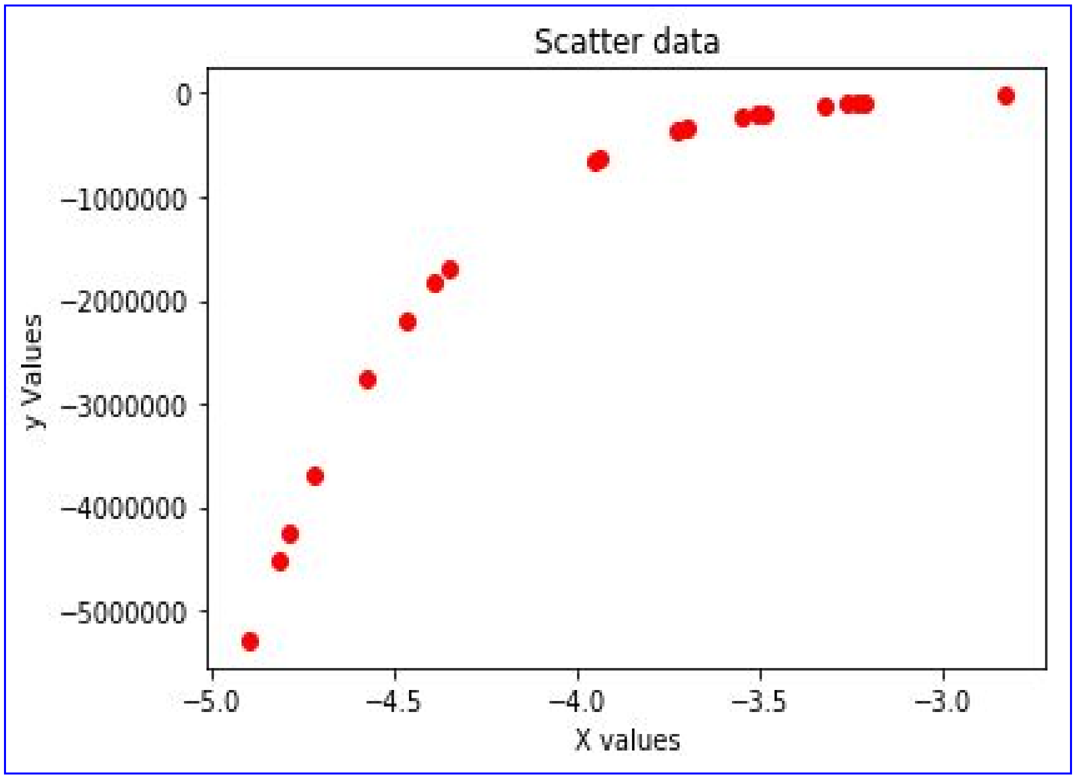
Machine Learning Assignment - 3 :

Here we are applying linear regression on a data set , but we are considering only 20 data points out of 100 points present.

Scatter Plot of the data Distribution :



The data is widely distributed as mentioned below:

Y ranges from : -5286527.14 to -21730.98
X ranges from : -4.9 to -2.83

We use the below libraries :

1. *sklearn.model_selection.KFold* : To do train-validation split of the data
2. *sklearn.linear_model.LinearRegression* : To create the Linear Regression
3. *sklearn.preprocessing.PolynomialFeatures*: To create the polynomial features

We are using the below error function for estimating the models behaviour and the plots to identify the region of Overfit, Underfit and Goodfit.

1. *mean_squared_error*
2. *mean_absolute_error*
3. *median_absolute_error*

<u>For Goodness of Fit we use the r2-Value :</u>

The goodness of fit of a statistical model describes how well it fits a set of observations. Measures of goodness of fit typically summarize the discrepancy between observed values and the values expected under the model in question.

total ss = regression ss +residual ss
r-square = regress sum of square/ total sum of square

r2 is always 0-1
0 : poor fitting
1 : good fitting

if R2 value is : 0.4745
The regression model can explain about 47.45 % variation in the y values.

<u>Now on the data what we do is :</u>

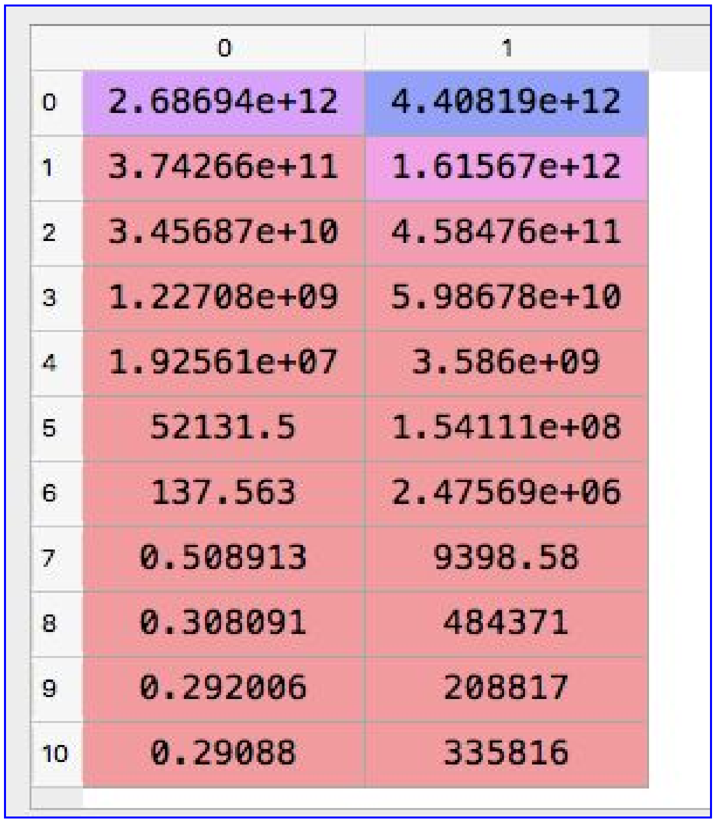
1. We loop over the different degree polynomial and
2. For each of them we calculate each of the error metrics
   2.1. During error metric calculation we use the KFold library and take an average Over the errors calculated for different folds.
3. We put it in a matrix form.
4. We calculate the r2 value as Goodness of fit measure and according we select the degree(variable name : deg) for which we have the best goodness of fit (r2-value) measure.

<u>The error metric values calculated are as below :</u>

Column 0 : Train Error and Column 1 : Test Error

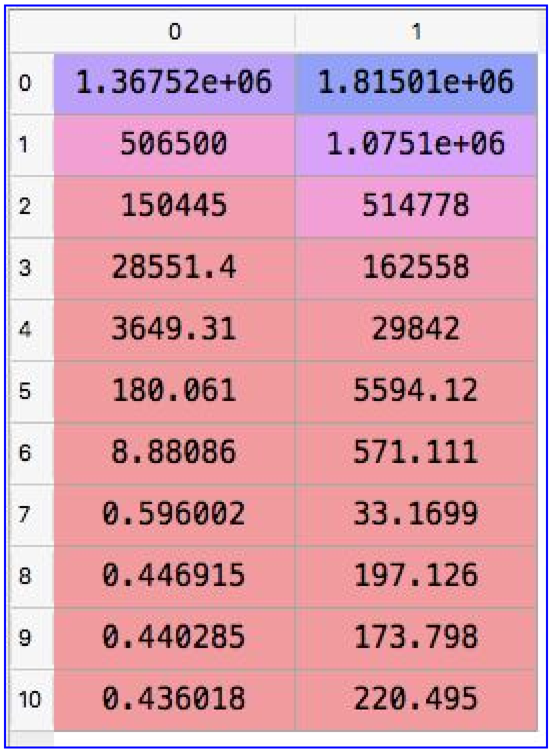
Row No’s : degree of the polynomial .

Mean Squared Error :



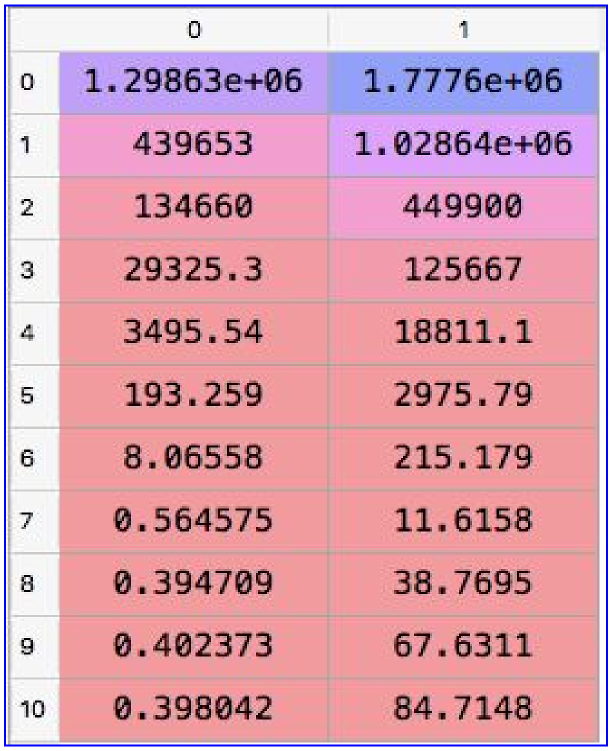
	0	1
0	2.68694e+12	4.40819e+12
1	3.74266e+11	1.61567e+12
2	3.45687e+10	4.58476e+11
3	1.22708e+09	5.98678e+10
4	1.92561e+07	3.586e+09
5	52131.5	1.54111e+08
6	137.563	2.47569e+06
7	0.508913	9398.58
8	0.308091	484371
9	0.292006	208817
10	0.29088	335816

Mean Absolute Error :



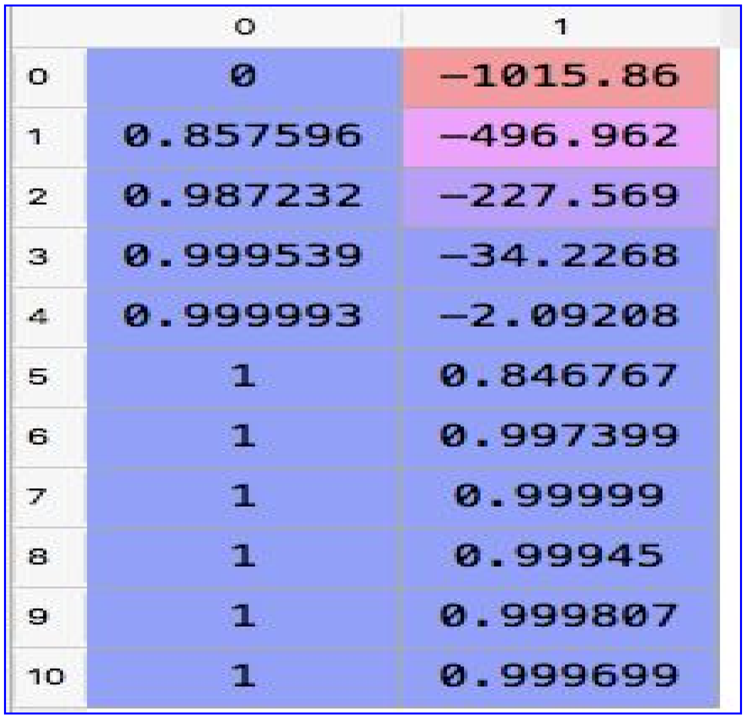
	0	1
0	1.36752e+06	1.81501e+06
1	506500	1.0751e+06
2	150445	514778
3	28551.4	162558
4	3649.31	29842
5	180.061	5594.12
6	8.88086	571.111
7	0.596002	33.1699
8	0.446915	197.126
9	0.440285	173.798
10	0.436018	220.495

Median Absolute Error :



	0	1
0	1.29863e+06	1.7776e+06
1	439653	1.02864e+06
2	134660	449900
3	29325.3	125667
4	3495.54	18811.1
5	193.259	2975.79
6	8.06558	215.179
7	0.564575	11.6158
8	0.394709	38.7695
9	0.402373	67.6311
10	0.398042	84.7148

The goodness of fit measure Values are r2_values:



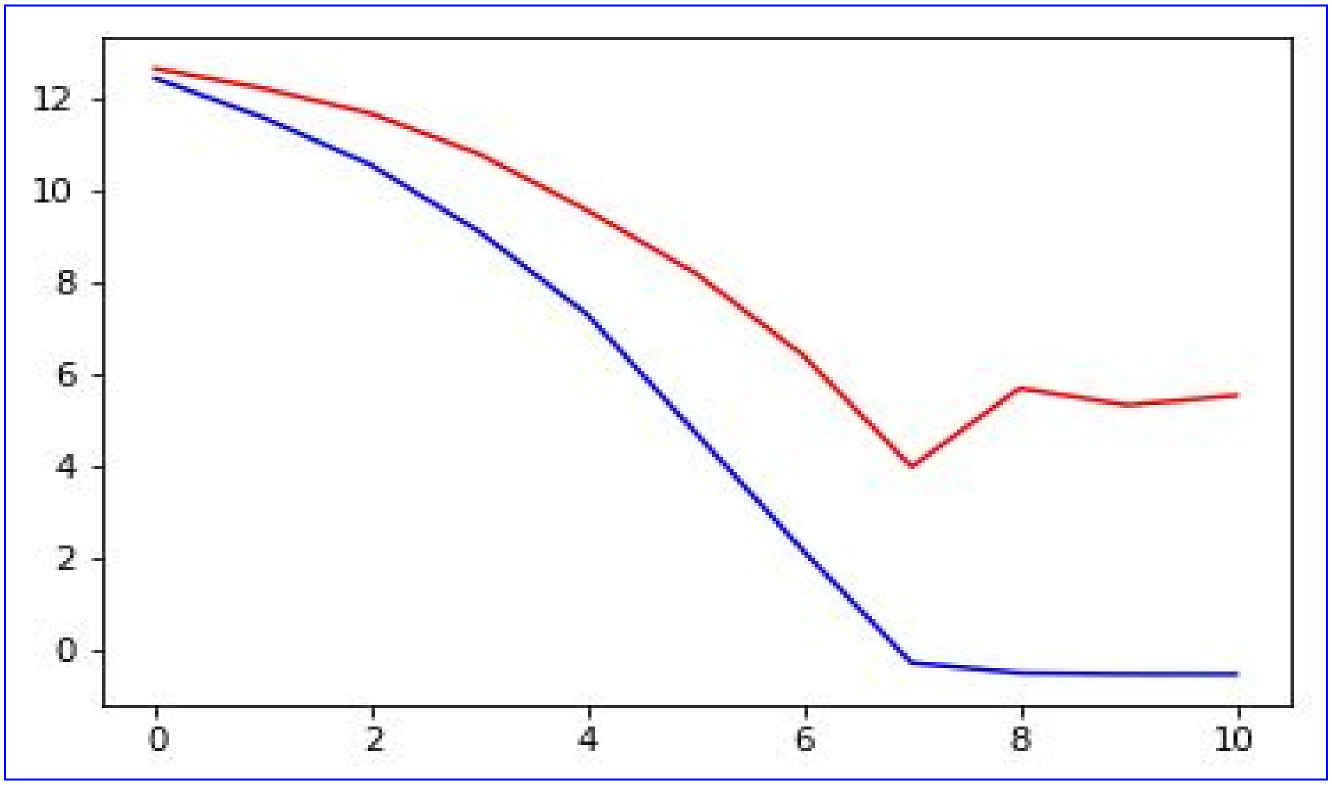
	0	1
0	0	-1015.86
1	0.857596	-496.962
2	0.987232	-227.569
3	0.999539	-34.2268
4	0.999993	-2.09208
5	1	0.846767
6	1	0.997399
7	1	0.99999
8	1	0.99945
9	1	0.999807
10	1	0.999699

As Clearly Visible that for Order 7 the r2 Values is maximum we choose the deg 7 as the degree of the model .

Optimal_degree = 7

Identification of UnderFit Overfit and Best fit :

Red : Test Error
Blue line : Train error



At degree 0 : Train and Test error both are very high : **UNDERFIT**
At degree 8 to 10 : Train Error :Low and Test Error : high : **OVERFIT**
At around degree 7 : both are stable and minimum : **BEST-FIT**

Same has been Indicated by the goodness fit measure .

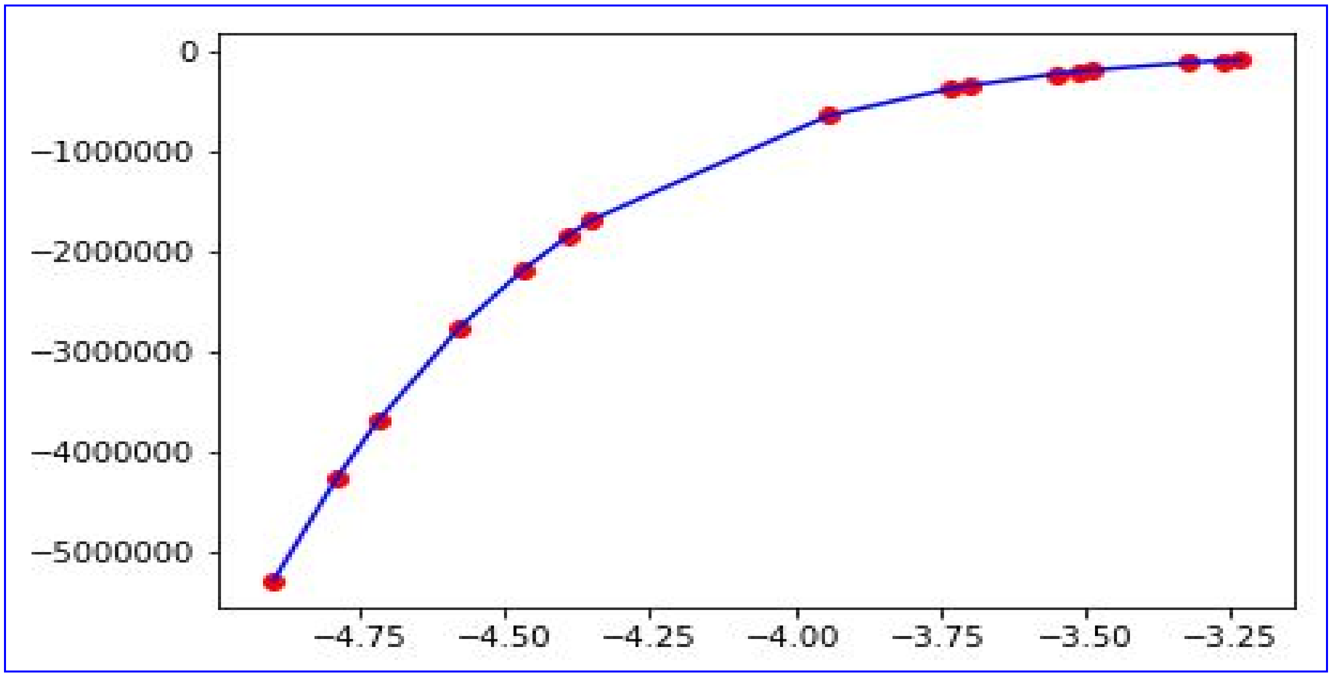
Now that we have got the best fit at degree = 7 we use the same to create our final model and apply train test split to visualize as well .

Polynomial Coefficients:
0.00000000e+00,1.83751104e+07,1.60463200e+07,7.91590331e+06 , 2.39956074e+06, 4.52254571e+05, 5.01250001e+04, 2.62356468e+03

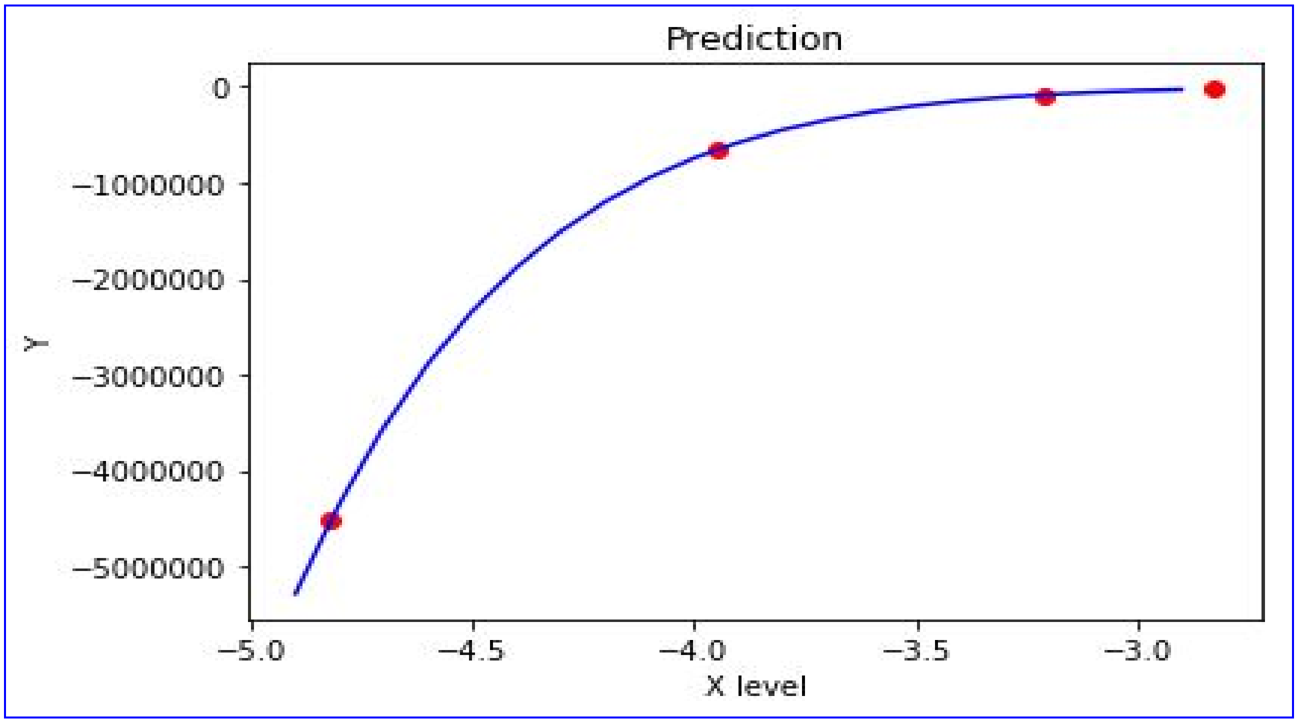
Intercept : 9127378.06773753

Variance : 1335.8439318633496

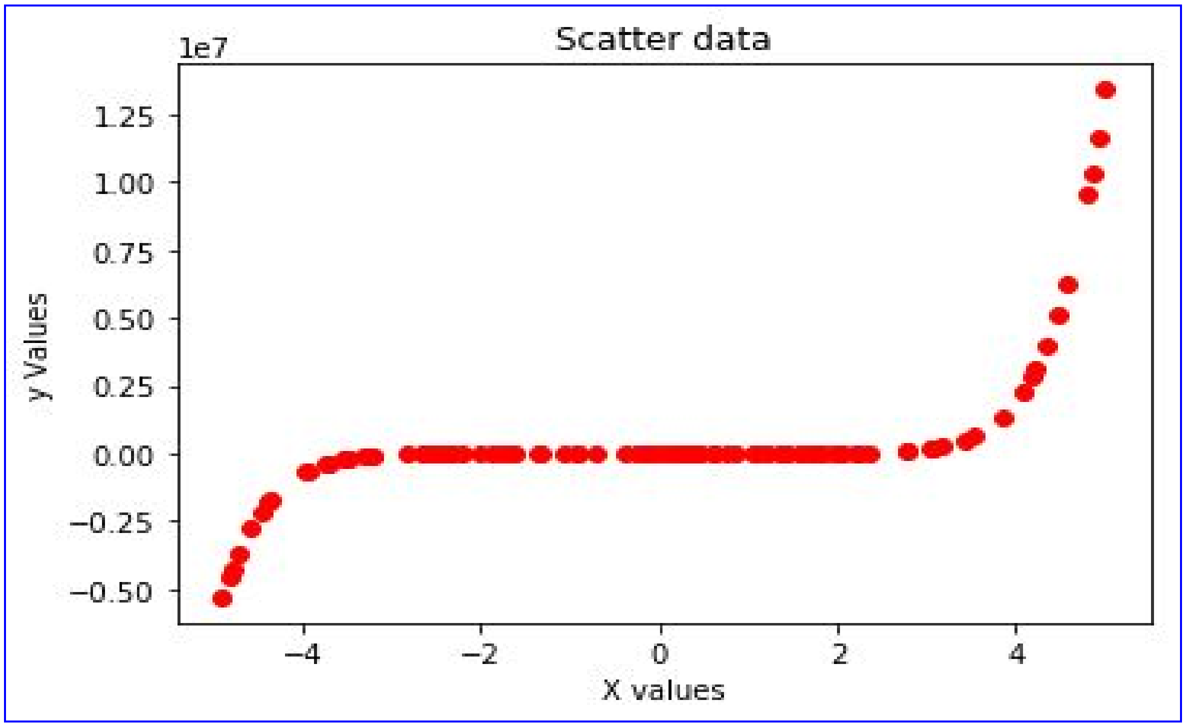
The training Set fit with optimal degree of 7 :



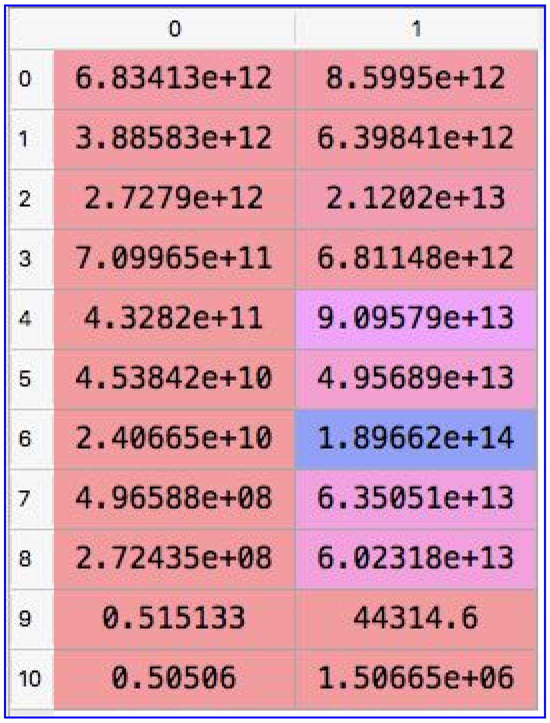
<u>The testing Set fit with optimal degree of 7 :</u>



Repeating the same experiments with now the whole data set :

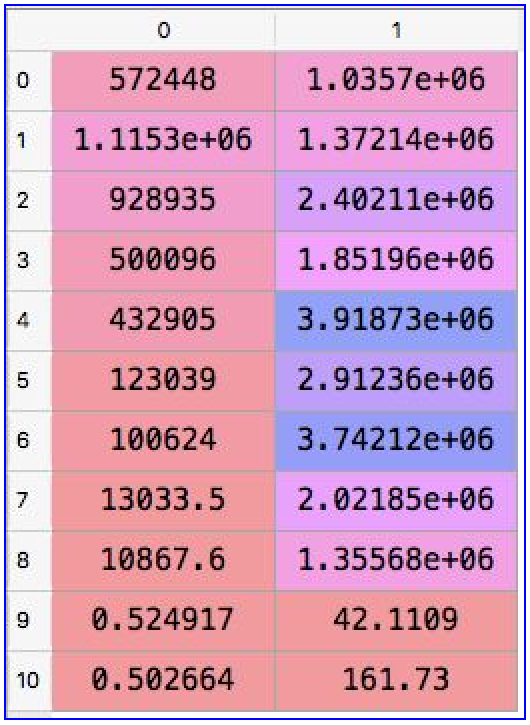


Mean Squared Error :



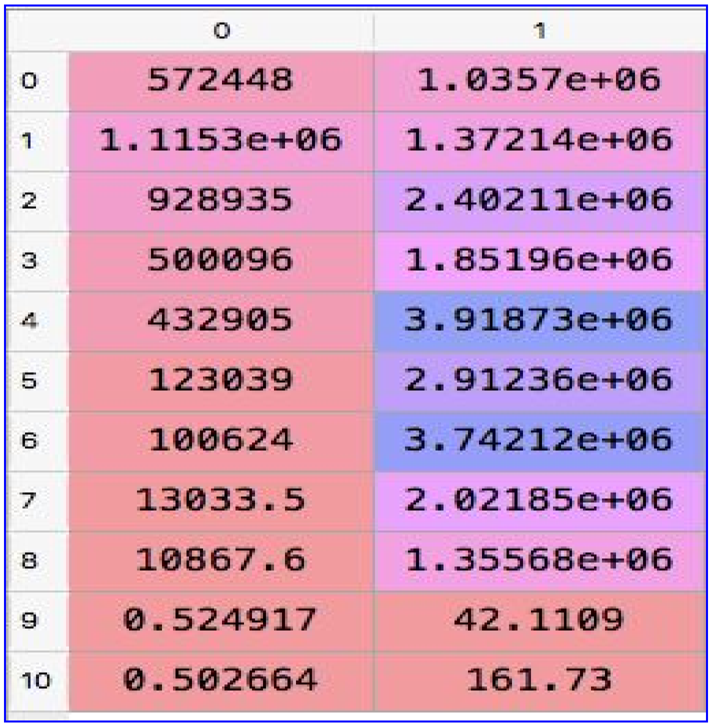
	0	1
0	6.83413e+12	8.5995e+12
1	3.88583e+12	6.39841e+12
2	2.7279e+12	2.1202e+13
3	7.09965e+11	6.81148e+12
4	4.3282e+11	9.09579e+13
5	4.53842e+10	4.95689e+13
6	2.40665e+10	1.89662e+14
7	4.96588e+08	6.35051e+13
8	2.72435e+08	6.02318e+13
9	0.515133	44314.6
10	0.50506	1.50665e+06

Mean Absolute Error :



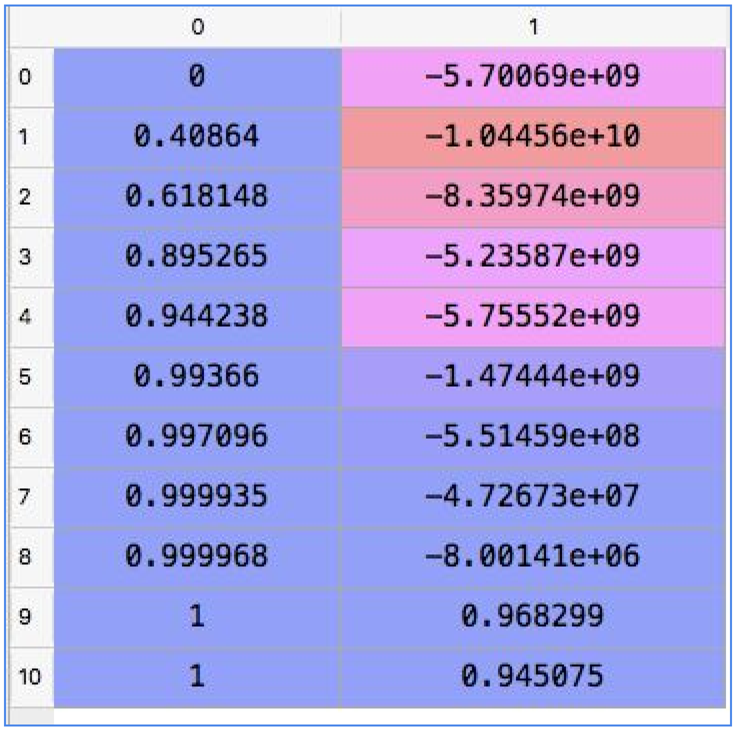
	0	1
0	572448	1.0357e+06
1	1.1153e+06	1.37214e+06
2	928935	2.40211e+06
3	500096	1.85196e+06
4	432905	3.91873e+06
5	123039	2.91236e+06
6	100624	3.74212e+06
7	13033.5	2.02185e+06
8	10867.6	1.35568e+06
9	0.524917	42.1109
10	0.502664	161.73

Median Absolute Error :



	0	1
0	572448	1.0357e+06
1	1.1153e+06	1.37214e+06
2	928935	2.40211e+06
3	500096	1.85196e+06
4	432905	3.91873e+06
5	123039	2.91236e+06
6	100624	3.74212e+06
7	13033.5	2.02185e+06
8	10867.6	1.35568e+06
9	0.524917	42.1109
10	0.502664	161.73

Goodness of Fit measure :



	0	1
0	0	-5.70069e+09
1	0.40864	-1.04456e+10
2	0.618148	-8.35974e+09
3	0.895265	-5.23587e+09
4	0.944238	-5.75552e+09
5	0.99366	-1.47444e+09
6	0.997096	-5.51459e+08
7	0.999935	-4.72673e+07
8	0.999968	-8.00141e+06
9	1	0.968299
10	1	0.945075

The Optimal Degree Obtained is : 9 .

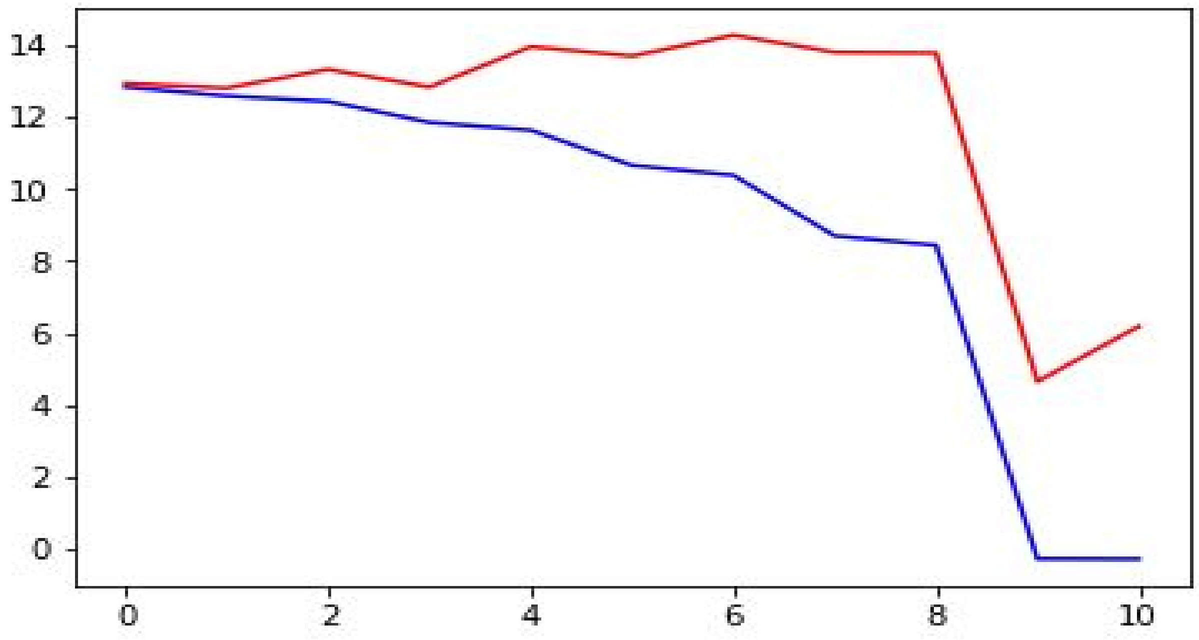
Observation:

For 20 data points the degree was 7 and now for 100 data points we get a optimal degree of 9 . So basically as we increase the data points the chances of learning better increase.

Now for 100 data points we have the below error function :

Red : Test Error

Blue line : Train error



At degree 0 : Train and Test error both are very high : **UNDERFIT**
At degree 10 : Train Error :Low and Test Error : high : **OVERFIT**
At around degree 9 : both are stable and minimum : **BEST-FIT**

Same has been Indicated by the goodness fit measure .

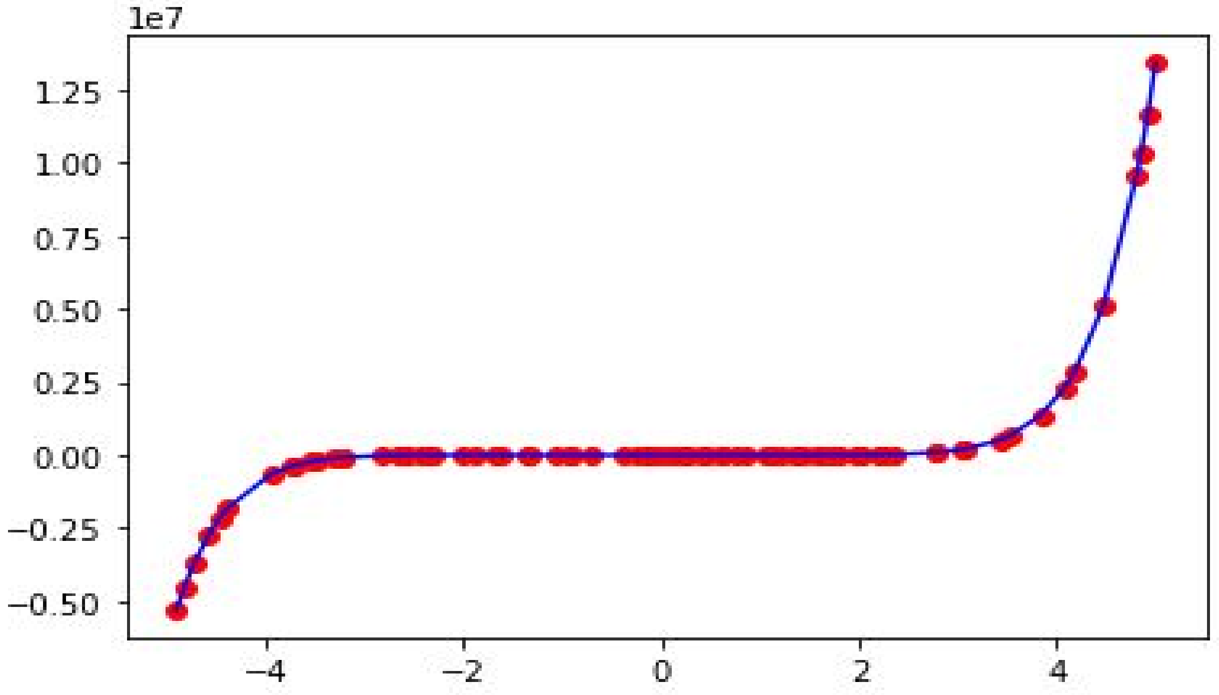
Now that we have got the best fit at degree = 9 we use the same to create our final model and apply train test split to visualize as well .

Polynomial Coefficients:
0, 5.97605199, 2.93962589, 5.66787466, 7.94284174, 8.40782067, 8.59773922, 1.49942007 8.60003644 5.00001346

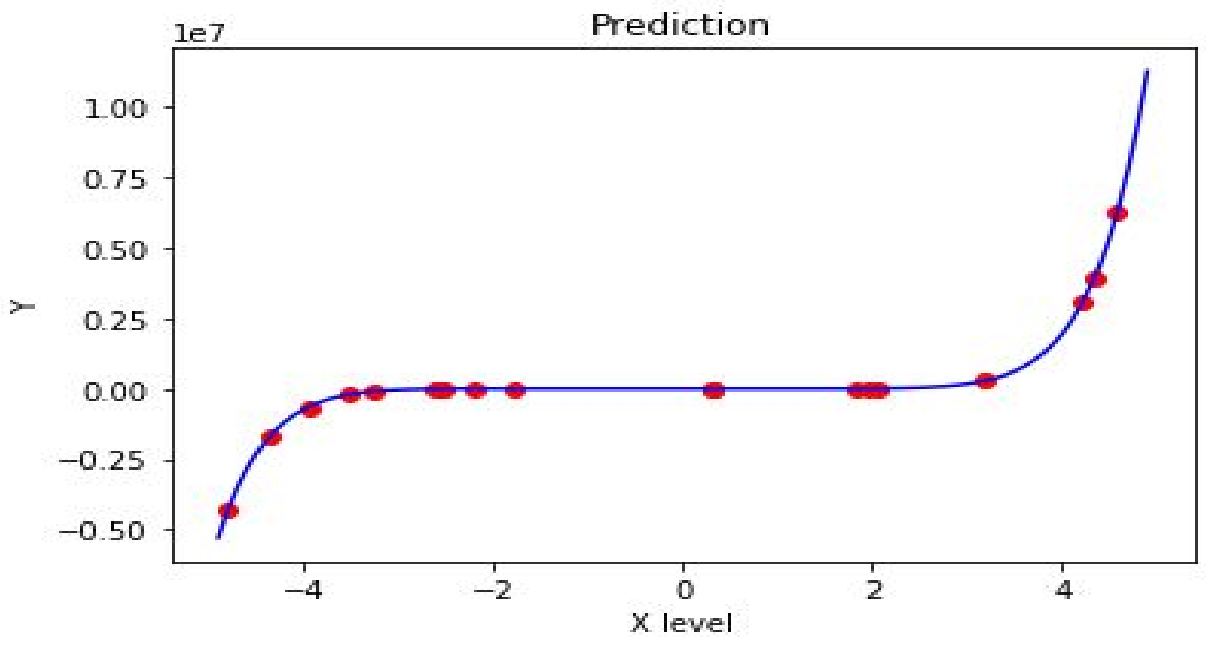
Intercept : 5.62820036

Variance : 0.5530508254255564

<u>The training Set fit with optimal degree of 9 :</u>



<u>The test Set fit with optimal degree of 9 :</u>



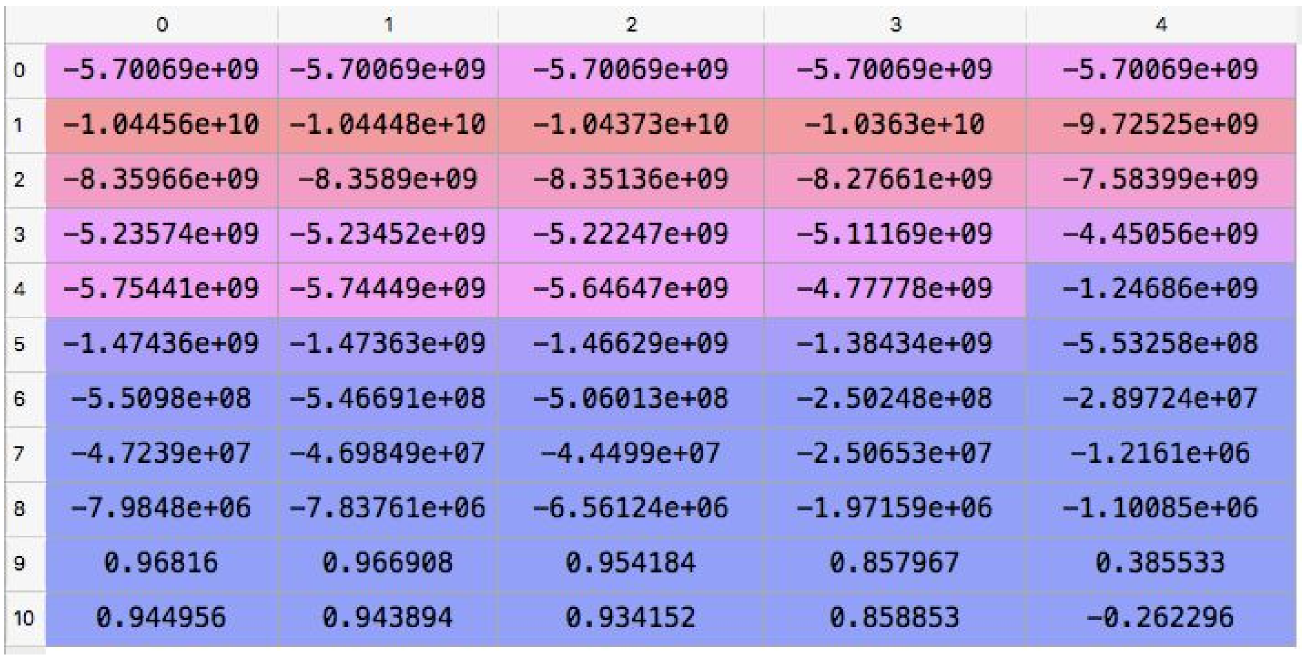
<u>Ridge Regularizer on data set of 100 points :</u>

Now we input regularization into the model which will help in smoothing the model curve. We first of all apply the regularizer to the data set of 20 points and see its prediction :

1. We vary the value of the alpha parameter as below possible values :
   alpha_range = [1e-2,1e-1,1,10,1e2]
2. For each one of them we iterate at each degree from 0 to 10 .
3. We are using KFOLD and all the errors and goodness of fit measure values are an average of the folds.

Ten we select the alpha and degree combination which gives us the best goodness of fit measure.

<u>The goodness of fit values for different alpha values are as depicted below:</u>



	0	1	2	3	4
0	-5.70069e+09	-5.70069e+09	-5.70069e+09	-5.70069e+09	-5.70069e+09
1	-1.04456e+10	-1.04448e+10	-1.04373e+10	-1.0363e+10	-9.72525e+09
2	-8.35966e+09	-8.3589e+09	-8.35136e+09	-8.27661e+09	-7.58399e+09
3	-5.23574e+09	-5.23452e+09	-5.22247e+09	-5.11169e+09	-4.45056e+09
4	-5.75441e+09	-5.74449e+09	-5.64647e+09	-4.77778e+09	-1.24686e+09
5	-1.47436e+09	-1.47363e+09	-1.46629e+09	-1.38434e+09	-5.53258e+08
6	-5.5098e+08	-5.46691e+08	-5.06013e+08	-2.50248e+08	-2.89724e+07
7	-4.7239e+07	-4.69849e+07	-4.4499e+07	-2.50653e+07	-1.2161e+06
8	-7.9848e+06	-7.83761e+06	-6.56124e+06	-1.97159e+06	-1.10085e+06
9	0.96816	0.966908	0.954184	0.857967	0.385533
10	0.944956	0.943894	0.934152	0.858853	-0.262296

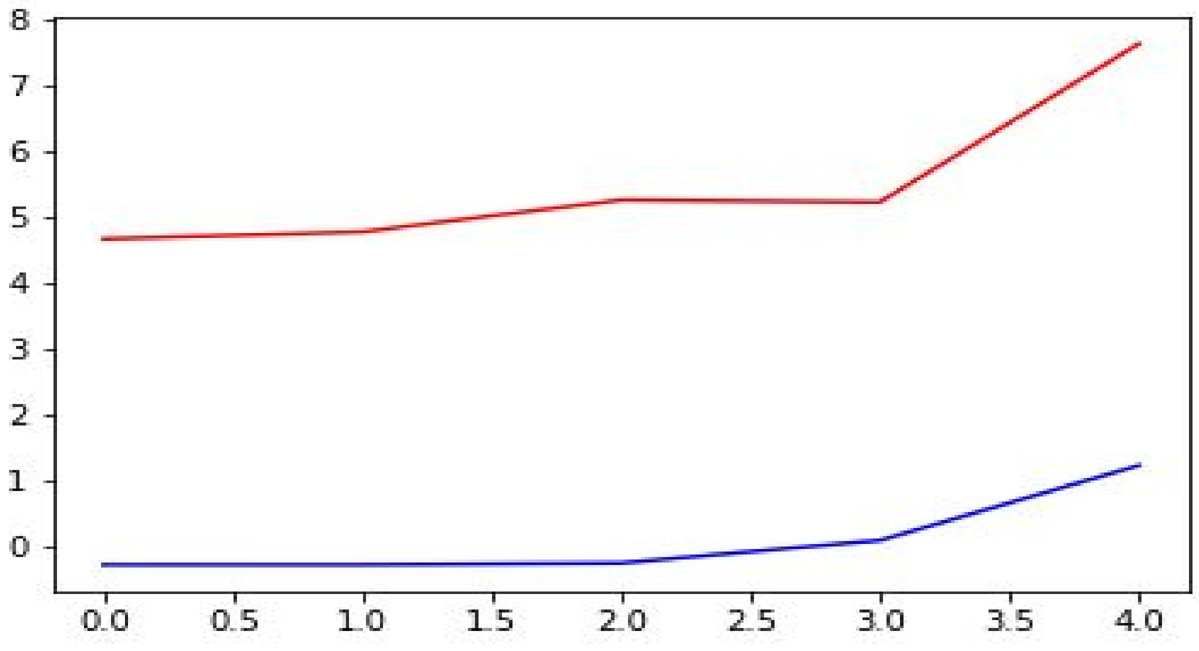
And the optimal values of :

Degree = 9
Alpha = 0.01

The value of degree is same as the data set without the regularizer . The difference might be noticed in the obtained coefficients and the intercept .

Now for 100 data points we have the below error function :

Red : Test Error
Blue line : Train error



At C = 100 : Train and Test error both are very high : **UNDERFIT**
At C = 0.01 : both are stable and minimum : **BEST-FIT after this test error starts increasing**

Same has been Indicated by the goodness fit measure .

Now that we have got the best fit at alpha = 0.01 and degree = 9 we use the same to create our final model and apply train test split to visualize as well .

Polynomial Coefficients:
0, 6.03943865, 3.42768939, 5.71569535, 7.73592179, 8.35476811, 8.61729302, 1.5077892 8.60080755, 5.00001039

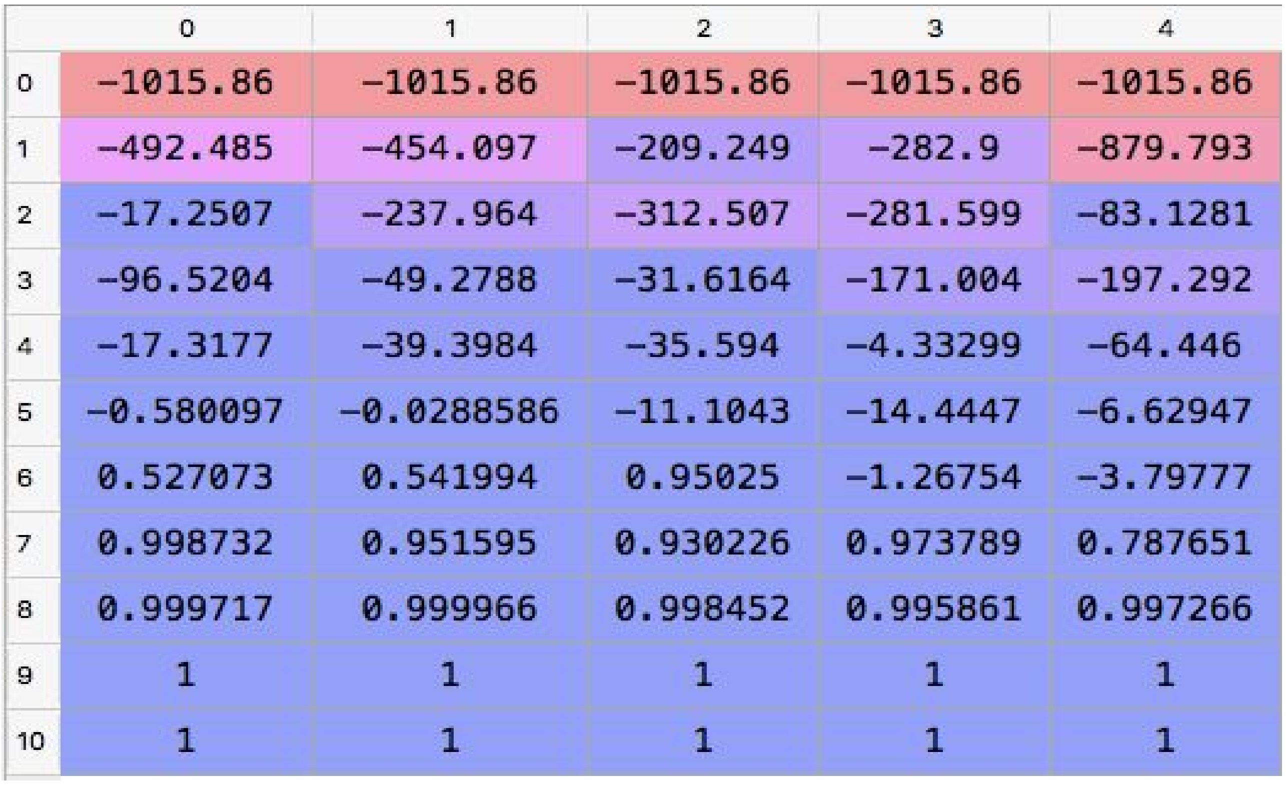
Intercept : 5.41377067

Variance : 32338.743972342894

Ridge Regularizer on data set of 20 points :

Same process as done with 100 data points .

The goodness of fit values for different alpha values are as depicted below:



	0	1	2	3	4
0	-1015.86	-1015.86	-1015.86	-1015.86	-1015.86
1	-492.485	-454.097	-209.249	-282.9	-879.793
2	-17.2507	-237.964	-312.507	-281.599	-83.1281
3	-96.5204	-49.2788	-31.6164	-171.004	-197.292
4	-17.3177	-39.3984	-35.594	-4.33299	-64.446
5	-0.580097	-0.0288586	-11.1043	-14.4447	-6.62947
6	0.527073	0.541994	0.95025	-1.26754	-3.79777
7	0.998732	0.951595	0.930226	0.973789	0.787651
8	0.999717	0.999966	0.998452	0.995861	0.997266
9	1	1	1	1	1
10	1	1	1	1	1

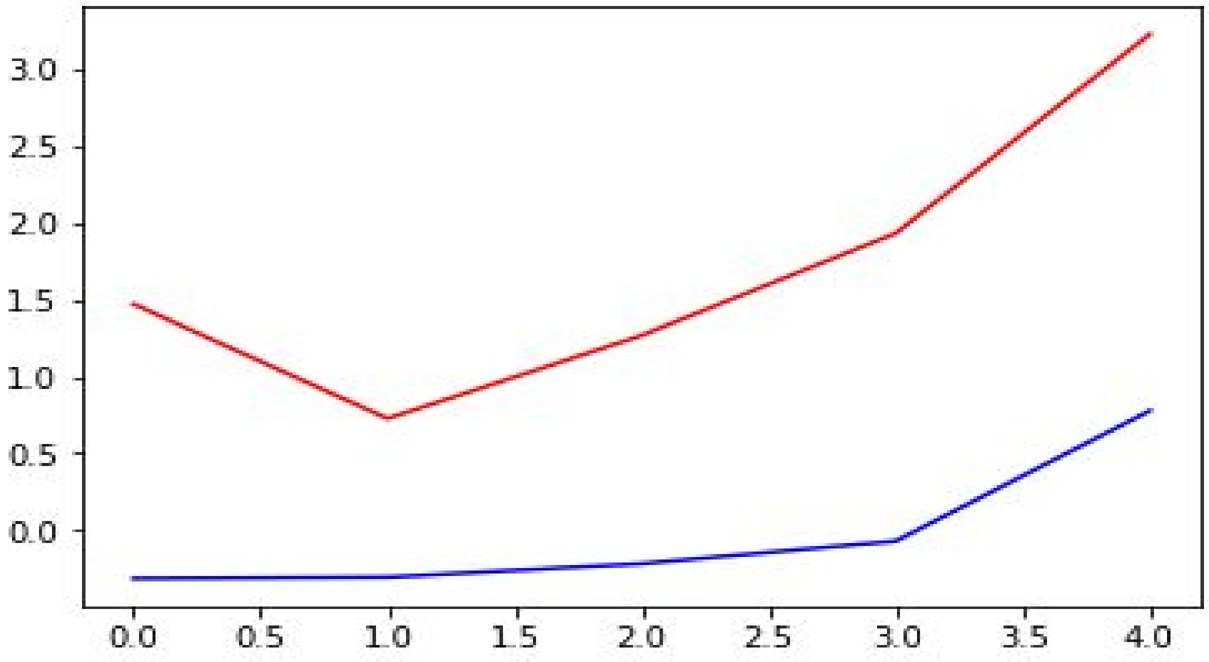
And the optimal values of :

Degree = 10
Alpha = 0.01

The value of degree has changed earlier it was 7 now its 9 , as because the data points are very very less the effect of the regularizer is evident .

<u>Now for 20 data points we have the below error function :</u>

Red : Test Error
Blue line : Train error



At C = 100 : Train and Test error both are very high : **UNDERFIT**
At C = 0.001 : Train and Test error both are very high : **OVERFIT**
At C = 0.01 : both are stable and minimum : **BEST-FIT after this test error starts increasing**

Same has been Indicated by the goodness fit measure .

Now that we have got the best fit at alpha = 0.01 and degree = 9 we use the same to create our final model and apply train test split to visualize as well .

Polynomial Coefficients:
0, 0.02071802, -0.07201634, 0.15407658, -0.20120288, 0.0214545, 0.69594328, -2.674471 7.46970985, 4.84972232, -0.00782562

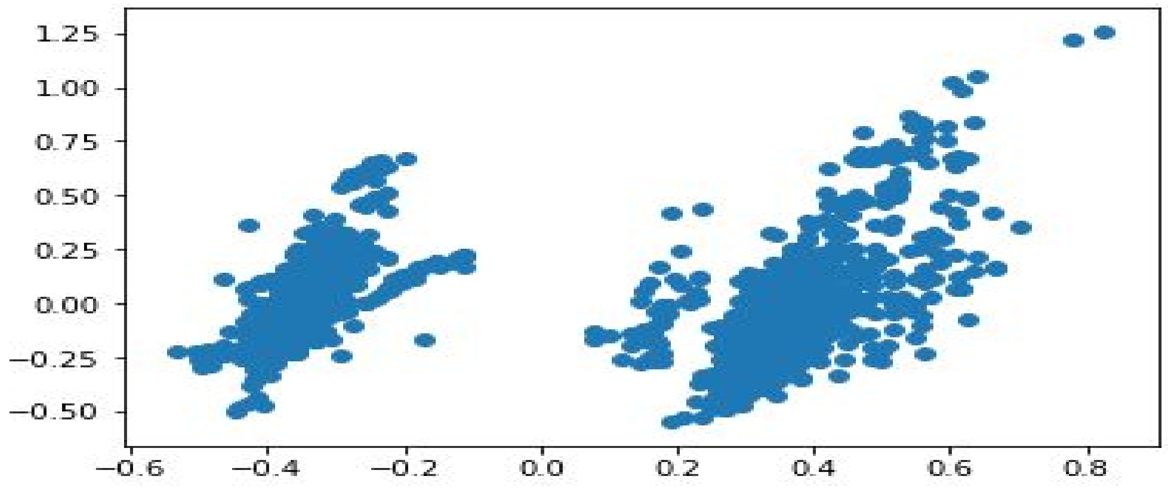
Intercept : 40.38255382

Variance : 1.2404

The SVM :

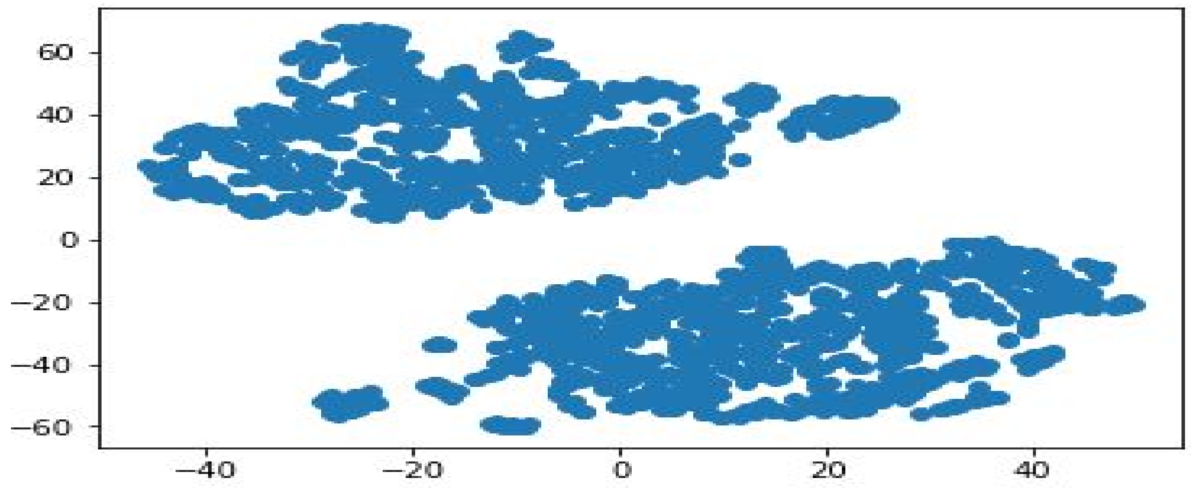
Here we first load the data set, in-order to analyze the data.
No of Train Records = 2000
No of Test Records = 1001
We are supposed to divide the train data itself in 50 percent train and 50 percent validation data.

The data is distributed as below if you use PCA with no of components as = 2.

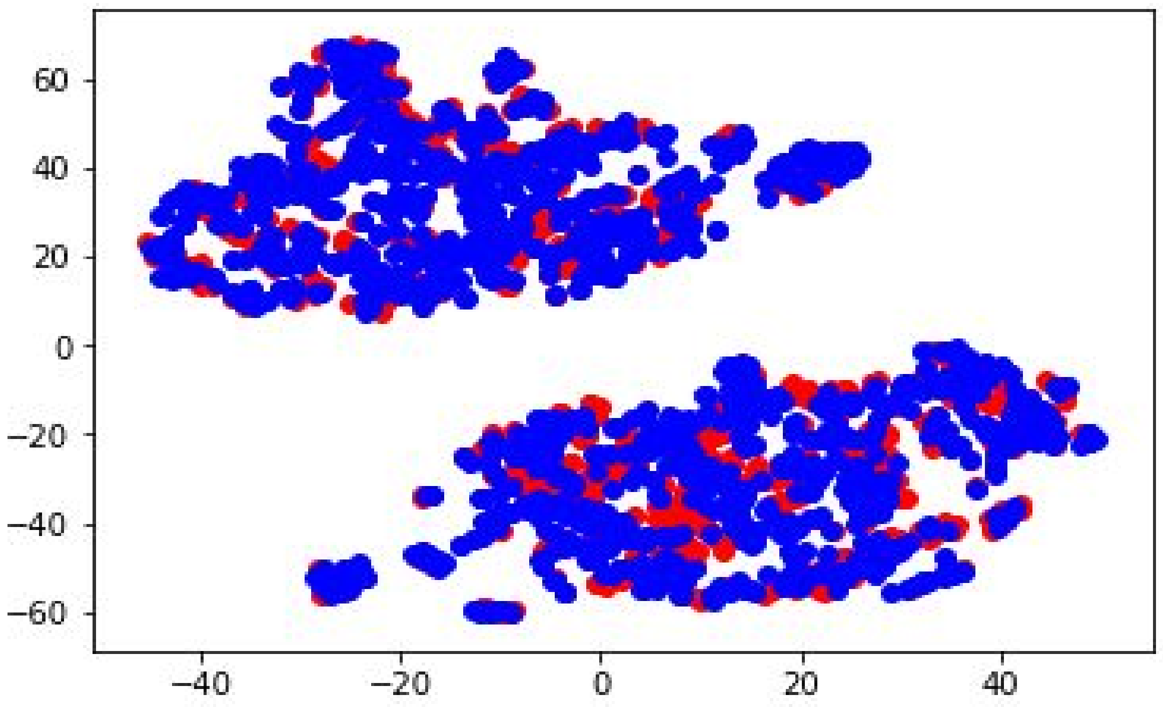


It seems that data is linearly separable , but let us use the t-SNE(**t**-distributed stochastic neighbor embedding) to project the data in 2 dimensional space.

The data is distributed as below if you use t-SNE with no of components as = 2.



Although the data seems separable clearly when projected to 2-D space but its not what we want we would like to project the data class wise we have 2 classes present in the data.



Now it gives us a better picture as we can now clearly see that that the data is mixed up when projected in the 2 dimensional space.

Kernel used : Linear

Now as part of the main code we will run through different values of the C parameter to find the best possible C with the highest degree of accuracy . During accuracy calculation we use the KFold library and take an average over the errors calculated for different folds.

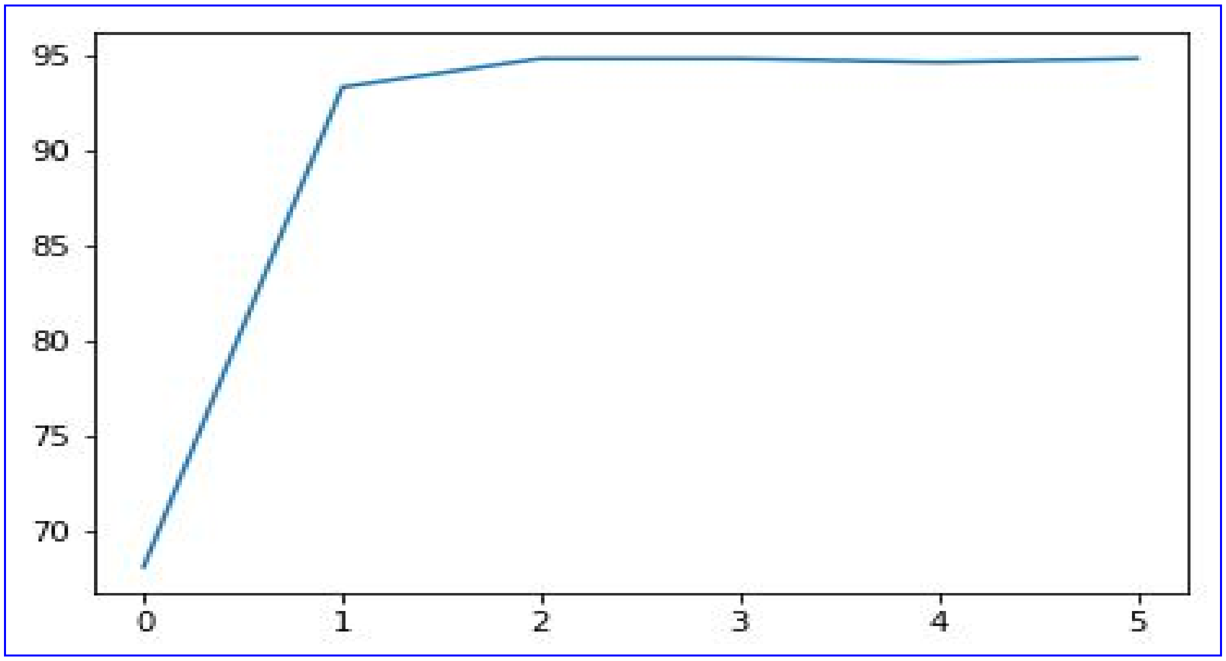
C_2d_range = [1e-3,1e-2,1e-1,1,10,1e2]

And we create the accuracy array to log the accuracy corresponding to each value of C .

Opt_C = 0.1
Max_Accuracy = 94.8

Accuracy_array = 94.8
[68.1,93.3, 94.8, 94.8, 94.6, 94.8]

classification accuracy as a function of the parameter *C:*



Polynomial Coefficients:
3.0090448, -1.48837243, 0.59240381, 0.38472849, 0.07690939, 0.55007765, 0.45035059 0.16824594
Intercept :
-0.9360495

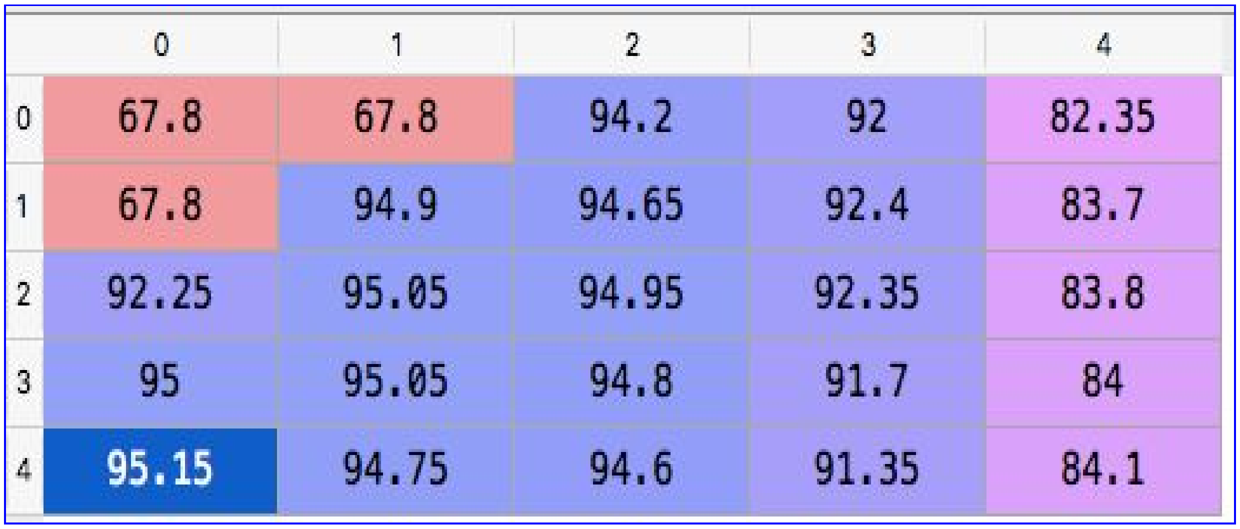
Kernel used : RBF

Now as part of the main code we will run through different combinations of the C and gamma parameters to find the best possible combination with the highest degree of accuracy . During accuracy calculation we use the KFold library and take an average over the errors calculated for different folds.

Now as part of the main code we will run through different values of the C parameter to find the best possible C with the highest degree of accuracy .

C_2d_range = [1e-3,1e-2,1e-1,1,10,1e2]
gamma_2d_range = [1e-1, 1, 1e1]

Accuracy Matrix where the entry *(i, j)* of the matrix corresponds to the classification accuracy on the cross validation set with *i*th value of *C* and *j* th value of *gamma:*



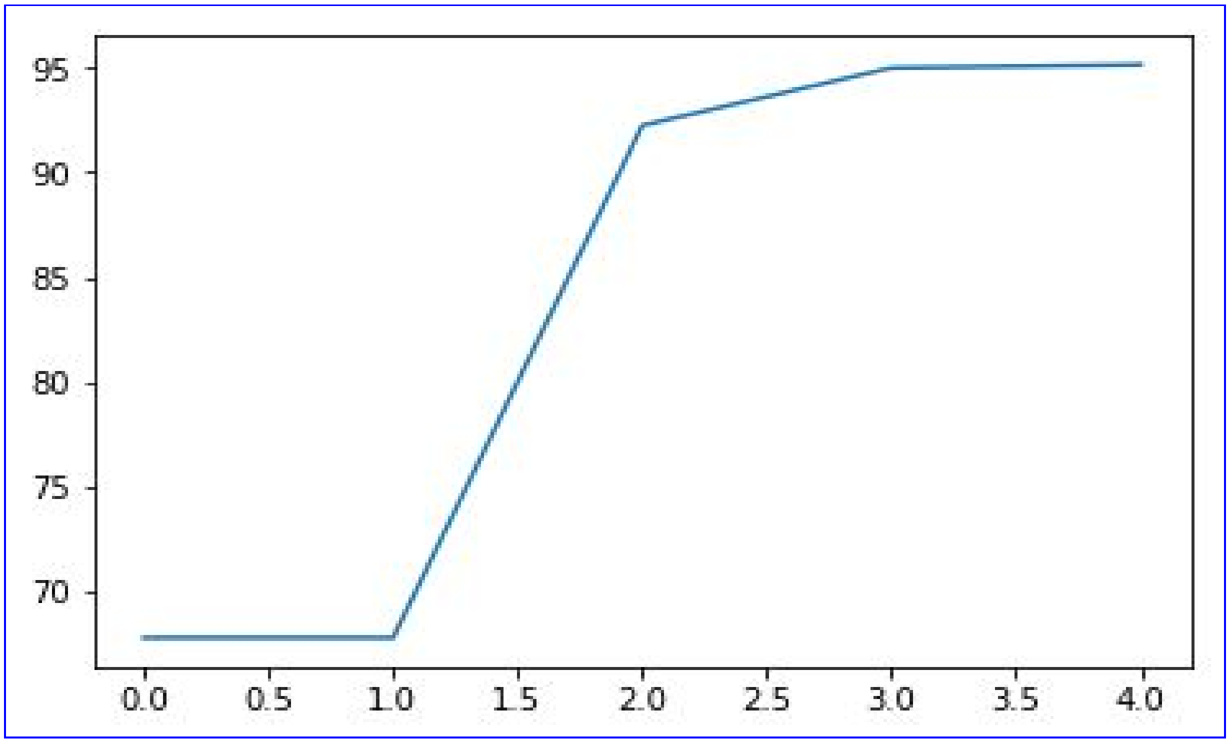
	0	1	2	3	4
0	67.8	67.8	94.2	92	82.35
1	67.8	94.9	94.65	92.4	83.7
2	92.25	95.05	94.95	92.35	83.8
3	95	95.05	94.8	91.7	84
4	95.15	94.75	94.6	91.35	84.1

And we create the accuracy array to log the accuracy corresponding to each value of C .

Maximum Accuracy : 95.15
Optimal Value of C is : 150
Optimal Value of Gamma : 0.01
Optimal Value of Sigma : 7.0710678

Corresponding Optimal Intercept : 6.42736528

classification accuracy as a function of the parameter *C (For Optimal gamma):*



Now we use these set of parameters obtained to predict the values for the test set and save them .