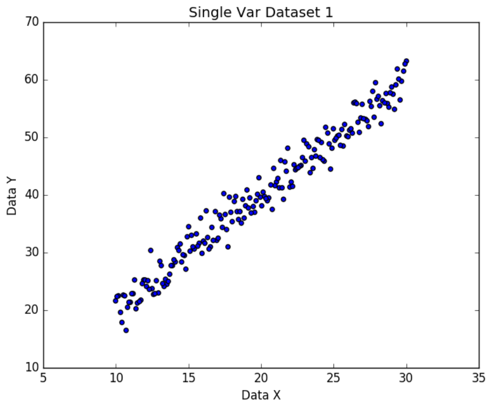
Design Issues:

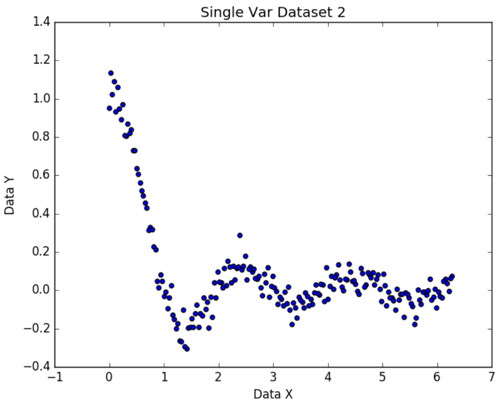
- As being new to the python, I had lot of confusion at early stage in array and metric.
- There were syntactical confusions for doing arithmetic operations, to overcome complexity of linear/polynomial implementation I used Iterative cycle to develop the code, where I first implemented very basic model to get familiar with python, and at each step I have added new functionality in modular fashion

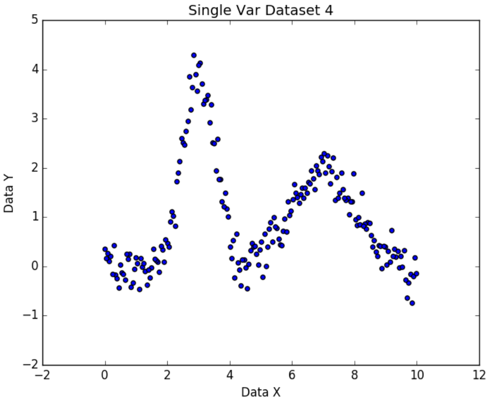
4. Results and Discussion:

Single Variable Regression:

A] Datasets Graphs:







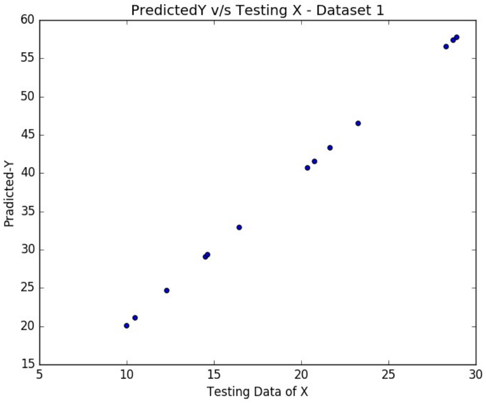
As we can see from the graph that, dataset 1 has linear data set, by going from dataset 2 towards 4, variance in data values is increasing. Hence in order to achieve perfect fit for the dataset 2 towards 4 , we need to use polynomial model instead of linear model to find regression.

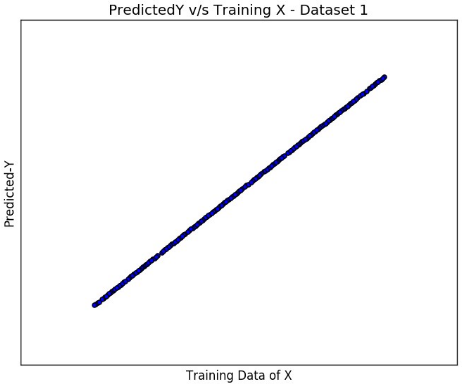
B] Experiments Results:

Data-Set	K-Fold	Polynomial Degree	Training Error	Testing Error
1	15	1	4.21490890057	4.53802046366
1	10	4	4.10154024012	4.53085947383
1	12	6	4.02997979816	4.30502257805
2	10	4	0.01207080250	0.00662935822
3	10	1	0.51830252003	0.32597773857
3	10	5	0.12943639196	0.10140347446
4	10	4	0.80784536967	1.24375014449
4	10	7	0.43599330529	0.60879587735

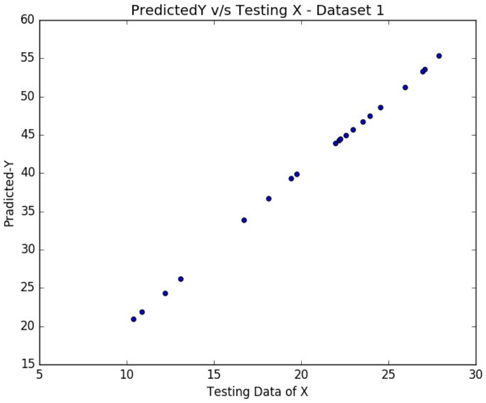
Graphs:

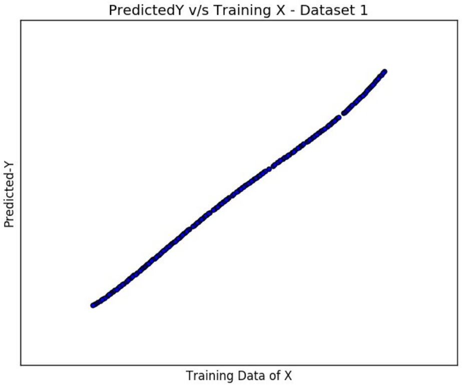
1) Dataset 1: 15 Fold – 1 Degree



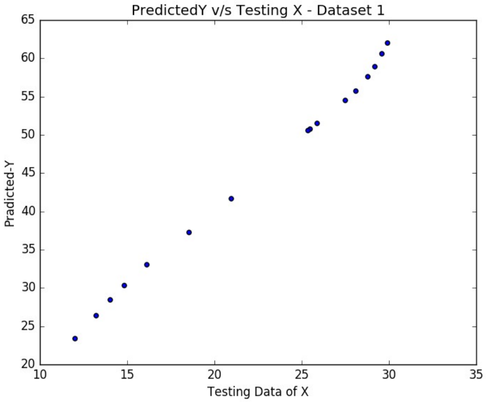


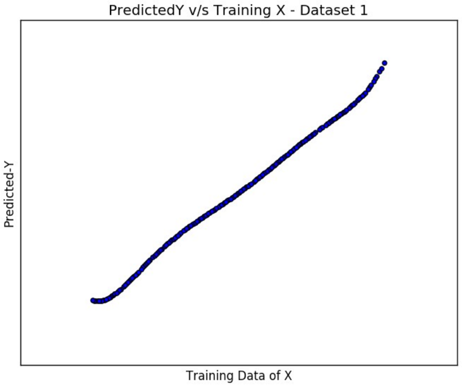
2) Dataset 1: 10 Fold – 4 Degree



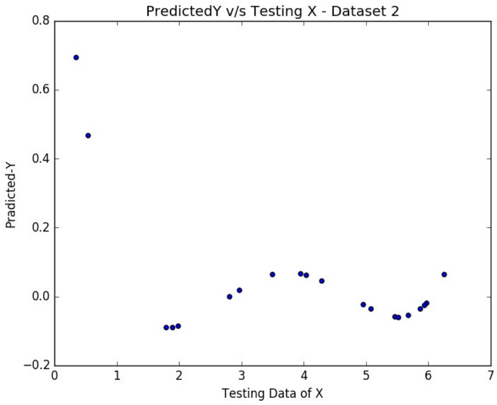


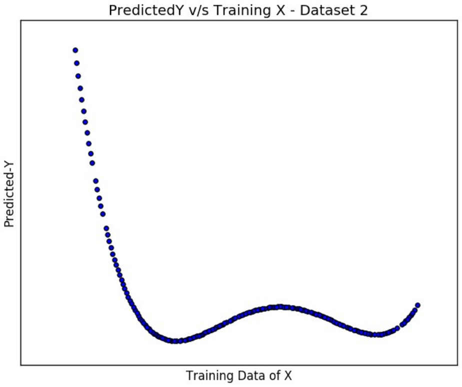
3) Dataset 1: 12 Fold – 6 Degree



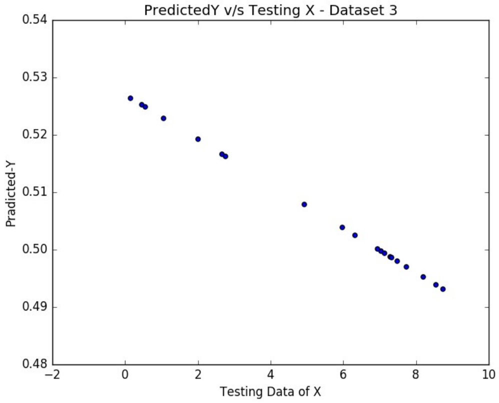


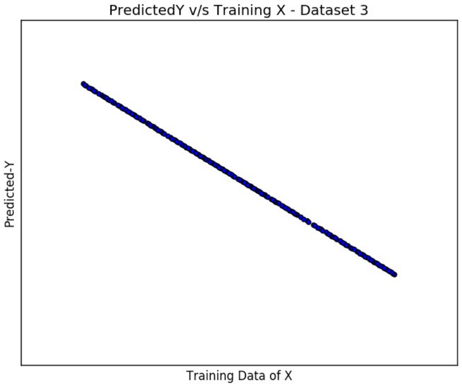
4) Data-Set 2: 10 Fold – 4 Degree



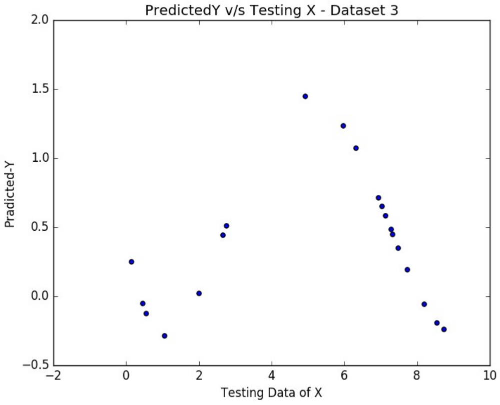


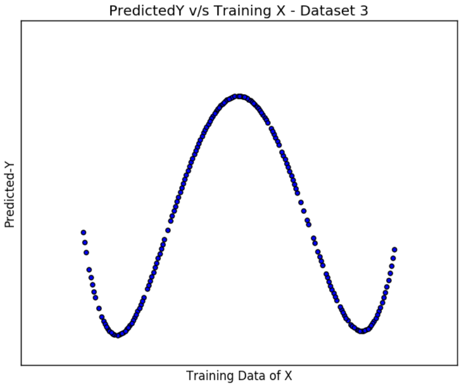
5) Data-Set 3: 10 Fold – 1 Degree



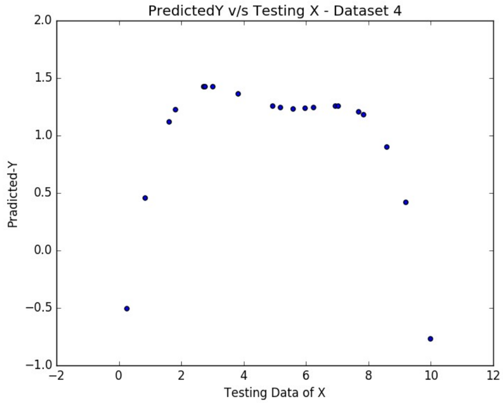


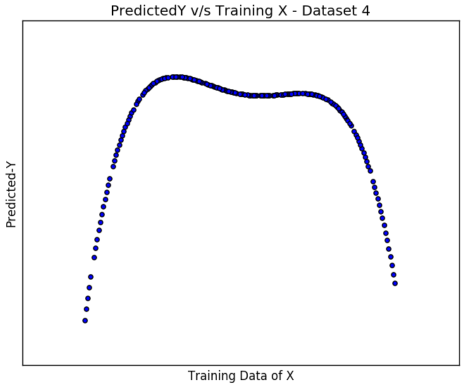
6) Data-Set 3: 10 Fold – 3 Degree



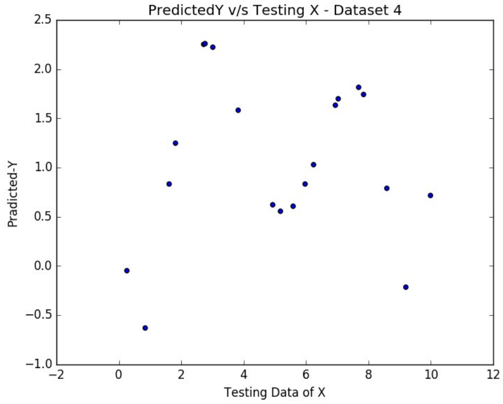


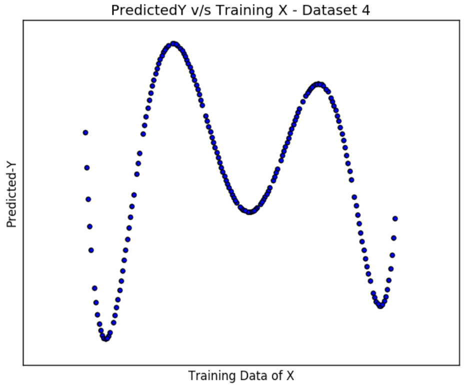
7) Data-Set 4: 10 Fold – 4 Degree





8) Data-Set 4: 10 Fold – 7Degree





C] Result Obtained with scikit-learn library from python:

FileName	Testing Error	Training Error
Svar-set1.dat	3.6542	4.2030
Svar-set2.dat	0.0519	0.0608
Svar-set3.dat	0.5058	0.4885
Svar-set4.dat	1.1295	1.2090

D] Polynomial Degree Observation:

By Testing Different polynomial, I observed that model executed with polynomial degree **around 7**, fits the data well, given in **dataset-2, dataset-3 and dataset-4,** while dataset-1 fits well with polynomial degree from **1 onwards.**

E] K-Fold Observation:

- By increasing the number of K-Fold value, we will get the smaller subset for training data and larger subset of testing data, which results into the higher error-rate.
- For cross-validation I have values from 10 – 18 to observe the error pattern, and found that for lower data set, I found the lower error rate.
- To improve this lag of error-rate with increased k-Fold, if we increase the polynomial degree then we can reduce the error. Reason is model with higher polynomial values fits the non-linear data set better.

Multi-variable Regression:

<u>Results:</u>

Data-Set	K-Fold	Training Error	Testing Error
1	10	0.258052722907	0.265106545138
1	12	0.26034889963	0.240758948055
1	15	0.258309659599	0.264351036684
2	10	0.0201450702752	0.0178157839351
3	10	0.250605451201	0.252013077217
3	12	0.250776594037	0.250380279524
4	10	0.00417610210993	0.00430714529063

C] I have solved the problem using gradient descent for iterative approach. Result that I achieved is almost similar to the explicit method. There was a minor difference. Result is as below.

Iterative Solution (Gradient Descent)	Explicit Solution
[0.99453738] [0.99768262] [0.98428219]	[1.00035924] [0.99871656] [0.99454442]

5. Reference:

- **http://www.astro.ufl.edu/~warner/prog/python.html**
- **http://www.stat.cmu.edu/~cshalizi/uADA/12/lectures/ch10.pdf**
- **http://www.datarobot.com/blog/regularized-linear-regression-with-scikit-learn/**