ELL784 Assignment 1

Prashant Atkale

2021AMA2631

Question 1: Part A

In this part of assignment, we had to implement the concepts of Linear Regression to solve the problem of polynomial curve fitting. I have done part in two-part sections, one with cross validation and other with by using regularization and then compared the results

Here, the noise in the data set is Gaussian. Hence, we must minimise the least-squares error. This part has been solved using Moore-Penrose Pseudoinverse analytical solution and compared the coefficients of polynomial with the model polynomial coefficients.

The below figure shows the scatter plot for feature and label

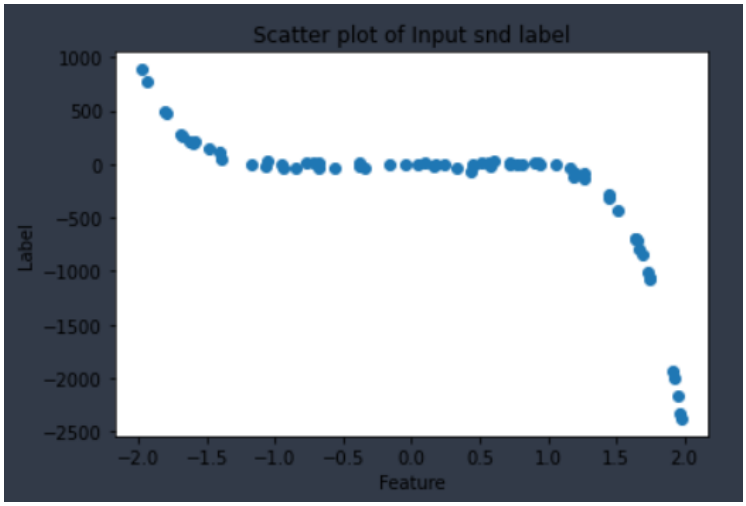


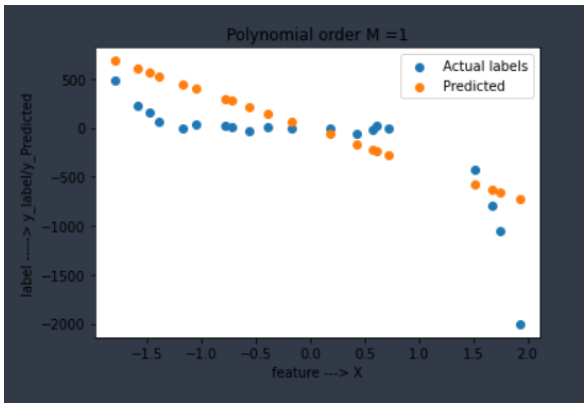
Figure 1: Data distribution for data set with Gaussian noise

For both the parts, we assume the data to be from a polynomial of degree M and then generate a design matrix containing feature vectors (1, x^1, x^2, ..., $x^{M)}$ for each x in the data set.

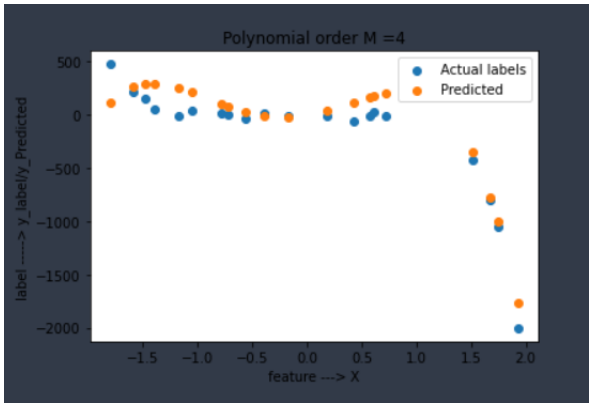
We then minimise the least-squares error using the above-mentioned method. Using the implementation of the above methods done using NumPy, I have obtained the following results and observations.

Cross Validation: 20 Data Points

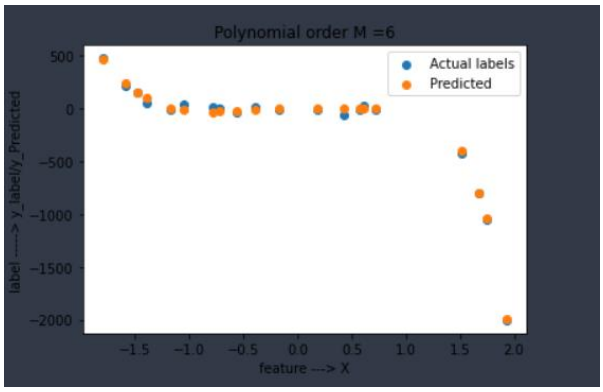
Polynomial features are created using Polynomial Features tool of Scikit-Learn. Cross validation with cv=10 is performed using linear regression model. Firstly, to get an idea about the degree of the polynomial in the data set, we plot the polynomial curves of various degrees obtained by fitting it on the data set.
Following are the different polynomials are obtained,



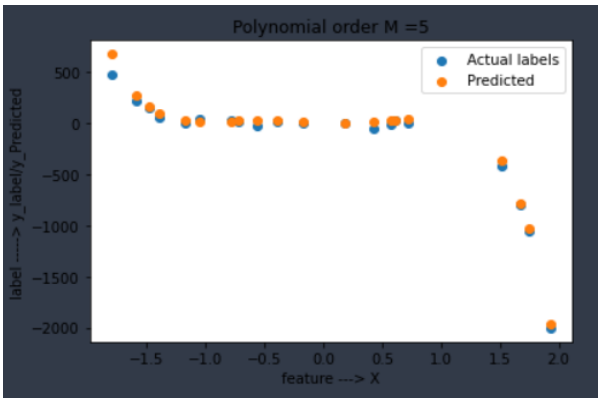
Polynomial order M=1



Polynomial order M=4



Polynomial order M=6



Polynomial order M=5

The above graphs show how the polynomial curve changes on increasing the degree of the polynomial. We see that curves of degree 1 and 4 barely manage to predict the data with accuracy, while the curves of higher degrees fit the data in a much better way. This is because increasing the polynomial degree, increases the complexity of the model and now it can model a much more complex curves due to the increased number of parameters.

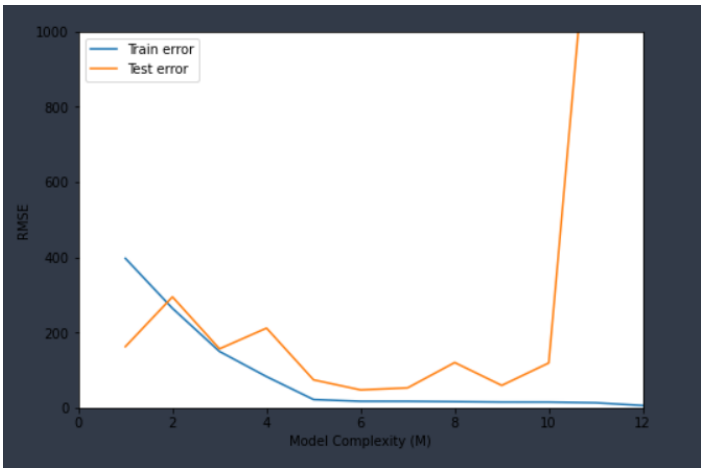


Figure: Train/Test Error VS model complexity

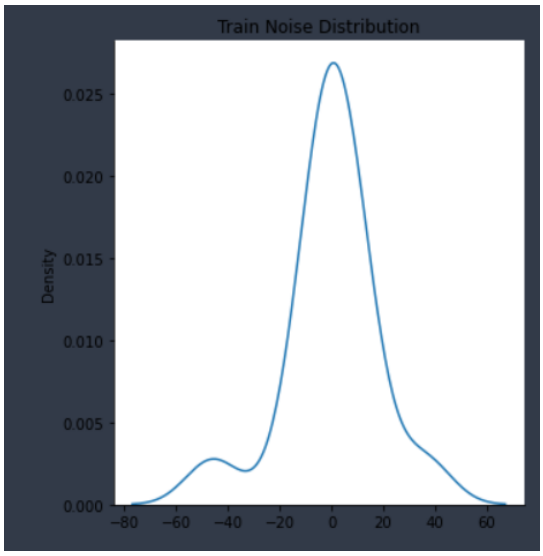
The above plot shows how the losses reduce on increasing the model complexity, i.e., increasing the degree of the polynomial. We can clearly see that the loss decreases by a considerable amount at M=6 and remains the same for the next degree. Error is exploding above polynomial degree M=7. So we can go for polynomial of order 6.

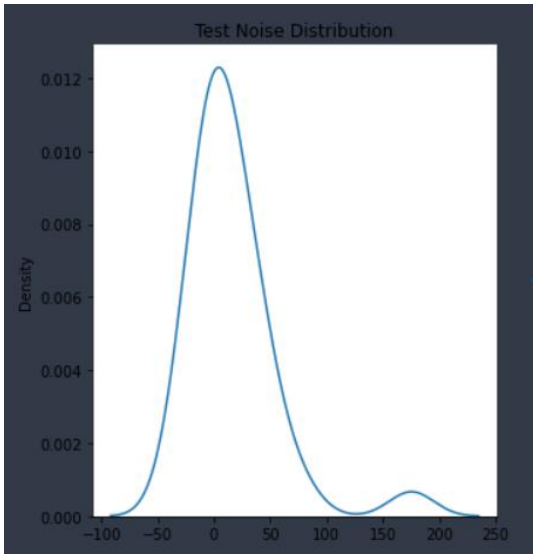
For Polynomial of order M=6,

Train -----→ root mean squared error = 16.95927669992311

Test ------→ root mean squared error = 47.1883672236705

Noise Variation





Noise variation is gaussian as mentioned in the question with mean 0.

For Train set ---> Mean = 6.597896832058073e-15 ; variance = 287.61706618455486

Test set ---> Mean = 6.597896832058073e-15; variance = 287.61706618455486

Analytical Solution

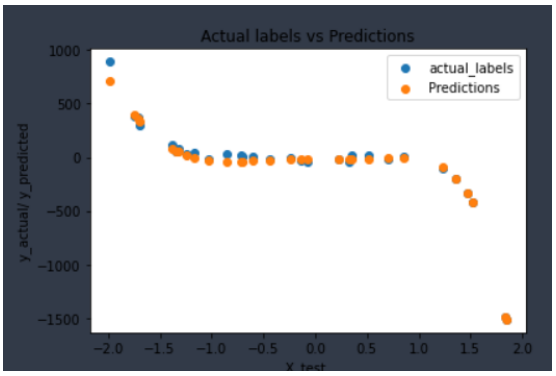
- Using Moore- Penrose analytical solution, Weights values are given below

  Weights = [-14.72975461, -49.71679681, 37.36305104, 133.61045061,

  -8.22718971, -84.08926617, -12.62226398]

- Using Coefficients of our model we get following Weights values,
  Weights = [0. , 0.20862906, -44.48827644, 80.19760607,
  56.46875995, -71.41464834, -25.20788791]

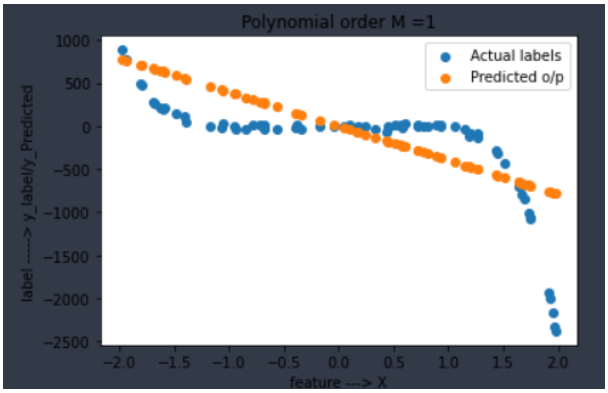
  Coefficients in both the cases is quite different because of limited data points. Also, our model has high test error value.



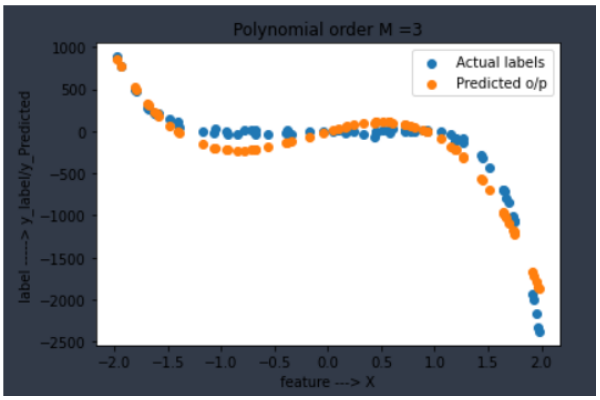
- This plot shows the predictions when tested on test data set provided.
- We have got good predictions for unseen data points

Cross Validation: 70 Data Points

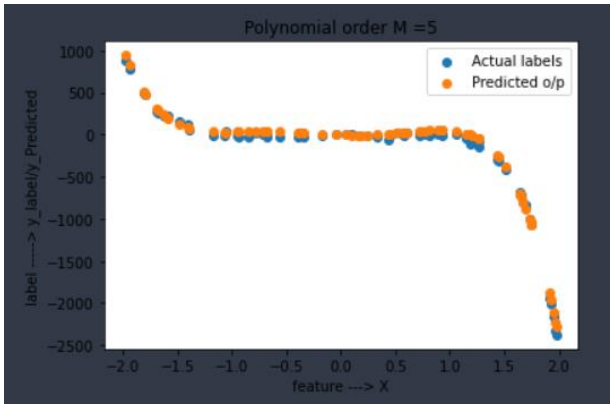
Above analysis again performed for 70 data points. Different Polynomial order are shown below



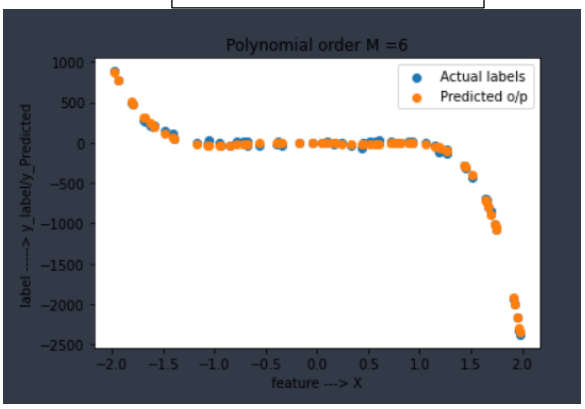
Polynomial Order M= 1



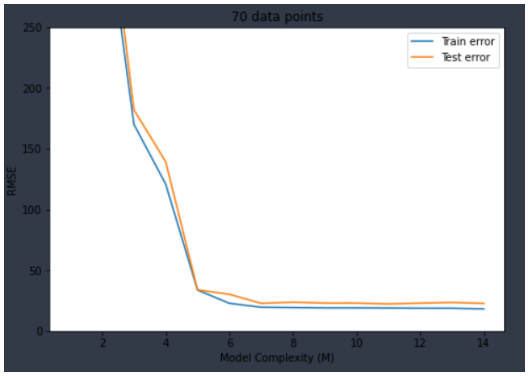
Polynomial Order M=3



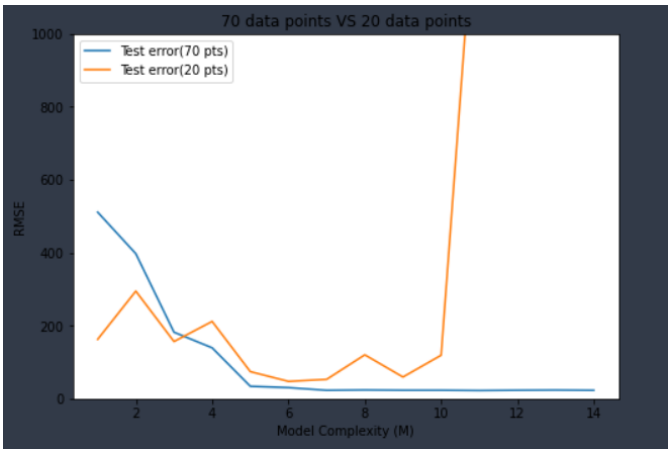
Polynomial Order M= 5



Polynomial Order M= 6



- As you can see, here Polynomial of Order 6 is best fitting to data with less error on test and train.
- From the graph we can conclude that M=6 has less train and test error while subsequent degrees error remains almost same so we go for M=6



- This graph shows Test error for 20 data points and 70 data points
- For 70 data points test error is very low compared to 20 points.
- Thus for 70 points our model generalizes well for unseen points

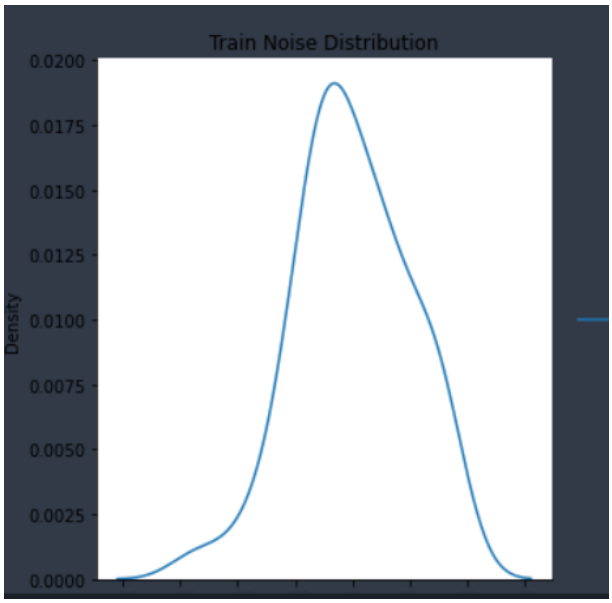
For Chosen Polynomial of Order M= 6 :

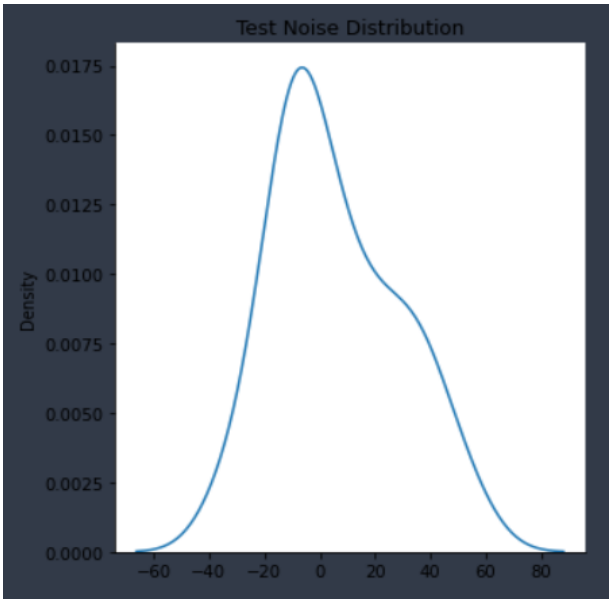
Train root mean squared error is 19.47289559022754

Test root mean squared error is 22.672273108391504

Erro for 70 data points is small compared to 20 data points.

Noise Variation





Following are mean and variance of Noise

Test set ---> Mean = 6.597896832058073e-15 ; variance = 287.61706618455486

Train set ---> Mean = -2.2041325680721474e-14 ; variance = 379.1936626679032

ANALYTICAL SOLUTION

- Using Moore- Penrose analytical solution, Weights values are given below

  Weights = [-9.45053354, 8.10314309, 6.91754879, 12.80357873,

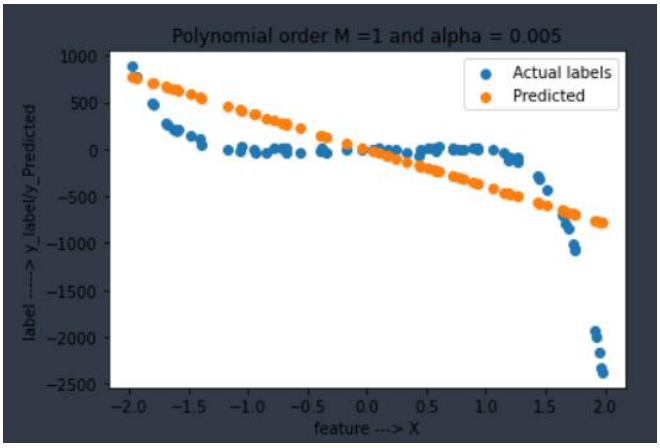
  8.76927798, -16.81572811, -15.12776902, -10.61724254]

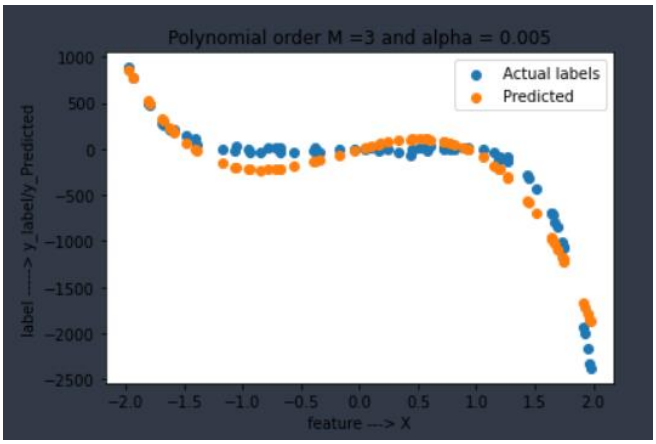
- Using Coefficients of our model we get following Weights values,
  Weights = [0. , 7.8054969 , -12.06558326, 22.96921415,
  19.52383156, -23.57539506, -16.62475056, -9.51612292]

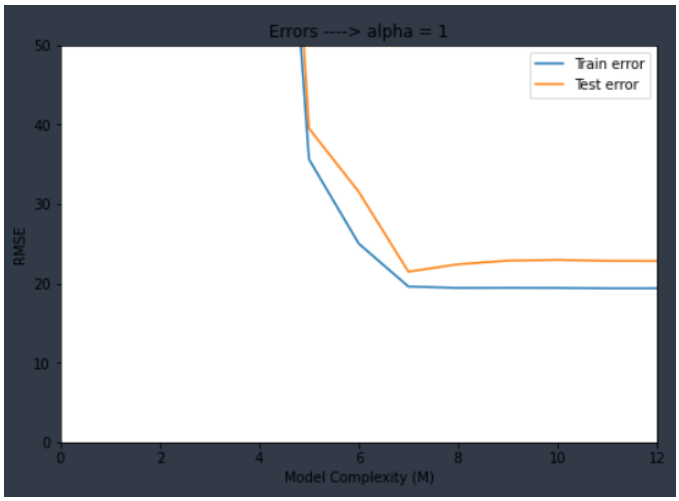
  Some coefficients have nearly same value because of the increased data points

Regularization:

Ridge regularization is used. I have taken set of hyperparameter that is alpha values and found best alpha value and polynomial order. By plotting error for different alpha values.







- Found best parameters alpha =1 with degree =7
- As it was showing less train/test error

For Polynomial of order M= 7.

Train root mean squared error is 19.60441779538938

Test root mean squared error is 21.474141003281336

Error is almost same as cross validation as we have seen above.

Analytical Solution

- Using Moore- Penrose analytical solution, Weights values are given below

  Weights = [-9.45053354, 8.10314309, 6.91754879, 12.80357873,

  8.76927798, -16.81572811, -15.12776902, -10.61724254]

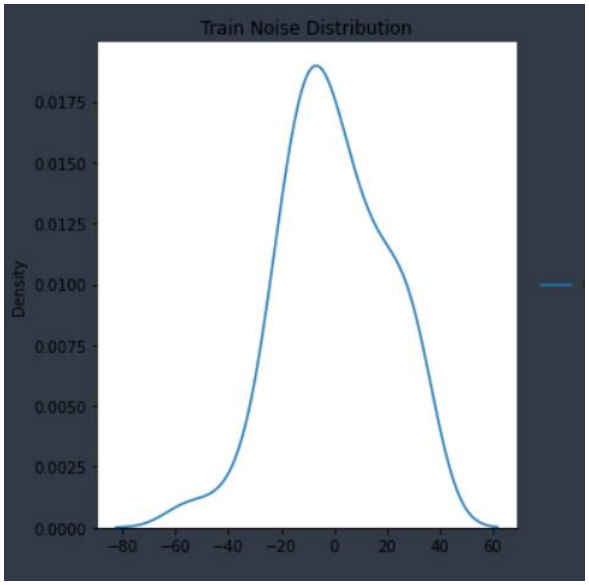
- Using Coefficients of our model we get following Weights values

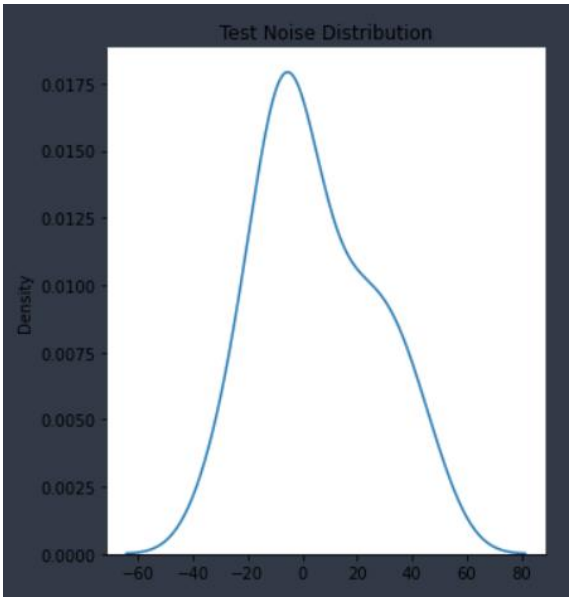
  Weights = [0. , 13.32252291, 0.43358118, 5.91503218,

  11.58704784, -13.98079644, -15.34040744, -10.96807427]

Weight values are near about compared to both cases

Noise Variation





Gaussian Noise having mean and Variance values as follows

Test set ---> Mean = -4.0602442043434294e-15; variance = 384.3331970961798

Train set ---> Mean = -4.0602442043434294e-15; variance = 384.3331970961798

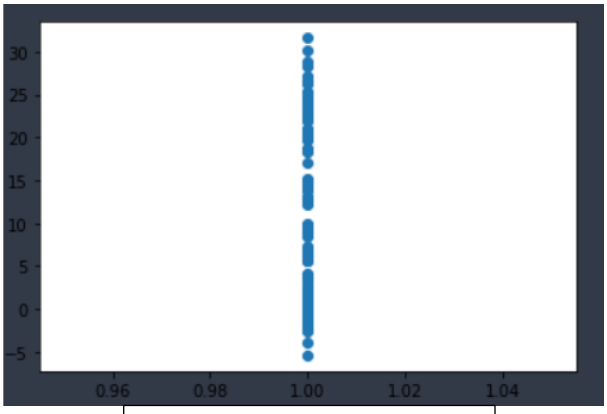
Question 2

For this part, we have to use linear regression models to model the given time-series data. On X axis we have date given and y axis has labels given

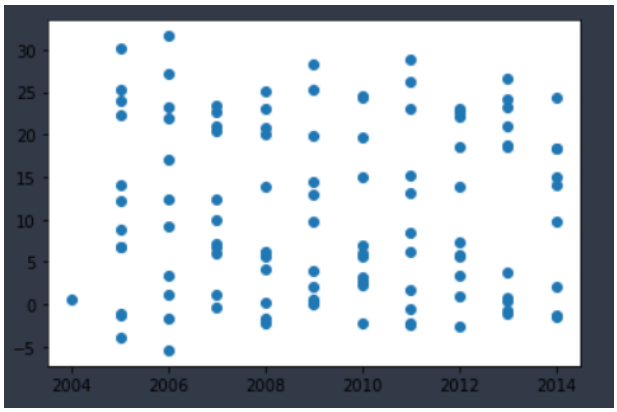
I have converted date column into three columns such as month, day, year and then checked the variation with the labels. I case of day and year there was constant correlation with respect to labels. The variation of month data with labels is kind of inverted parabola. So chosen feature as month

From the above data, we can see that the has somewhat similar patterns over months of every year. The data patterns thus, repeat over the year. This motivates us to plot the data for the various months and years.

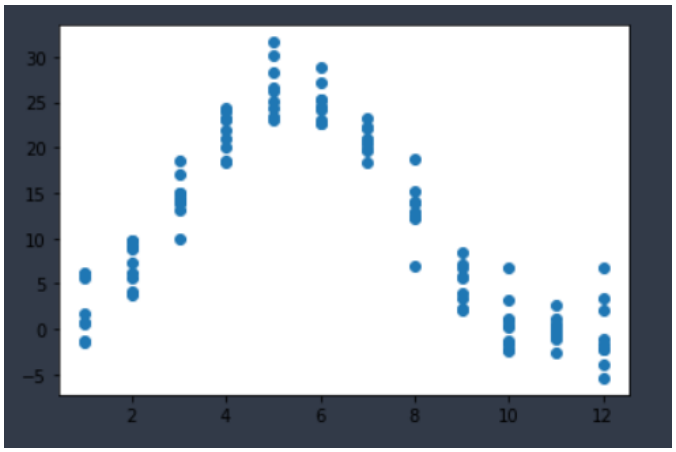
Following is the scatterplot for month, day, year with respect to labels



Scatter plot of Day vs labels

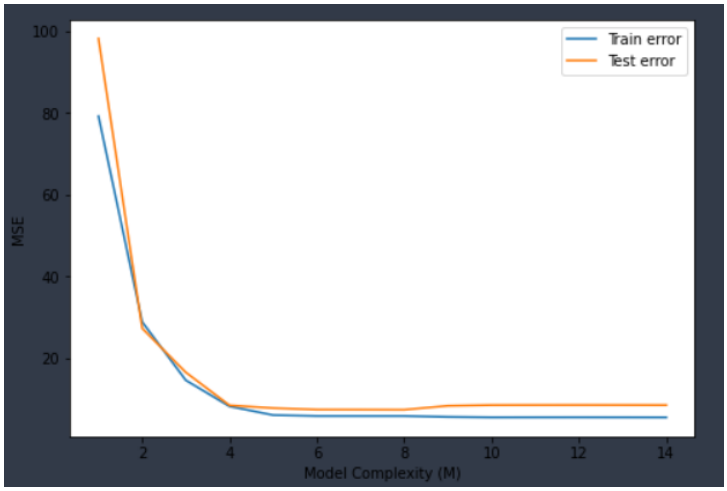


Scatter plot of year vs labels



Scatter plot of Month vs labels

From the data plots, we can see a clear trend in the values based on their months. We can try to fit a curve and model a polynomial which is a function of the month by fitting it on the given data. One trivial way is that we just return the average values of the month for which we have to predict the value but that would not work if we were required to give prediction on a continuous set. So, we try and fit a polynomial on this month wise data.



- This plot shows the Train/test error for different order polynomials
- From degree 6 onwards there is little change in error values
- From degree 7 onwards test error increased slightly
- our optimal polynomial order is M= 6 for this case
- Here cross validation is used to generalize more for unseen data

For polynomial of order 6,

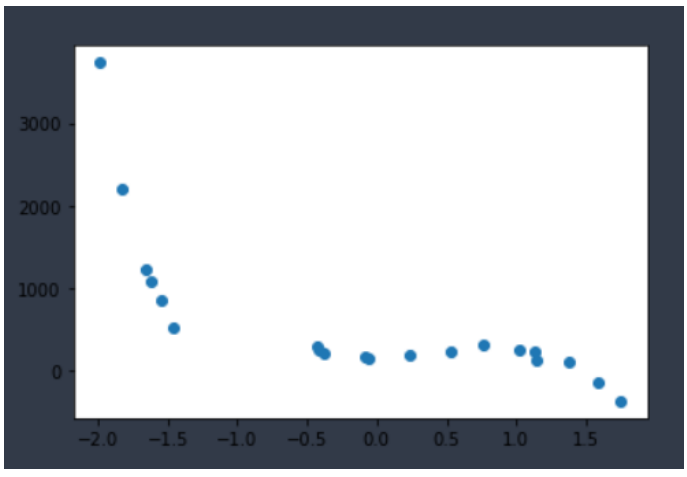
Train ----→ root mean squared error= 2.4359409390478315

Test -----→ root mean squared error = 2.7621613738942554

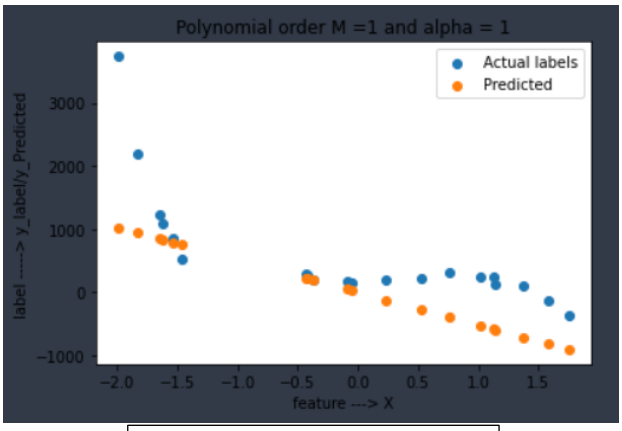
This question is again repeated for regularization along with grid search and hyperparameter tuning is also done, but the results aren't improved much.

Question 1B (Bonus Question)

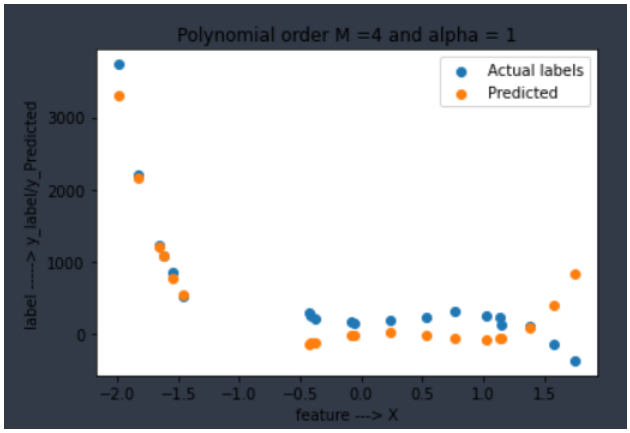
In this part we have to see the nature of noise. Following is the variation of features and labels.



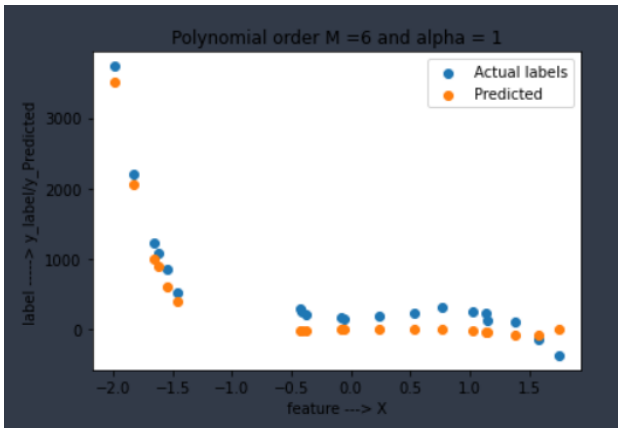
Following plots shows the polynomial of different order with different hyperparameters. So here we can see how there is variation over different polynomial orders. Similar plots have shown for various polynomials ranging from 1 to 15 in code,



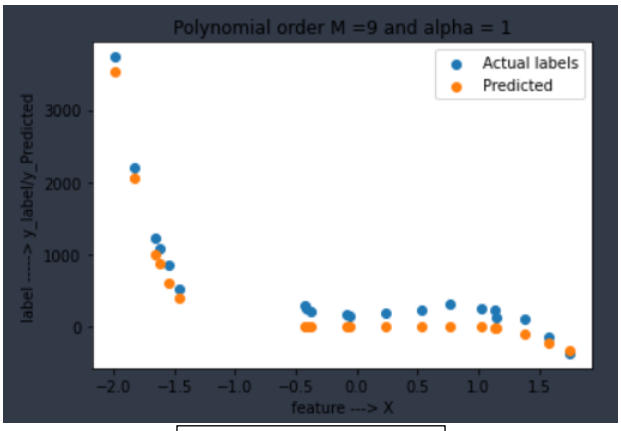
M = 1 and alpha =1



M = 4 and alpha =1

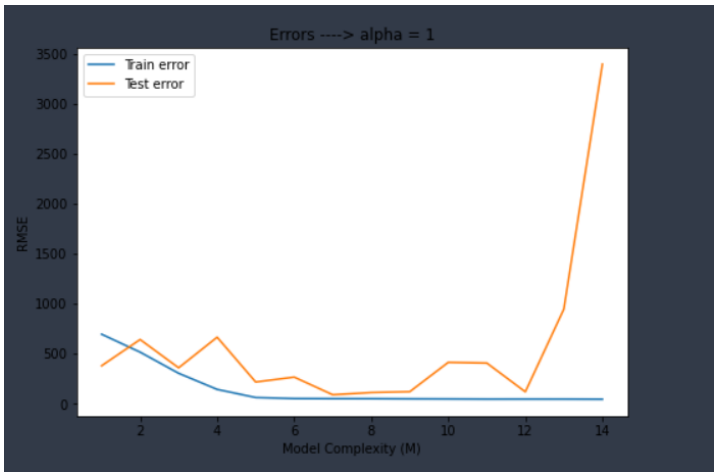


M = 6 and alpha =1



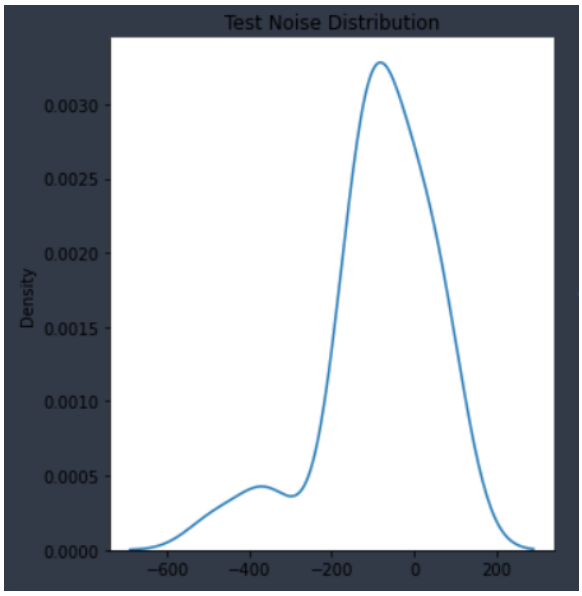
M = 9 and alpha =1

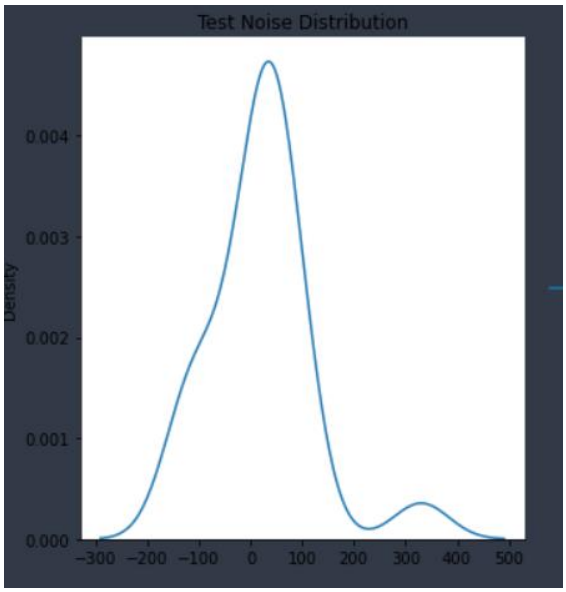
Here polynomial order 9 is selected and hyperparameter alpha is taken as 1. As the error for m=9 is less compared to others as shown in below plot.



In above plot, we can see test error is exploding beyond polynomial order M=12. The lowest test error is at M=9 and after that error is increasing. So optimal order of polynomial is taken as 9 in this case.

Noise Distribution





- The Noise distribution in this case is Bi- modal gaussian in above case, but for the train errors its gaussian in nature