Intelligent Systems Lab

Lab No- 1

Name- Adhyyan Tripathi

Roll no -8

Sec C

Reg no – 201700403

Q) Write a program to implement the concepts of regression learnt in class via polynomial curve fitting.

Report-

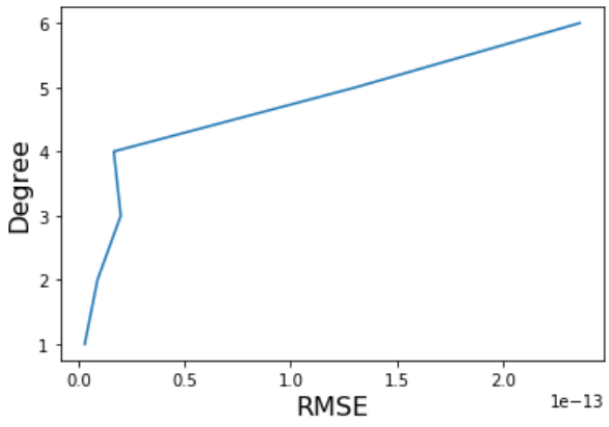
1. Using RMSE error function to analyse the dataset.

When we consider a small dataset, of value 20, the model generally produces very low RMSE value.

As the degree of the curve increases the RMSE value increases when the dataset is of small value.

```
For Degree  1 RMSE=  2.9036099814798757e-15 r2= 1.0
For Degree  2 RMSE=  8.906175611336929e-15 r2= 1.0
For Degree  3 RMSE=  1.987937954846129e-14 r2= 1.0
For Degree  4 RMSE=  1.6601157128353305e-14 r2= 1.0
For Degree  5 RMSE=  1.3074143970603592e-13 r2= 1.0
For Degree  6 RMSE=  2.3623187665836224e-13 r2= 1.0
```

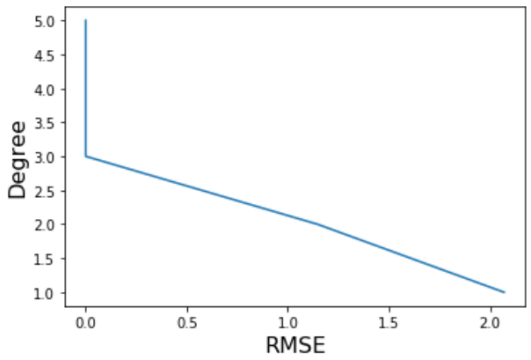
Out[5]: `[<matplotlib.lines.Line2D at 0x1fa66443f48>]`



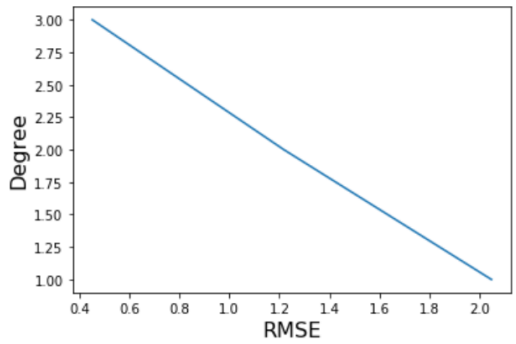
The above graph shows the relation of RMSE value with the degree of the curve. This phenomenon is recorded for 20 data points randomly collected from “Training_dataset.csv”.

When the same is observed for 2000, 5000 and 10,000 data points, the results are completely opposite to each other. The RMSE value for lower degrees is relatively higher than to the higher degrees.

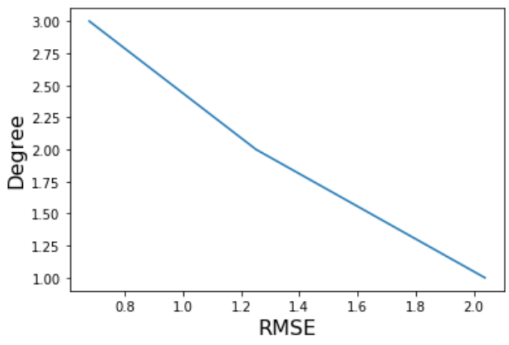
```
Out[1]: [<matplotlib.lines.Line2D at 0x127f5fdff08>]
```




```
Out[2]: [<matplotlib.lines.Line2D at 0x127f6f1bf08>]
```




```
Out[3]: [<matplotlib.lines.Line2D at 0x127f60912c8>]
```

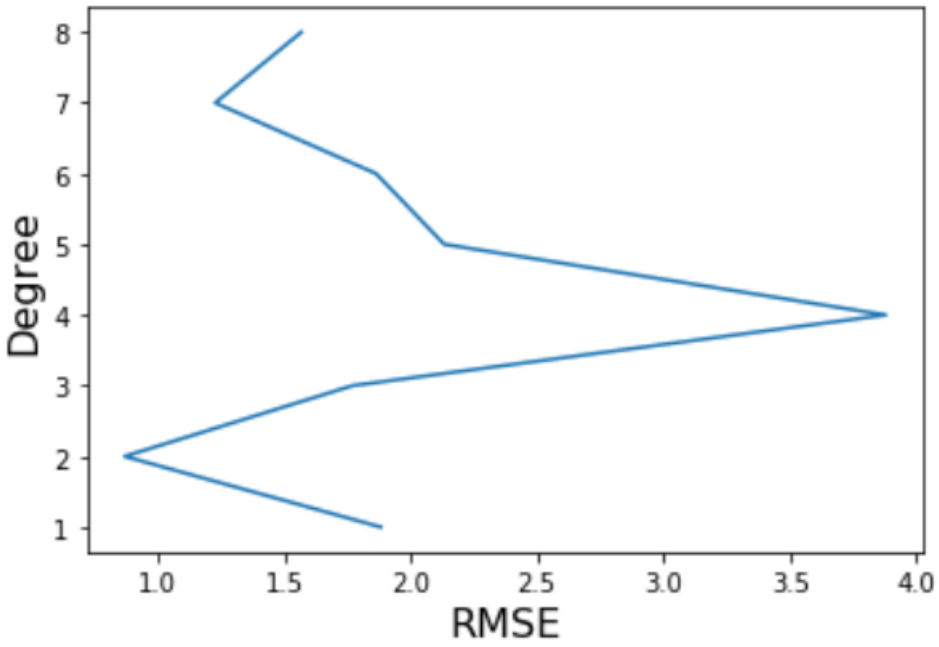


2. Using validation approach – Train test split (**For 20 data points**)

First, selecting 20 random data points from the given "Training_dataset.csv" and then, splitting the 20 data points to 80:20 ratio of training and testing.

The observation is given below-

```
RMSE=  1.8817507084499059 r2= -0.11089748352995232
RMSE=  0.867718844897183 r2= 0.6558006885532792
RMSE=  1.769122120006284 r2= 0.5363269517784401
RMSE=  3.8800115356493086 r2= -3.0825734282770725
RMSE=  2.1322950695223266 r2= -23.248972072049195
RMSE=  1.8624603006565335 r2= -1.9210596812813687
RMSE=  1.2261678588743532 r2= 0.5283176099963871
RMSE=  1.5669091205379826 r2= 0.017918323189954455
```



For degree 1- R2 value is below 0 which implies the model is below mean of the data, hence it is an underfit.

For degree 2- R2 value is the best among all the observed degrees also with the least RMSE value, hence it is the best fit.

For degree 3- R2 value is close to the best fit, but it is not equal to best fit.

For degree 4 to 7- R2 value is below 0, these are underfits.

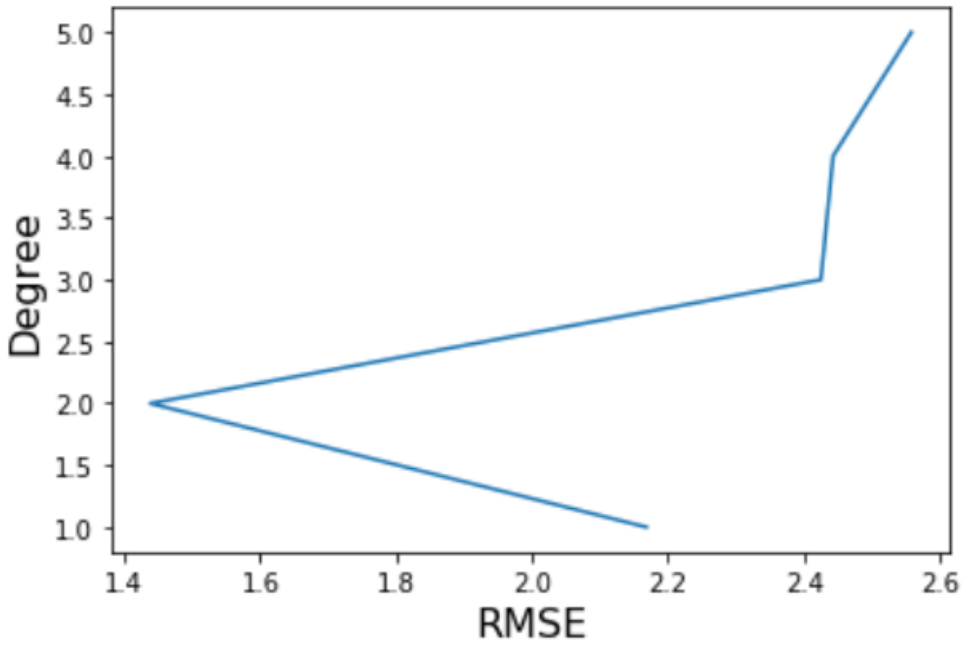
For degree 8 – R2 value is close to 0, so it is an underfit.

Train test split (**For 2000 data points**)-

First, selecting 2000 random data points from the given "Training_dataset.csv" and then, splitting the 2000 data points to 80:20 ratio of training and testing.

The observation is given below-

```
RMSE=  2.1684745163321386 r2= 0.4742525700018627
RMSE=  1.4380331700608247 r2= 0.7622210904810698
RMSE=  2.4253551851844675 r2= 0.34153637287958716
RMSE=  2.4433348785959335 r2= 0.3548005512980261
RMSE=  2.5585244956135305 r2= 0.21115264966102354
```



For degree 1 – R2 value is close to 0 so for degree 1 it is an underfit.

For degree 2- R2 value is the best fit among all the observations.

For degree 3- R2 value is close to 0 so it is an underfit.

For degree 4- R2 value is again an underfit.

For degree 5- R2 value is also an underfit.

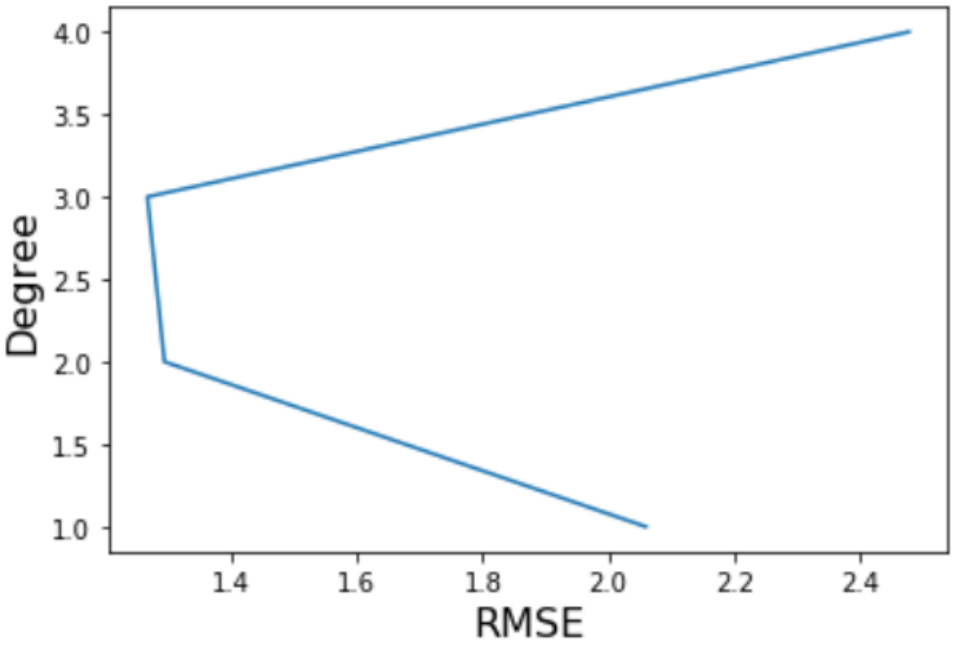
For degree 6 – R2 value is again close to 0 so it is an underfit.

Train test split (**For all 10000 data points**)-

First, selecting 10000 random data points from the given "Training_dataset.csv" and then, splitting the 10000 data points to 80:20 ratio of training and testing.

The observation is given below-

```
RMSE=  2.059902967234297 r2= 0.4984893506058947
RMSE=  1.2926106757787073 r2= 0.7961609546939208
RMSE=  1.2654596724495526 r2= 0.813248995527094
RMSE=  2.479141192320283 r2= 0.26865607226463706
```



For degree 1- R2 value is more, close towards 0 so it is an underfit.

For degree 2- R2 value is close to best fit but not best fit.

For degree 3- R2 value is the best fit.

For degree 4- R2 value is more, close towards 0 so it is an underfit.

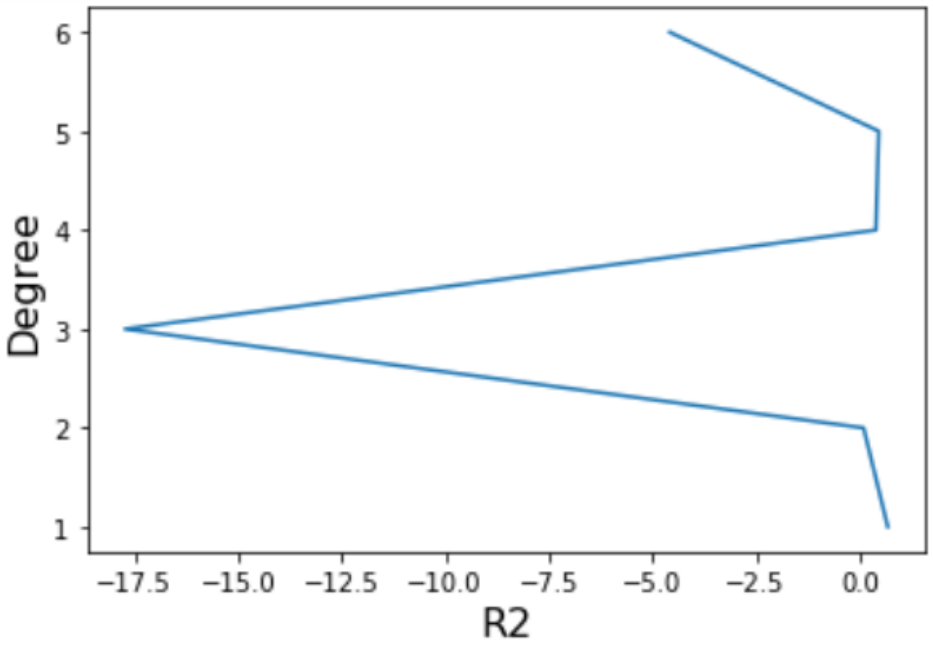
Introducing **Regularization (L2 Ridge regularization)**-

When the model is regularized, it helps avoiding overfitting of the model. This can be observed by the R2 score of every degree. The most optimal value for

regularization parameter in my case is found out to be 0.25 with cross validation score -1247.652413161.

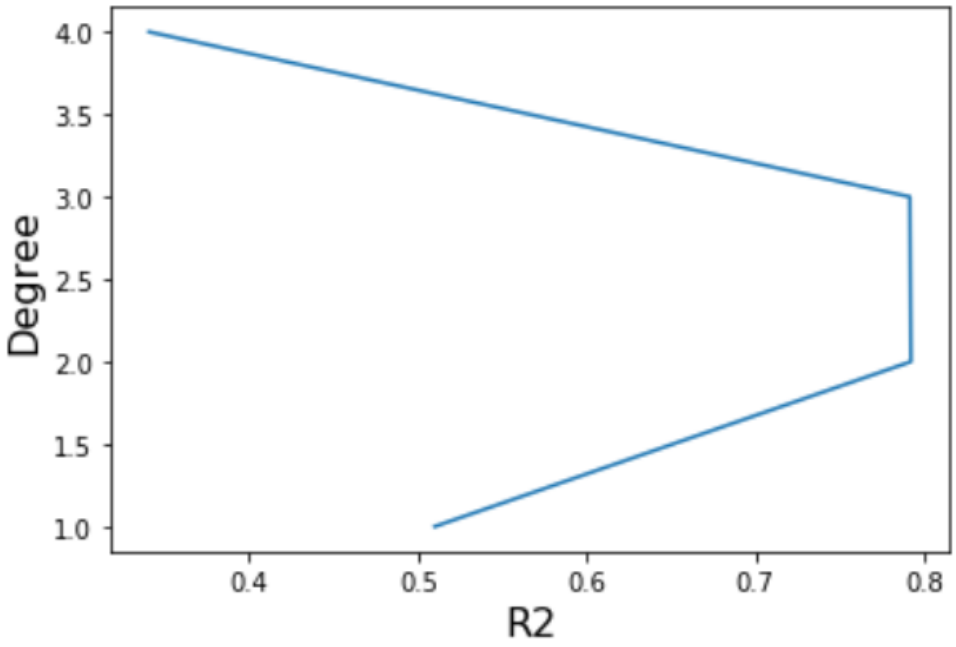
For 20 datapoints-

```
For degree 1-
RMSE=  1.512978883979078 r2= 0.6842613650528859
For degree 2-
RMSE=  2.8849073288205096 r2= 0.10024969774245529
For degree 3-
RMSE=  3.059824067575352 r2= -17.72504664902674
For degree 4-
RMSE=  1.7662760913943452 r2= 0.39860602775300524
For degree 5-
RMSE=  1.6660466455922813 r2= 0.4649230987394134
For degree 6-
RMSE=  2.5732533102878663 r2= -4.576111662237862
```



For 10000 datapoints-

```
For degree 1-
RMSE=  2.0129909749783406 r2= 0.5098981381251297
For degree 2-
RMSE=  1.317377480822378 r2= 0.7920267561268236
For degree 3-
RMSE=  1.334748399030543 r2= 0.7914053692672142
For degree 4-
RMSE=  2.3707904506463104 r2= 0.3406484543490814
```



Conclusion –

- For a small dataset the RMSE value decreases as we increase the degree.
- For a large dataset the RMSE value increases as we increase the degree.
- When train test split is applied, we came to know about underfitting, best fitting and overfitting.
- For small dataset the model is usually underfitted, and for larger dataset as the degree increases, the model gets overfitted.
- When we apply regularization, it prevents our model from getting overfitted.