Machine Learning CS60050

# Assignment 1

# Curve Fitting With Gradient Descent

Shivam Kumar Jha

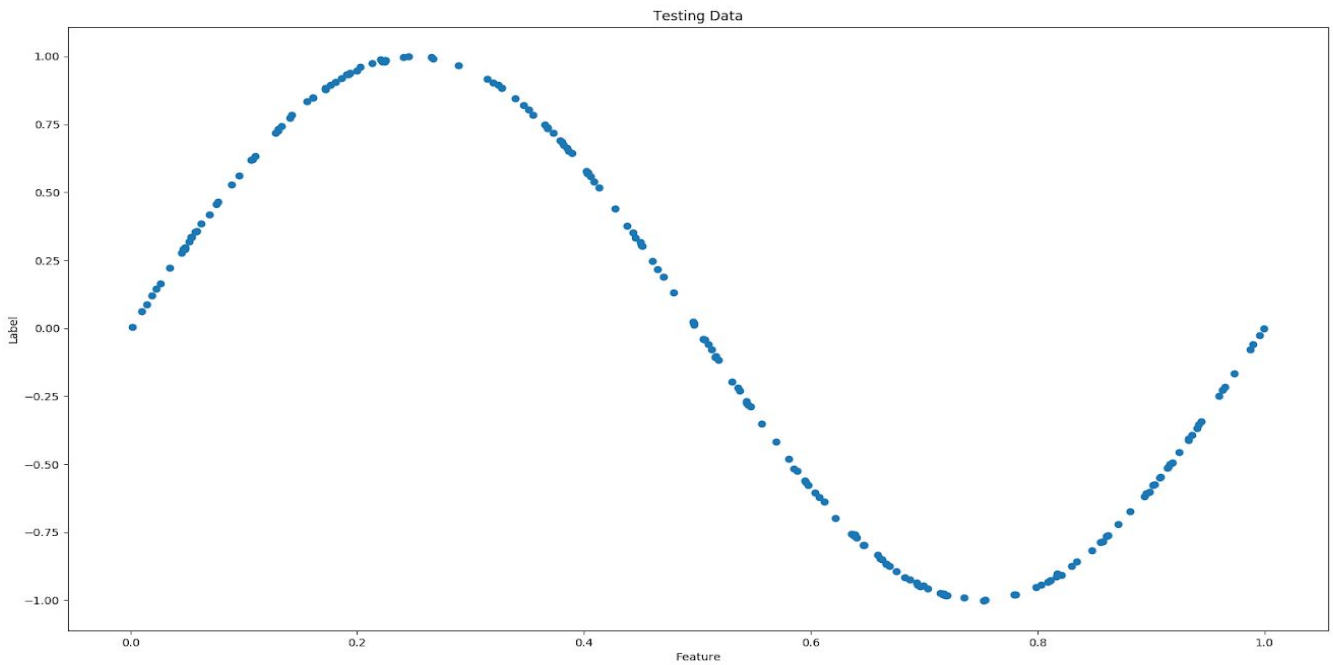
17CS30033

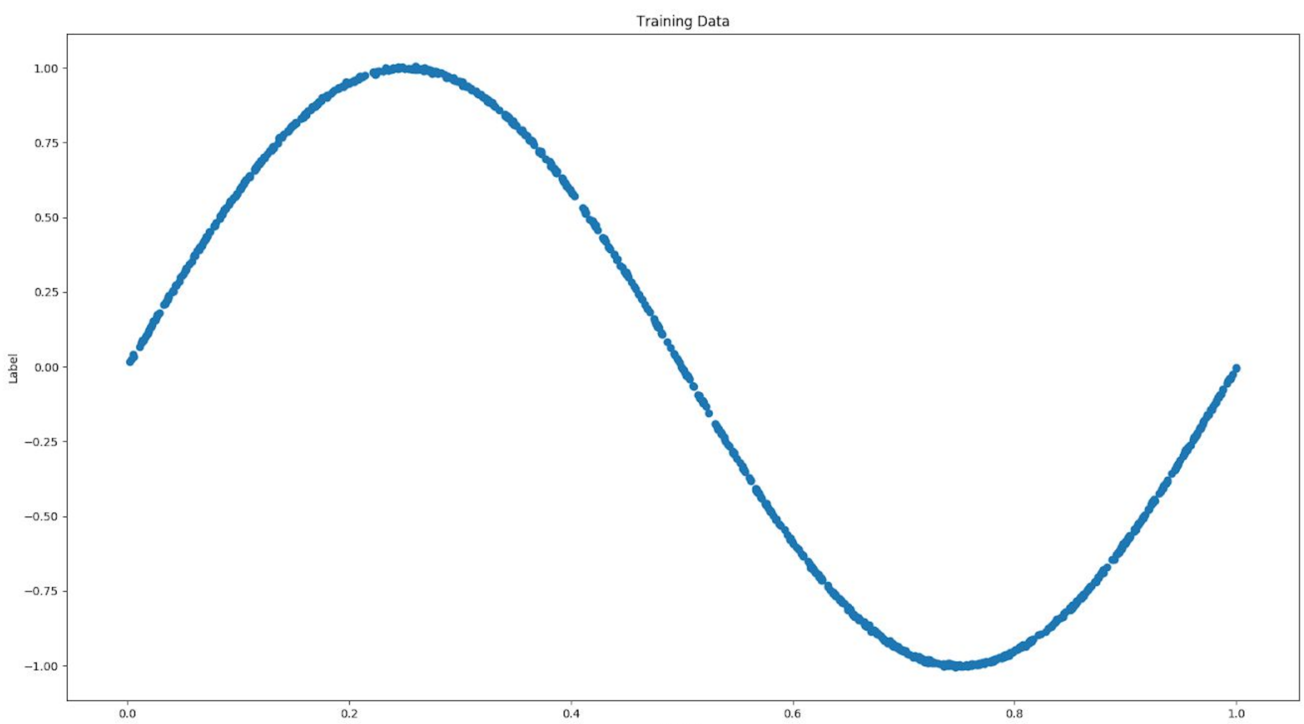
## General Information

I've used Python as the language for doing the assignment and information on how to run the code is written in README.pdf. A **learning rate of 0.05** has been used along with a converge criteria that the difference between **two consecutive costs should not be greater than 0.00000001**. All the plots are also present in the submission folder.

Initially, I tried with a more relaxed convergence point but the graphs were not coming with much accuracy. I also started using python itself but later switched to numpy vectors for speed. Vectorization involves expressing mathematical operations, such as the multiplication we're using here, as occurring on entire arrays rather than their individual elements (as in a for-loop). With vectorization, the underlying code is parallelized such that the operation can be run on multiply array elements at once, rather than looping through them one at a time.

# Answer To 1a



The above plots are also available in higher resolution as "testing_data_plot.png" and "training_data_plot.png"

---

## Answer to 1b

The predicted labels for the features are present in the folder "predicted_labels" with names "predicted_labels_n.csv" where n represents the degree of the polynomial.

Below are the details for predicted parameters for different degrees of the polynomial.

NOTE: The parameters for the polynomials are also present in the file "predicted_parameters.txt".

### =========== Polynomial Degree 1 ===========

Parameters: [ 0.91609257 -1.85515881]

Squared Error on Test Data: **0.0955305340264786**

### =========== Polynomial Degree 2 ===========

Parameters: [ 0.97374505 -2.20096505 0.34052727]

Squared Error on Test Data: **0.09579851920170313**

### =========== Polynomial Degree 3 ===========

Parameters: [ -0.07671096 10.50647719 -31.22342516 20.91009886]

Squared Error on Test Data: **0.003248850901895813**

## =========== Polynomial Degree 4 ===========

Parameters: [ 0.08304287 7.17832471 -15.64570255 -3.95364149 12.65570878]

Squared Error on Test Data: **0.004675914873744095**

## =========== Polynomial Degree 5 ===========

Parameters: [ 0.19224162 5.40744098 -10.24041835 -4.97457161 2.51910479
7.5532347 ]

Squared Error on Test Data: **0.008864550892919437**

## =========== Polynomial Degree 6 ===========

Parameters: [ 0.07187978 7.23221242 -15.80129906 -2.21439591 7.14343828
6.20216435 -2.3066975 ]

Squared Error on Test Data: **0.0045908198829193265**

## =========== Polynomial Degree 7 ===========

Parameters: [ 0.03408172 7.64689745 -16.09554391 -3.72309084 6.58913232
8.27255123 3.22195633 -5.75157165]

Squared Error on Test Data: **0.002333572675940087**

## =========== Polynomial Degree 8 ===========

Parameters: [ 0.03688883 7.44725993 -14.80938571 -4.95991551 4.64976525
7.86202875 5.78645938 0.48757847 -6.41213449]

Squared Error on Test Data: **0.0014136094363091148**

=========== **Polynomial Degree 9** ===========

Parameters: [ 0.05625352 7.05546052 -13.32894872 -5.57128763 2.93826133 6.66167982 6.23979143 3.21321827 -1.1720244 -6.09228559]

Squared Error on Test Data: **0.0012158219405710774**

---

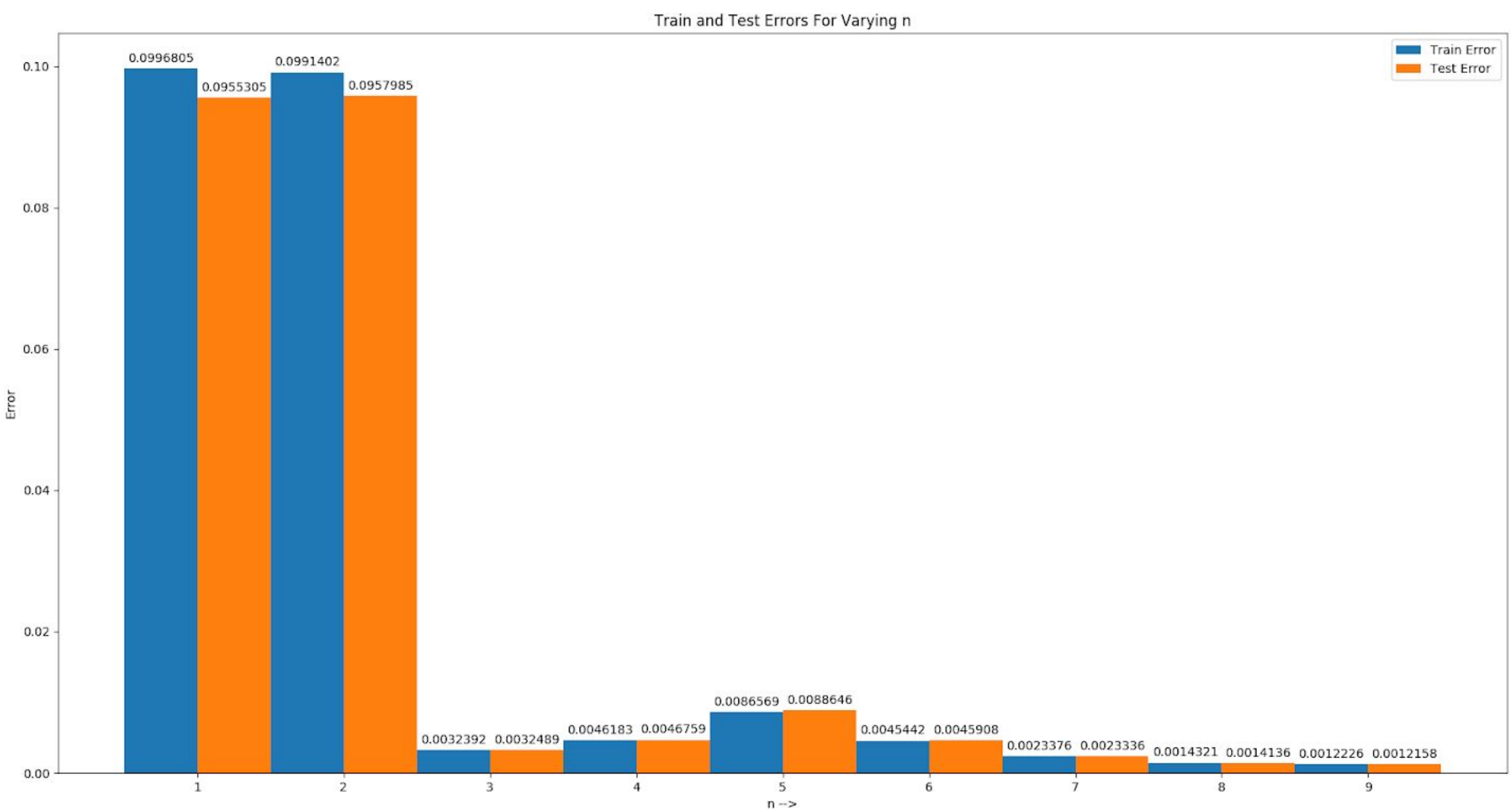
## Answer to 2a

All the plots of the predicted polynomials are available in the folder “polynomial_plots” with names “pol_n.png” where n denotes the degree of the polynomial.

---

## Answer to 2b

A higher resolution version of the same is available in the folder “result_plots” with name “train_test_varying_n.png”.



Clearly we can see that the predicted polynomial of degree 9 has the least amount of squared error in both test data and train data and hence it is the best fit.

---

## Answer to 3

The best curve as we can see above **is one with degree 9** and the **worst is one with degree 1** based on the training error. Below are the results obtained on regularisation over varying lambdas with same convergence as initial. Later on the graphs are present with discussion.

### ======= Lasso Regularisation Best Curve: Polynomial Degree 9 Lambda 0.25

Polynomial is: [ 0.05625352 7.05546052 -13.32894872 -5.57128763 2.93826133

6.66167982 6.23979143 3.21321827 -1.1720244 -6.09228559]

New Parameters: [ 0.42953137 3.11862538 -6.55988341 -3.47077663 0.4163378
2.4629589 2.78370323 1.95956445 0.49187971 -1.2750234 ]

Lasso Error on Test Data: 0.029714648357586894

Squared Error on Test Data: 0.015359470675159756

### ====== Lasso Regularisation Best Curve: Polynomial Degree 9 Lambda 0.5

Polynomial is: [ 0.05625352 7.05546052 -13.32894872 -5.57128763 2.93826133

6.66167982 6.23979143 3.21321827 -1.1720244 -6.09228559]

New Parameters: [ 0.09196533 6.66690367 -12.73702703 -5.33170068 2.78721081

6.3447844 5.95234688 3.07802229 -1.09101373 -5.7700993 ]

Lasso Error on Test Data: 0.06408823708329009

Squared Error on Test Data: 0.0017743944386146224

### ====== Lasso Regularisation Best Curve: Polynomial Degree 9 Lambda 0.75

Polynomial is: [ 0.05625352 7.05546052 -13.32894872 -5.57128763 2.93826133

6.66167982 6.23979143 3.21321827 -1.1720244 -6.09228559]

New Parameters: [ 0.18290551 5.80277059 -11.43934088 -4.83501561 2.41707769

5.62140738 5.31593633 2.80247204 -0.86529818 -4.99051297]

Lasso Error on Test Data: 0.0866127977135818

Squared Error on Test Data: 0.0036014155026287825

### ======= Lasso Regularisation Best Curve: Polynomial Degree 9 Lambda 1

Polynomial is: [  0.05625352   7.05546052 -13.32894872  -5.57128763   2.93826133

6.66167982   6.23979143   3.21321827  -1.1720244   -6.09228559]

New Parameters: [  0.27966654   4.86953959 -10.00491075  -4.30509084   1.98486954

4.80518057   4.60712516   2.50507031  -0.59674589  -4.09935204]

Lasso Error on Test Data: 0.10168082877204335

Squared Error on Test Data: 0.006536950707455677

### ====== Lasso Regularisation Worst Curve: Polynomial Degree 1 Lambda 0.25

Polynomial is: [ 0.91609257 -1.85515881]

New Parameters: [ 0.91610343 -1.85515503]

Lasso Error on Test Data: 0.0972626801135909

Squared Error on Test Data: 0.09553064357265667

### ====== Lasso Regularisation Worst Curve: Polynomial Degree 1 Lambda 0.5

Polynomial is: [ 0.91609257 -1.85515881]

New Parameters: [ 0.91610343 -1.85513184]

Lasso Error on Test Data: 0.09899468774549364

Squared Error on Test Data: 0.09553064365048165

### ====== Lasso Regularisation Worst Curve: Polynomial Degree 1 Lambda 0.75

Polynomial is: [ 0.91609257 -1.85515881]

New Parameters: [ 0.91610343 -1.85508546]

Lasso Error on Test Data: 0.10245861658275607

Squared Error on Test Data: 0.09553064434015798

### ======= Lasso Regularisation Worst Curve: Polynomial Degree 1 Lambda 1

Polynomial is: [ 0.91609257 -1.85515881]

New Parameters: [ 0.91610343 -1.85508546]

Lasso Error on Test Data: 0.10245861658275607

Squared Error on Test Data: 0.09553064434015798

### ======== Ridge Regularisation Best Curve: Polynomial Degree 9 Lambda 0.25

Polynomial is: [ 0.05625352 7.05546052 -13.32894872 -5.57128763 2.93826133

6.66167982 6.23979143 3.21321827 -1.1720244 -6.09228559]

New Parameters: [ 0.40377039 3.40262128 -7.07440987 -3.61156752 0.6259763
2.78686305 3.03791492 2.04020454 0.35404736 -1.64015965]

Ridge Error on Test Data: 0.07569459385177932

Squared Error on Test Data: 0.01369656646094227

### ======= Ridge Regularisation Best Curve: Polynomial Degree 9 Lambda 0.5

Polynomial is: [ 0.05625352 7.05546052 -13.32894872 -5.57128763 2.93826133

6.66167982 6.23979143 3.21321827 -1.1720244 -6.09228559]

New Parameters: [ 0.53791281 2.06331727 -4.91637 -2.87311896 -0.09242278
1.51447279 1.96297647 1.61908126 0.81218578 -0.22644409]

Ridge Error on Test Data: 0.07976417261481214

Squared Error on Test Data: 0.021689262491799456

### ======= Ridge Regularisation Best Curve: Polynomial Degree 9 Lambda 0.75

Polynomial is: [ 0.05625352 7.05546052 -13.32894872 -5.57128763 2.93826133

6.66167982 6.23979143 3.21321827 -1.1720244 -6.09228559]

New Parameters: [ 0.60663919 1.41729639 -3.96417244 -2.49765432 -0.35029837
0.99376532 1.49929986 1.41247137 0.96595344 0.32716295]

Ridge Error on Test Data: 0.08365561693418147

Squared Error on Test Data: 0.02604992456242261

### ======= Ridge Regularisation Best Curve: Polynomial Degree 9 Lambda 1

Polynomial is: [ 0.05625352 7.05546052 -13.32894872 -5.57128763 2.93826133

6.66167982 6.23979143 3.21321827 -1.1720244 -6.09228559]

New Parameters: [ 0.651248 1.01537488 -3.41730404 -2.2540715 -0.46432047
0.71946401 1.24102682 1.28302448 1.02543131 0.6002955 ]

Ridge Error on Test Data: 0.0878063649382745

Squared Error on Test Data: 0.028943113127782437

### ======= Ridge Regularisation Worst Curve: Polynomial Degree 1 Lambda 0.25

Polynomial is: [ 0.91609257 -1.85515881]

New Parameters: [ 0.91610343 -1.85515503]

Ridge Error on Test Data: 0.09820617212948918

Squared Error on Test Data: 0.09553064357265667

### ======= Ridge Regularisation Worst Curve: Polynomial Degree 1 Lambda 0.5

Polynomial is: [ 0.91609257 -1.85515881]

New Parameters: [ 0.91612887 -1.85505536]

Ridge Error on Test Data: 0.1008815565517441

Squared Error on Test Data: 0.0955309034512248

### ======= Ridge Regularisation Worst Curve: Polynomial Degree 1 Lambda 0.75

Polynomial is: [ 0.91609257 -1.85515881]

New Parameters: [ 0.91612887 -1.85505536]

Ridge Error on Test Data: 0.1008815565517441

Squared Error on Test Data: 0.0955309034512248

### ======= Ridge Regularisation Worst Curve: Polynomial Degree 1 Lambda 1

Polynomial is: [ 0.91609257 -1.85515881]

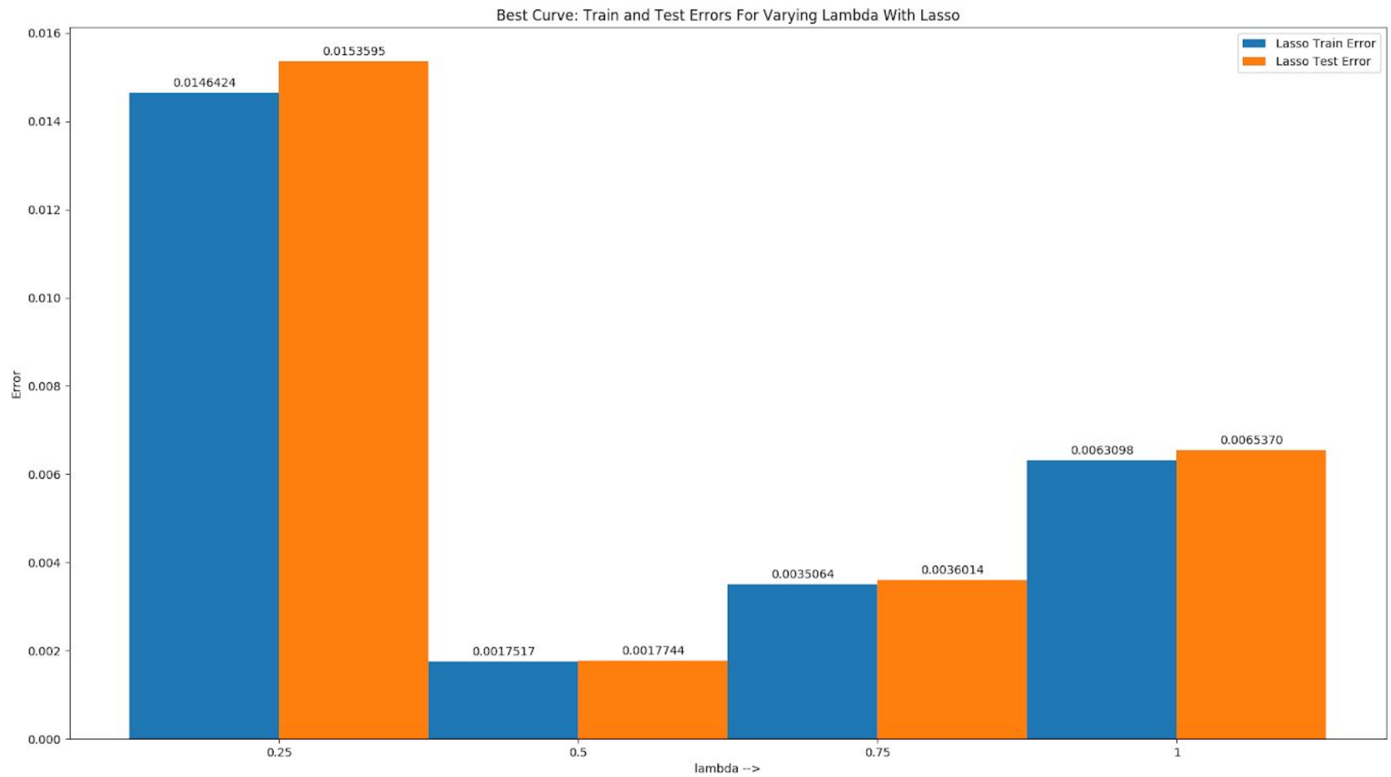
New Parameters: [ 0.91039132 -1.84354454]

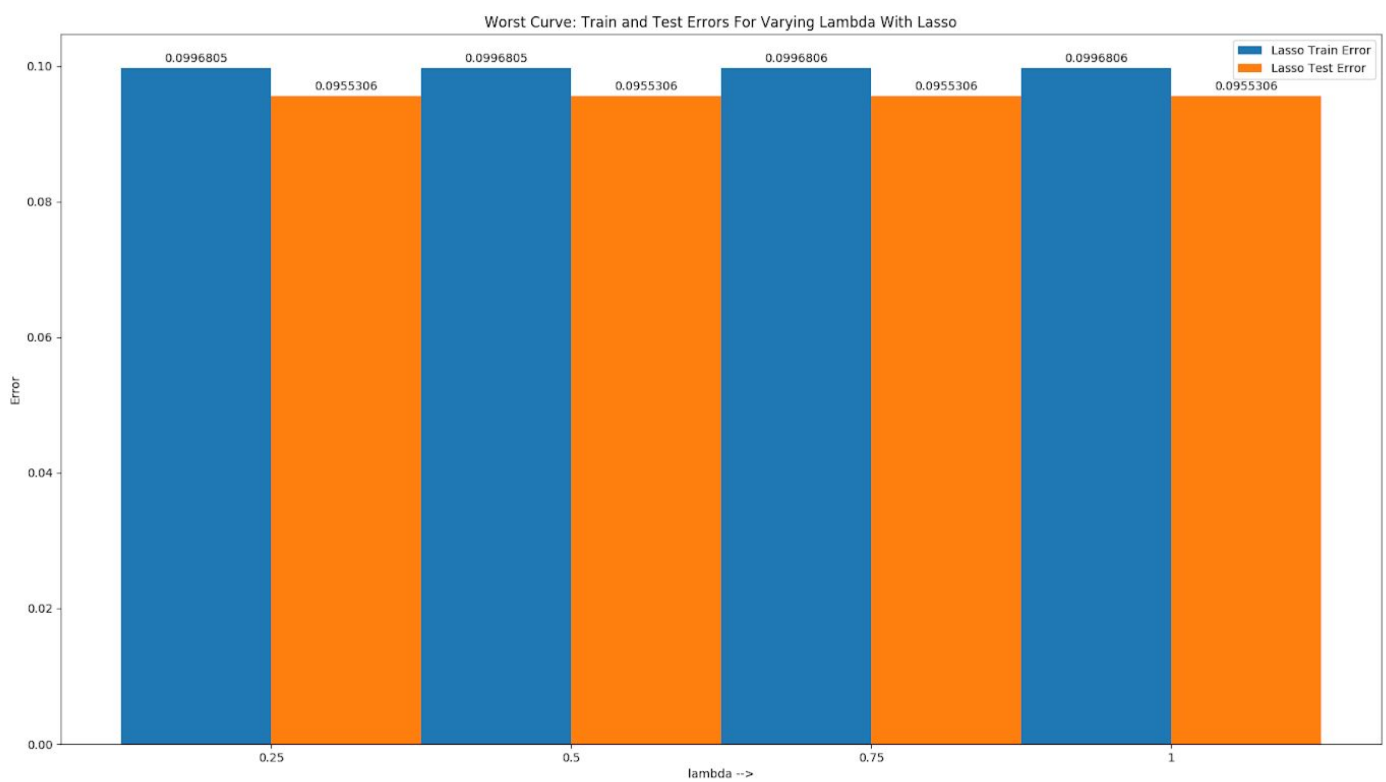
Ridge Error on Test Data: 0.10604735501131582

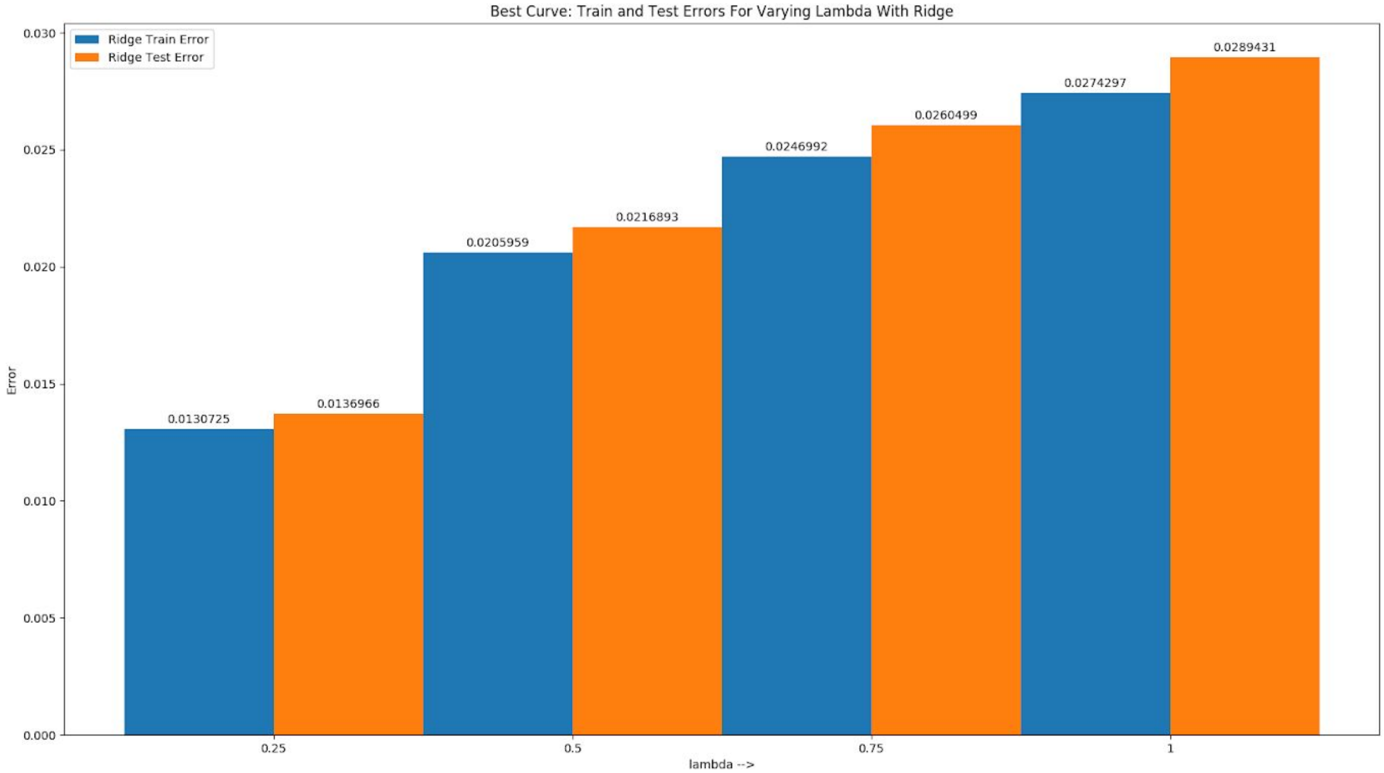
Squared Error on Test Data: 0.09547868297539797

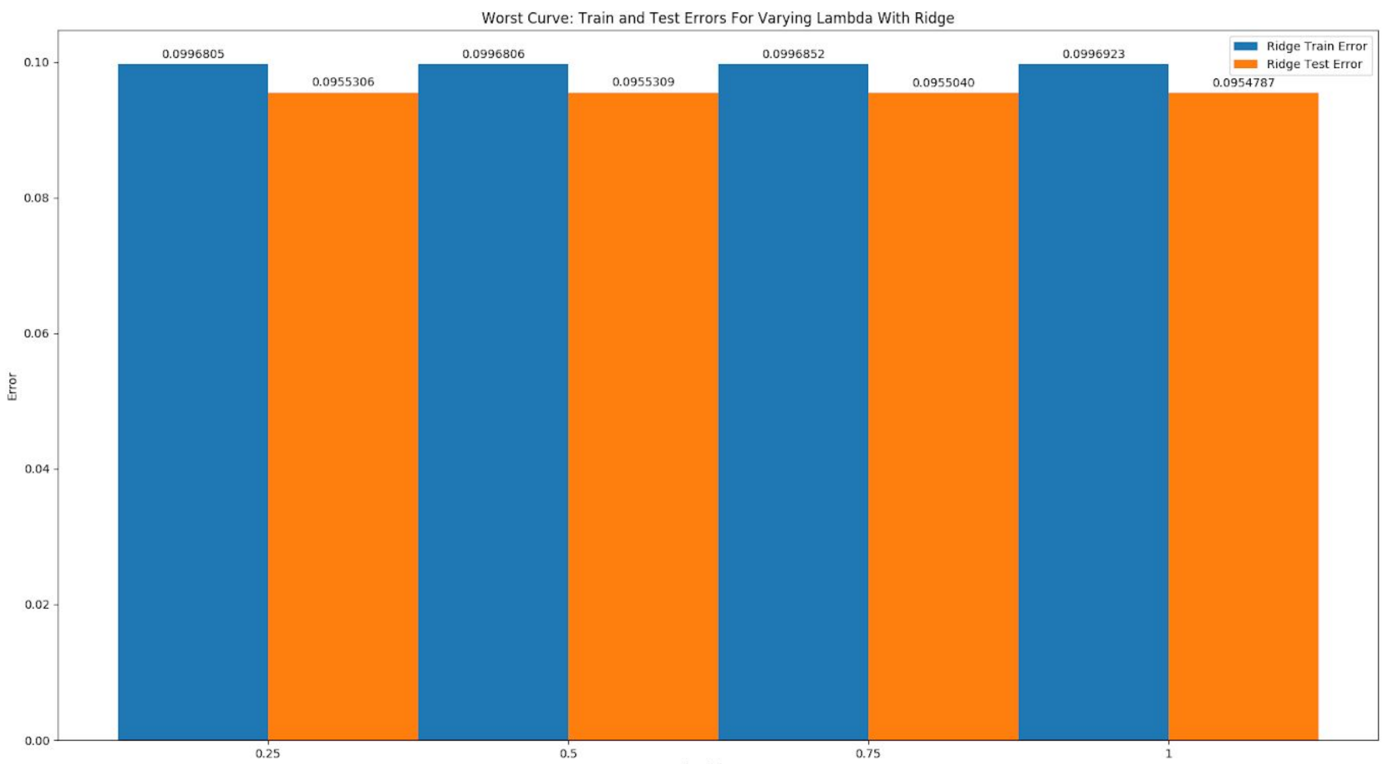
Below are the result graphs for varying lambda with best curve (degree 9) and worst curve (degree 2). The same are also available in the folder “result_plots”. For a better resolution image, please have a look into the folder.

I would prefer to have no regularisation at all since the best Lasso Regularisation with Lambda value 0.5 gives an error which is larger than the error we have received without regularisation. The regularisations are particularly useful in reducing the effect of extra features (with no contribution to actual label) but in the data we have only a single feature which exclusively decides the label at various degrees and hence there is no feature to be eliminated or reduced. Also, there are no particularly very high coefficient values which will be benefited by the process of regularisation either. Hence, no regularisation (as there is no overfitting) is the best option here.


Best Curve: Train and Test Errors For Varying Lambda With Lasso
Lasso Train Error
Lasso Test Error
Error
lambda -->
0.000
0.002
0.004
0.006
0.008
0.010
0.012
0.014
0.016
0.25
0.5
0.75
1
0.0146424
0.0153595
0.0017517
0.0017744
0.0035064
0.0036014
0.0063098
0.0065370


Worst Curve: Train and Test Errors For Varying Lambda With Lasso
Lasso Train Error
Lasso Test Error
Error
lambda -->
0.00
0.02
0.04
0.06
0.08
0.10
0.25
0.5
0.75
1
0.0996805
0.0955306
0.0996805
0.0955306
0.0996806
0.0955306
0.0996806
0.0955306


Best Curve: Train and Test Errors For Varying Lambda With Ridge
Ridge Train Error
Ridge Test Error
0.0130725
0.0136966
0.0205959
0.0216893
0.0246992
0.0260499
0.0274297
0.0289431
Error
lambda -->
0.25
0.5
0.75
1


Worst Curve: Train and Test Errors For Varying Lambda With Ridge
Ridge Train Error
Ridge Test Error
0.0996805
0.0955306
0.0996806
0.0955309
0.0996852
0.0955040
0.0996923
0.0954787
Error
lambda -->
0.25
0.5
0.75
1

## References:

The following sources were used for help during the completion of the assignment.


- https://hackernoon.com/practical-machine-learning-ridge-regression-vs-lasso-a00326371ece
- https://pandas.pydata.org/pandas-docs/stable/10min.html
- https://www.programiz.com/python-programming/matrix
- https://realpython.com/python-matplotlib-guide/
- https://towardsdatascience.com/one-simple-trick-for-speeding-up-your-python-code-with-numpy-1afc846db418
- CS60050 Slide 1, 2 and 4