CS-584 MACHINE LEARNING

ASSIGNMENT – 1

PARAMETRIC REGRESSION

By

JAIMIN SANGHVI(A20344798)

PROBLEM STATEMENT

- Linear Regression is general method for estimating and describing association between a continuous outcome variable(dependent) and one or more multiple predictors in one equation.
- Given a data set consisting of a set of coordinates in the form positionX, representing feature, positionY being a representation of fitness of each point.
- In the given assignment1(Parametric Regression), I have implemented techniques for parametric regression.
- I have implemented and evaluated algorithm for single variable regression and multivariate regression. I have implemented 10 fold cross validation to evaluate the performance of single variable and multi variated regression.
- I have evaluated algorithms by changing parameters such as training and testing data set, type of regression from linear to polynomial(degree-1,2,3,4..).

PROPOSED SOLUTION

- I have implemented parametric regression algorithms from the scratch using core logic of matrix multiplication in Python.
- I have implemented two separate program files for single variable regression[SingleFeature.py] and multivariate regression[MultiVariance.py].
- I have implemented single variable and multivariate regression in the following steps:
  - ✔ Load data into an object and plot it to choose regression model(Linear/Polynomial)
  - ✔ Distinguish training and training data according to k-fold
  - ✔ Evaluated regression co-efficient using train data
  - ✔ Apply regression co-efficient to test data and train data to evaluate predicted value of y
  - ✔ Evaluate mean squared error
  - ✔ Observe the effect of performance on linear and polynomial models for single variable regression
  - ✔ Plot data for minimum MSE
- In addition, I have evaluated given data set using Scikit learn libraries and compare it with derived training and testing errors as well as regression co-efficients
- By using an iterative approach, I have evaluated regression

problem and compare the regression co-efficients

IMPLEMENTATION DETAILS

DESIGN ISSUES

- It is little bit challenging to implement matrix multiplication without using any library.
- In extend, it was difficult to implement dynamic matrix for polynomial regression
- Issued with Gaussian Kernel function for dual regression

SOLUTION

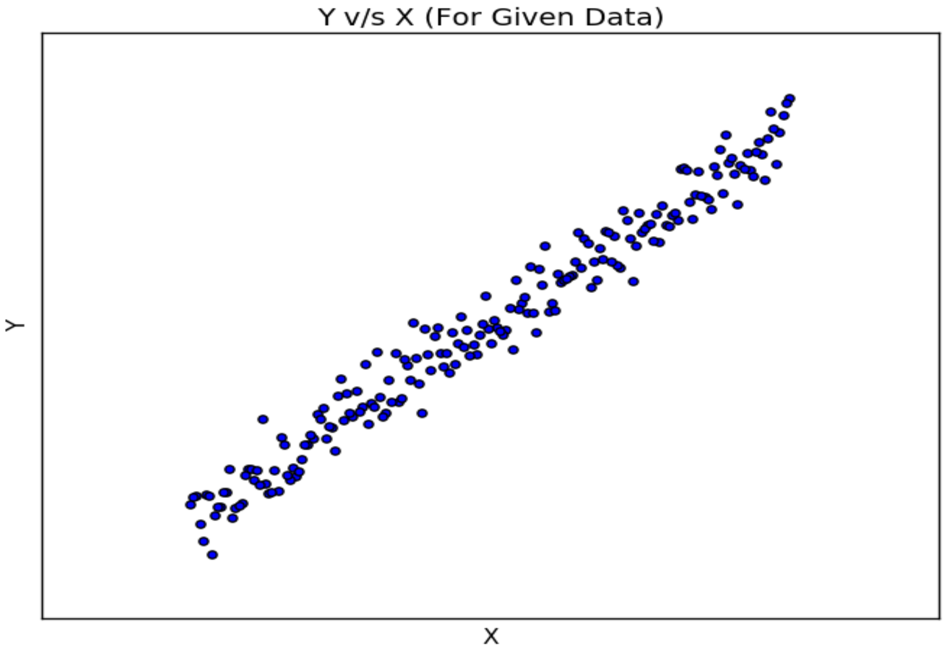
- To make it simple, I have used numpy library.
- I have generate outputs for different polynomials and compare it to find best model and plot it

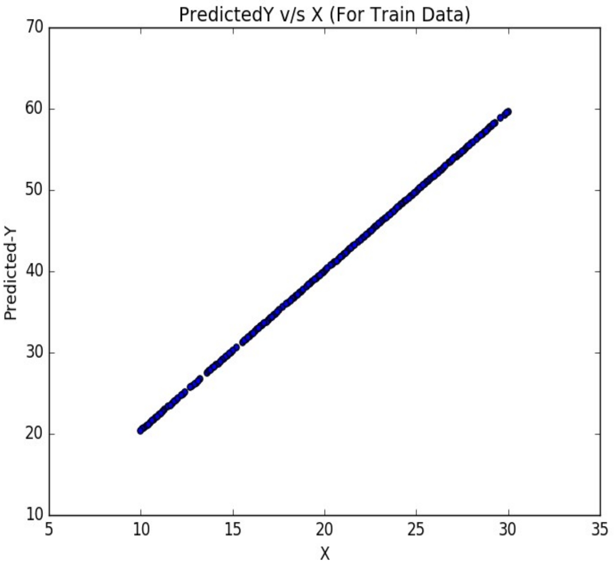
INSTRUCTION TO RUN

- I have implemented given problem solution in PyCharm IDE.
- Instruction to run given project file
  i. Load given project in IDE
  ii. Run SingleFeature.py file for single variable regression. It will run program for four given dataset simultaneously
  Note: For single variable regression, I have fetch dataset from URL. Hence it must need Internet connectivity
  i. Run MultiVariance.py file for Multi variance regression. It will run program for four given dataset simultaneously
  Note: For multivariate regression, I have fetch dataset from URL. Hence it must need Internet connectivity
  iii. Run SciKit_Learn.py file for training and testing error by Scikit library
  Note: For SciKit Library, I have used store dataset. You must need to store it with SciKit_Learn.py.

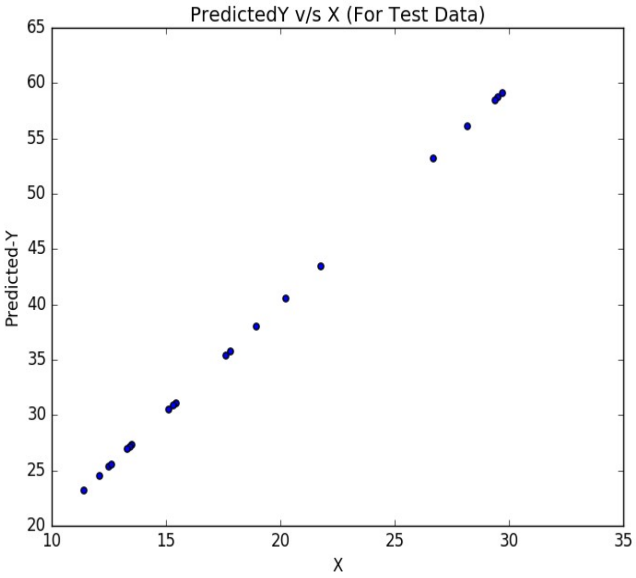
RESULT AND DISCUSSION

SINGLE VARIABLE REGRESSION (DATASET-1, K-fold=10)



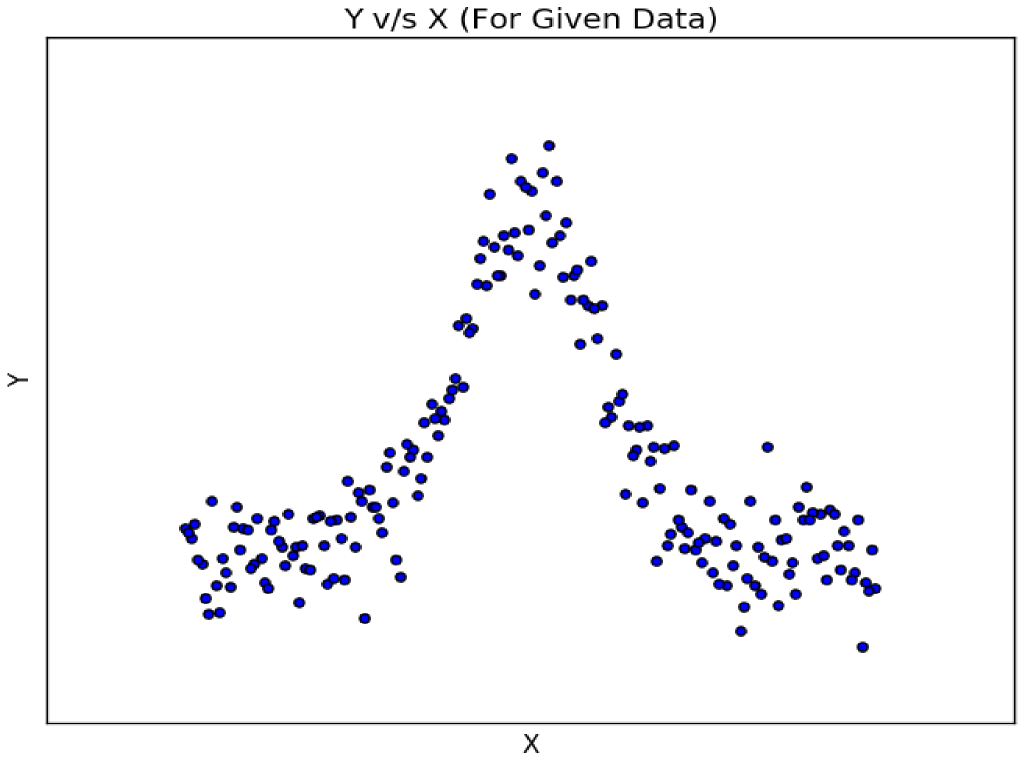


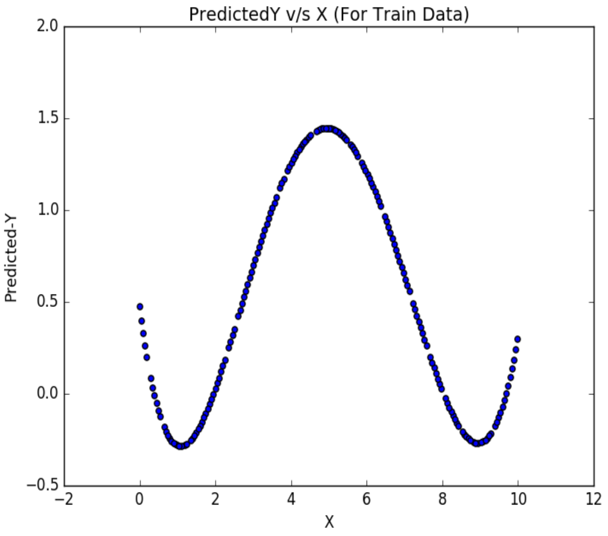
Training Data
(K-fold=10, Polynomial DEG=1)



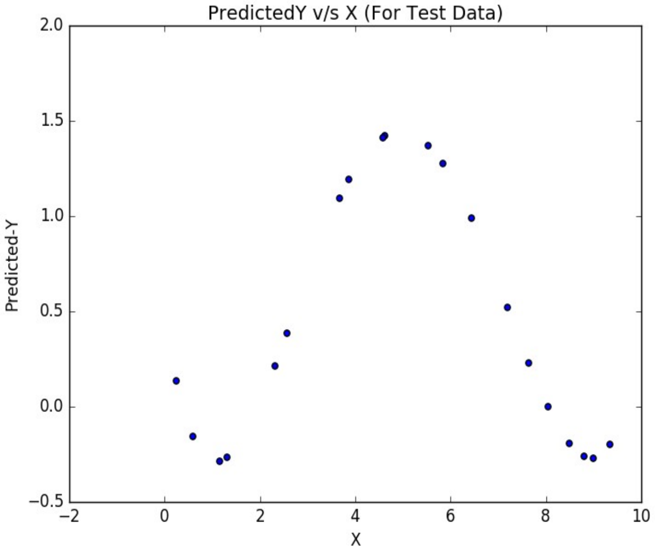
Testing Data
(K-fold=10, Polynomial DEG=1)

SINGLE VARIABLE REGRESSION (DATASET-3, DEG-5)





Training Data
(K-fold=10, Polynomial DEG=1)



Testing Data
(K-fold=10, Polynomial DEG=1)

- As per an observation, I can conclude that the testing error is always higher than training error.
- As per evaluation, I found that the result is improve as we increase degree of an polynomial.
- I have plotted the data for X and predictedY of the minimum mean squared error.
- After evaluating all set, I observed that first data is linear and other datasets are polynomial.

Compare Solution

- To compare the evaluated results, I have used in-build library **sklearn import linear_model.**
- Using this library, I have found linear training and testing error for single variable regression[Polynomial DEG=1, K-fold=1 and Dataset-1,2,3,4]

Dataset	Training Error(Evaluated)	Training Error(Sklearn)	Testing Error(Evaluated)	Testing Error (Sklearn)
svar-set1	4.3351	4.335	3.4944	3.495
svar-set2	0.0605	0.060	0.0518	0.051
svar-set3	0.4978	0.497	0.5063	0.506
svar-set4	1.2092	1.209	1.1373	1.137

I observed that as I reduced k-fold size, the testing data will decrease.

MULTIPLE VARIABLE REGRESSION

In multivariate regression, I have implemented algorithm for higher dimensional data using 10 K-folds. I have evaluated training and testing error for all given data set. In case of multivariate, there is small difference between training and testing error.

In addition, I have observed and compared the values of training and testing with the output(error) of sklearn library.

Dataset	Training Error(Evaluated)	Training Error(Sklearn)	Testing Error(Evaluated)	Testing Error (Sklearn)
mvar-set1	0.2582	0.2582	0.2630	0.2639
mvar-set2	0.0199	0.0200	0.0194	0.0195
mvar-set3	0.2512	0.2512	0.2463	0.2464
mvar-set4	0.0042	0.0042	0.0038	0.0039

ITERATIVE APPROACH:

I have implemented gradient algorithm and find gradient theta for a given set. In addition I have compared the gradient theta with evaluated results.

The comparison between gradient and evaluated theta is as below:

Theta	Evaluated Theta	Gradient Theta
Theta-1	0.9958	0.9999
Theta-2	0.9975	0.9999
Theta-3	0.9904	0.9998

REFERENCES

1.http://www.astro.ufl.edu/~warner/prog/python.html
2.http://www.holehouse.org/mlclass/10_Advice_for_applying_machine_learning.html
3.http://aimotion.blogspot.com/2011/10/machine-learning-with-python-linear.html
4.http://scikit-learn.org/stable/supervised_learning.html#supervised-learning