ECS308/658: Assignment 2 Report

DEEP POOJA
17074

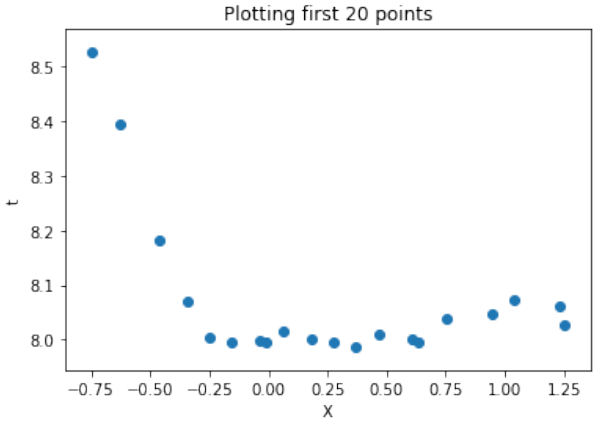
1 Objective

The objective here is to implement the concepts of regression learnt in class via polynomial curve fitting. To recap, polynomial curve fitting is an example of regression. In regression, the objective is to learn a function that maps an input variable x to a continuous target variable t.

2 Part 1A

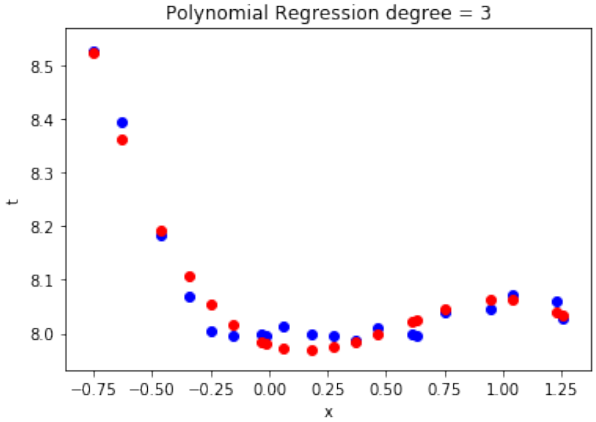
2.1 with first 20 data points only

Visualizing the 20 data points



It looks like data is neither linear nor quadractic. So consider a simple model initially a 3rd degree polynimial.

let's fit a 3rd degree polynomial and see how good it fits the data.



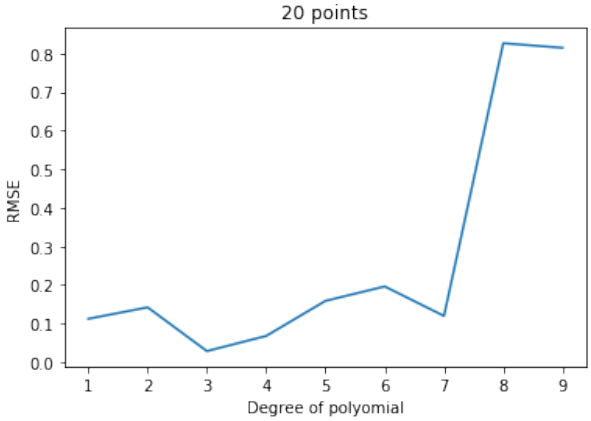
Not bad!, 3 degree polyomial is fittig the data quite nicely

Now we will use cross validation approach to calculate the error in curve fittig using different error formulation like:

MSE-Mean squared error: The mean_squared_error function computes mean square error, a risk metric corresponding to the expected value of the squared (quadratic) error or loss.
RMSE-Root mean squared error
Max error: It calculates the maximum residual error.
Mean absolute error: The mean_absolute_error function computes mean absolute error, a risk metric corresponding to the expected value of the absolute error loss or l1-norm loss
Median absolute error:The median_absolute_error is particularly interesting because it is robust to outliers. The loss is calculated by taking the median of all absolute differences between the target and the prediction.

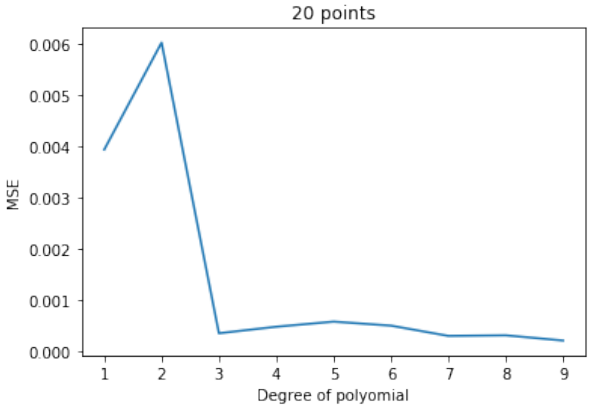
Root-Mean-Squared-Error

Best degree 3 with RMSE 0.028850177500881972



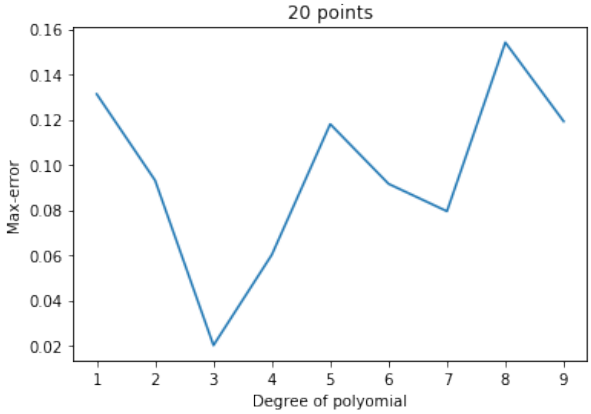
MSE-Mean squared error

Best degree 9 with MSE 0.00021702977066326558.



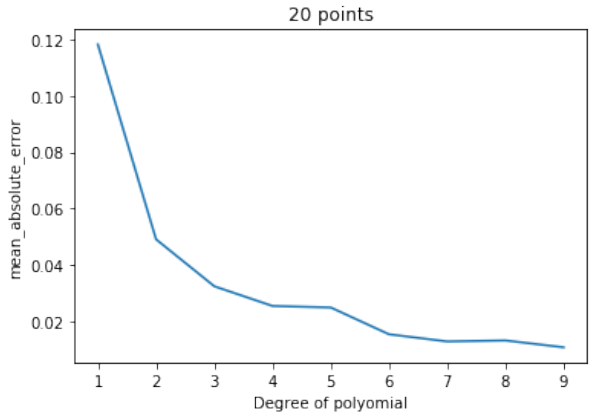
Beyond degree 3, the error value starts increasing a bit, and then gradually decreasing seems it's overfitting the data beyond degree 3 polynomial.
Max-Error

Best degree 3 with Max-error 0.020059805194133418



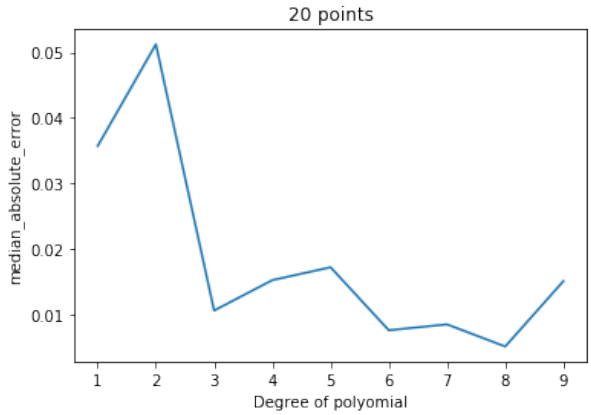
Mean_Absolute_Error

Best degree 9 with mean_absolute_error 0.010722751888103055



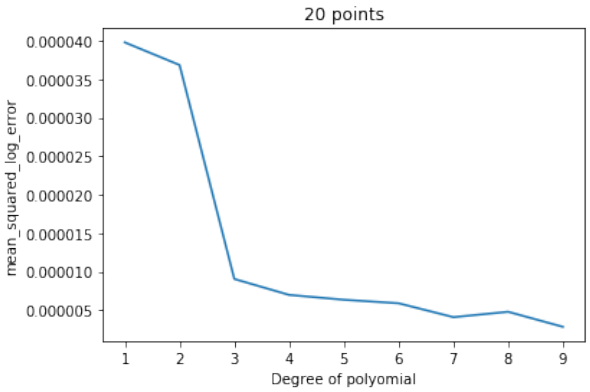
In this case also, it's overfitting the data, as in the error value is gradually decreasing for all degree.
Median_Absolute_Error

Best degree 8 with median_absolute_error 0.005140660733640168



mean_squared_log_error

Best degree 9 with mean_squared_log_error 2.848559859744757e-06



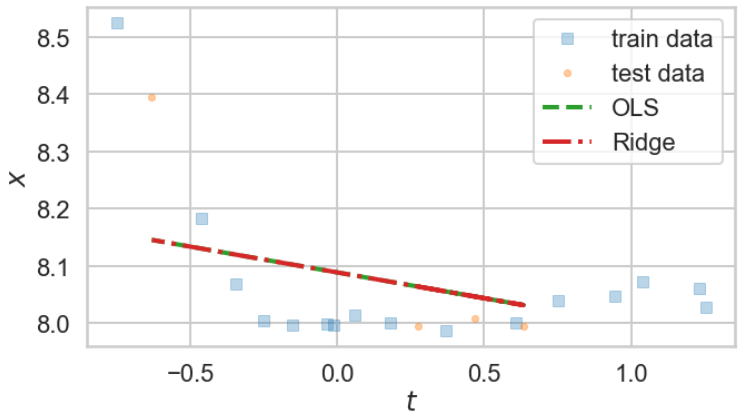
Observation:
In all the above errors formalisation, trend is common that is for degree 3 or less it seems underfitting the data and for higher like 7 or above, its overfitting the data.
In all the above errors formalisation, one thing is common the error is minimum at degree at least locally, so degree three polynomial could be the safe choice with intercept and coefficients 7.979761811089663 [0. -0.1430391 0.54037032 -0.31234552] respectively. The estimate of noise variance is as 0.00043530457100312534.

Introducing Regularization

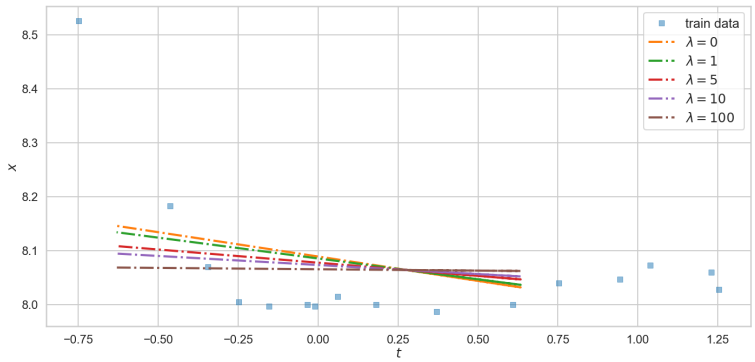
Simple Ridge Regression
OLS stands for ordinary least square.
Here β's are coefficient of ridge regression.

	OLS	Ridge $\lambda = 0$
β_0	8.092766	8.092766
β_1	-0.098813	-0.098813

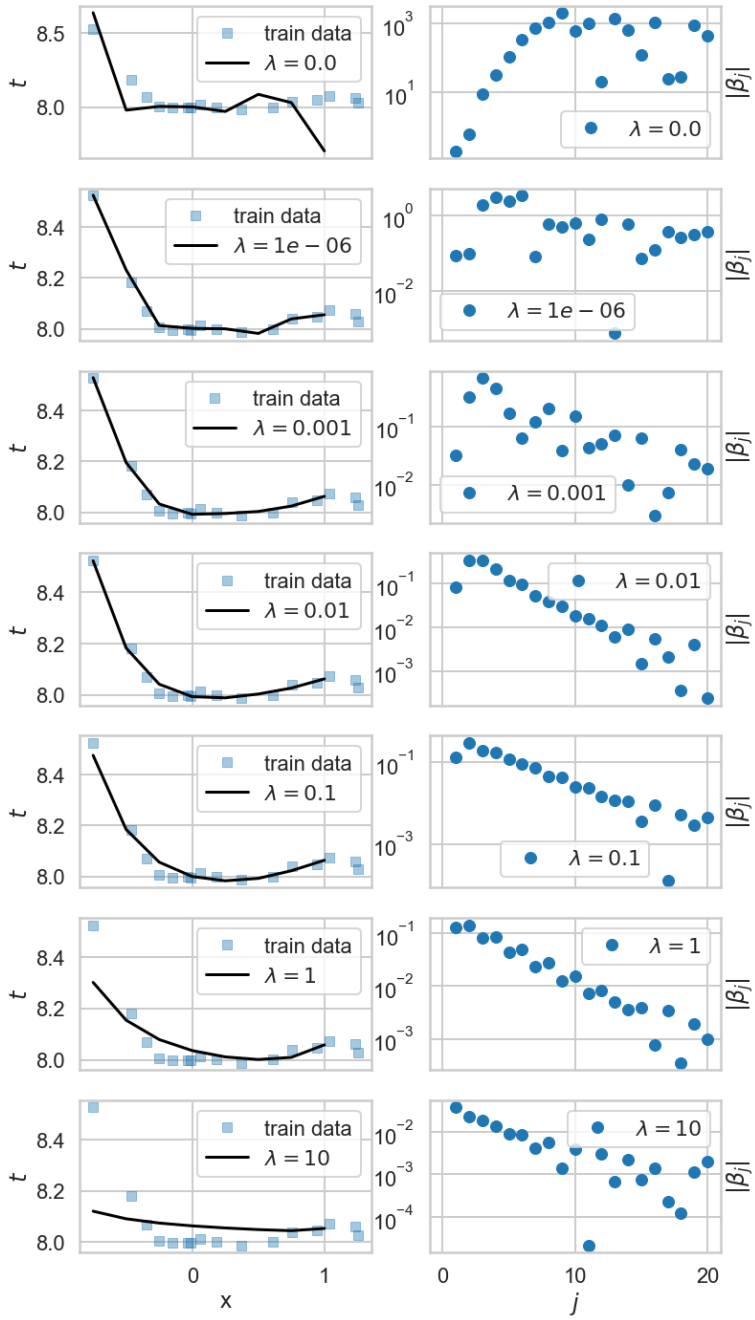
The beta coefficients for linear and ridge regressions coincide for $\lambda = 0$, as expected.



Let, see how the above plot vary with different set of λ values.



Polynomial Ridge Regression



I set the maximum degree of polynomial to be 20. Here β is coefficient of ridge regression for particular value of λ.

As we can see, as we increase λ from 0 to 1, we start out overfitting, then doing well, and then our fits develop a mind of their own irrespective of data, as the penalty term dominates.

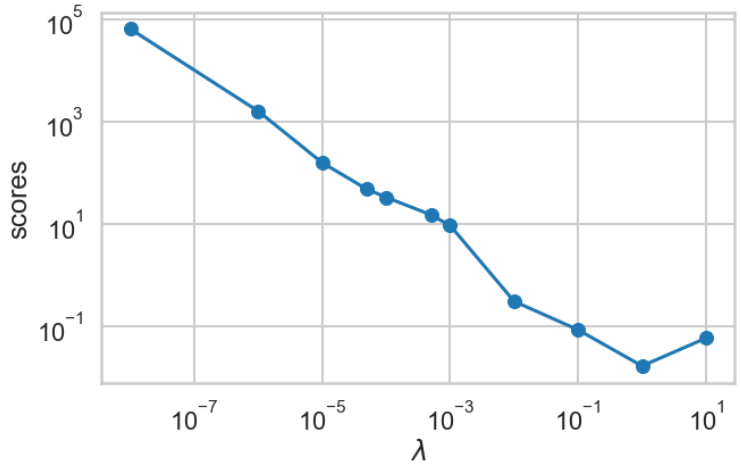
Now we have to fit the model on the training set with 4-fold cross validation. Save the fit and used GreidSearch to obtain best parameter. Which gives the result as follows.

Ridge(alpha=1.0, copy_X=True, fit_intercept=True, max_iter=None, normalize=False, random_state=None, solver='auto', tol=0.001)

'alpha': 1.0 (here alpha is equivalent to λ)

-0.016912509001399983

We also output the mean cross-validation error at differentλ(with a negative sign, as to maximize negative error which is equivalent to minimizing error).
Now we make a log-log plot of -fit_scores versus fit_lambdas.

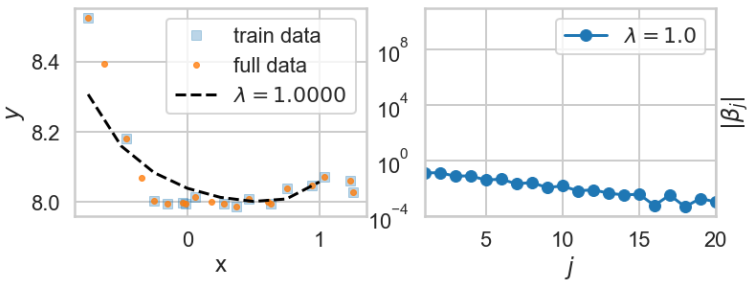


Now we will obtain best model (ridge regression) using k-fold (chosen k=5) cross validation

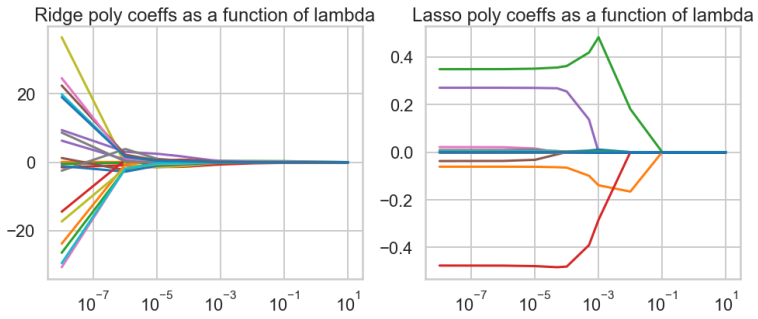
Best model searched: alpha = 0.01 (here alpha is equivalent to λ)
intercept = 7.992929844452785
betas = [0. -0.09033538 0.35586754 -0.34210477 0.1904485 -0.1191102 0.08623301 -0.04634857 0.04124388 -0.02261591 0.0209722 -0.01179648 0.01197766 -0.00564209 0.00724989 -0.00317925 0.00305193 -0.00374588 -0.00070137 -0.00364334 0.00143543],

At this point, we have determined the best penalization parameter for the ridge regression on our current dataset(20 points)using cross validation. Let's refit the estimator on the training set and calculate and plot the test set error and the polynomial coefficients. Noticing how many of these coefficients have been pushed to lower values or 0.
Polynomial coefficients

array([0. , -0.13524309, 0.14135547, -0.07903737, 0.08358823, -0.04176496, 0.0480798 , -0.02214734, 0.02766661, -0.01196167, 0.01563132, -0.00693129, 0.00823886, -0.00473823, 0.00352224, -0.00395929, 0.00061556, -0.00346787, -0.00049245, -0.0019726 , 0.00110023])



We have also obtain polynomial coefficients with Lasso regression and plotted their coefficient with varying parameter λ.

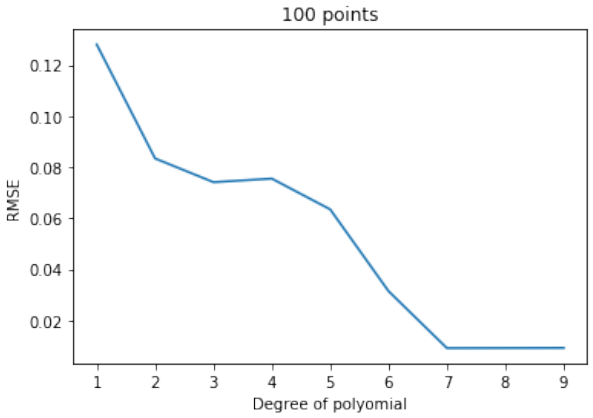


Presumably the orange line in the Lasso plot is for the linear term (it becomes important as the other terms "drop out").

2.2 with all 100 points

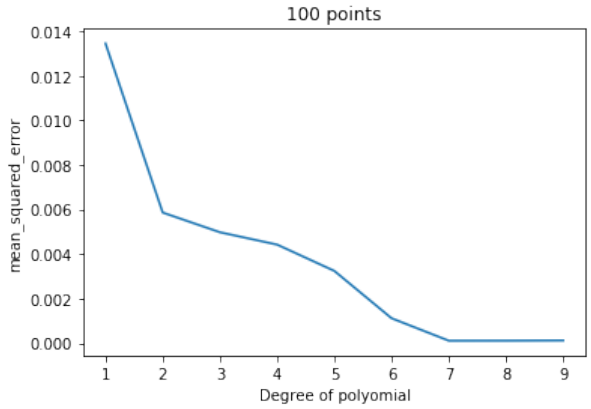
Root-Mean-Squared-Error

Best degree 7 Root mean_squared_error 0.011435097525081995



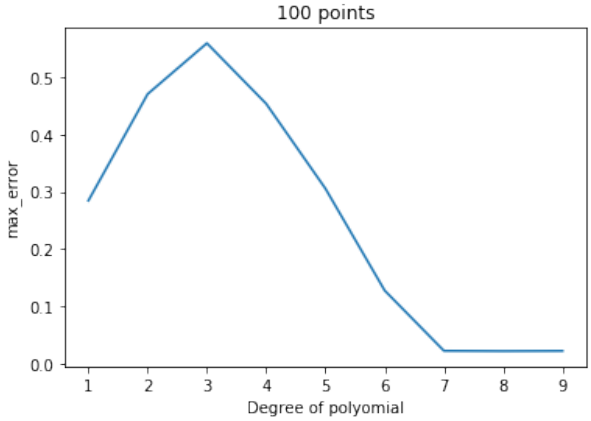
MSE-Mean squared error

Best degree 7 with mean_squared_error 0.00015900692434743684



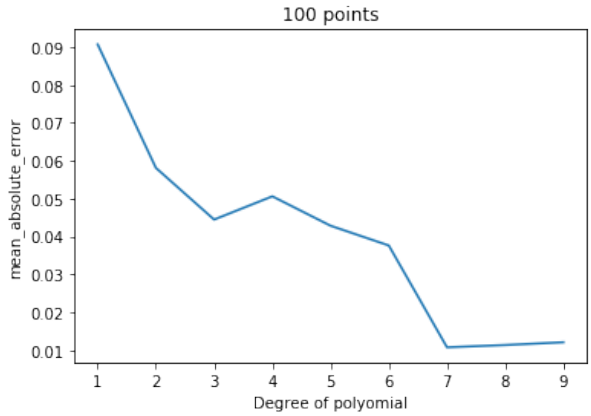
Max-Error

Best degree 7 max_error 0.016275272833910392



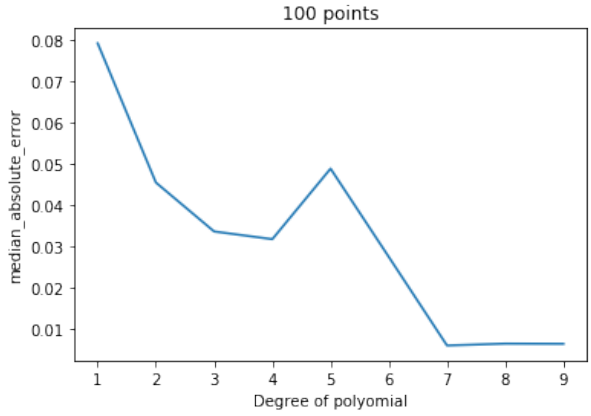
Mean_Absolute_Error

Best degree 8 mean_absolute_error 0.010428755212909984



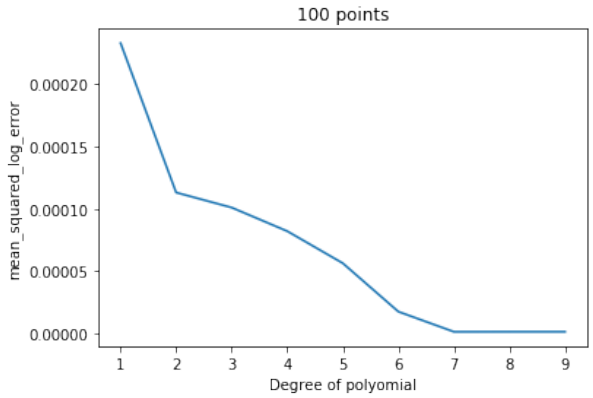
Median_Absolute_Error

Best degree 7 median_absolute_error 0.006210794447810031



mean_squared_log_error

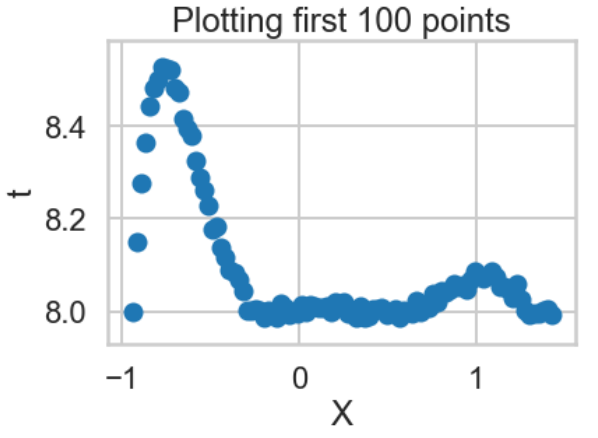
Best degree 8 with mean_squared_log_error 1.922518593702872e-06

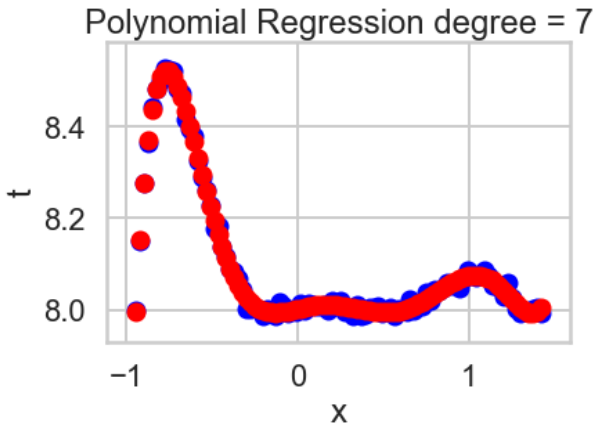


Observation:
In all the above errors formalisation, trend is common that is for degree 3 or less it seems underfitting the data and for higher like 8 or above, its overfitting the data.

In all the above errors formalisation, one thing is common the error that is degree 7 is best fitting the data with 100 points in contrast to 20 points we had safely assumed degree 3 as a best fit. Also with 20 points, polynomial with degree 7 was a choice because error attains minima there in many of the above error formalisation but we have avoided it to avoid the chances of overfitting the data.

so polynomial with degree 7 is best fitting the data with intercept and coefficients 8.002144160116925 [0. 0.08578064 -0.01674391 -1.4437579 2.32327251 0.68517352 -2.56216065 1.00083176] respectively. The estimate of noise variance is as $9.743145986743975e^{-05}$.





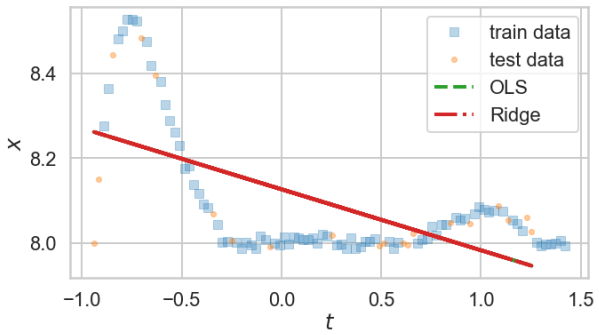
Introducing Regularisation

Simple Ridge regression

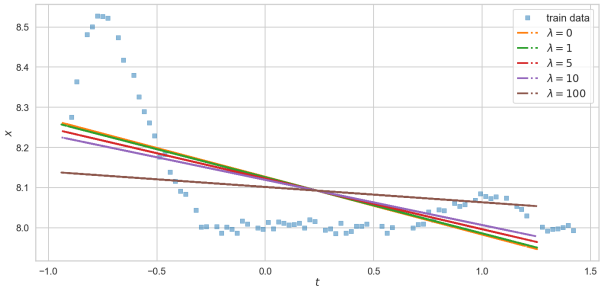
Here β's are coefficient of ridge regression.

	OLS	Ridge $\lambda = 0$
β_0	8.126637	8.088985
β_1	-0.143642	-0.090137

Note: The coefficients are different than what we have obtained fro 20 points.

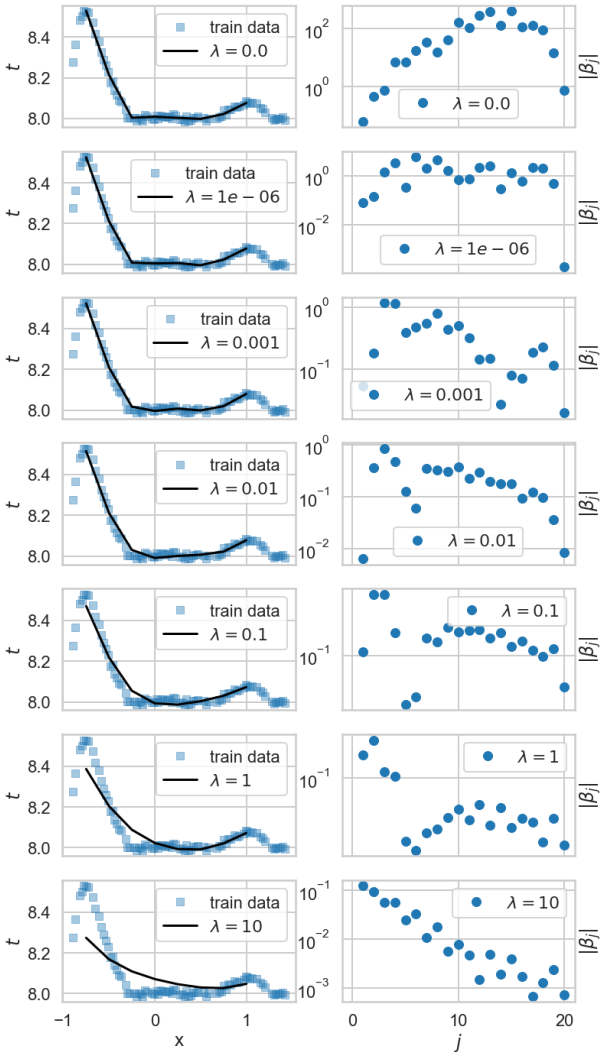


Varying the parameter λ



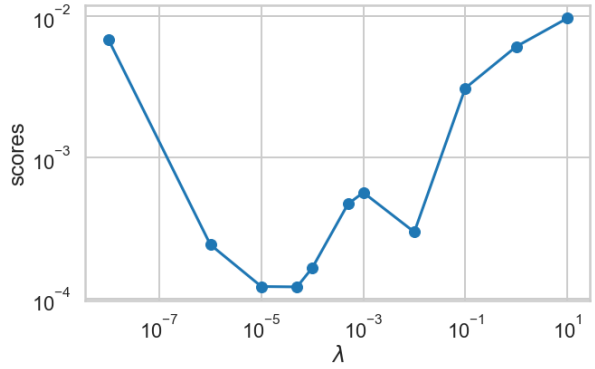
Polynomial Ridge Regression

Similar to above analysis we set the maximum degree to 20.
As we can see, as we increase $\lambda$from 0 to 1, we start out overfitting(see the coefficients of λ=0) its very large, then doing well, and then our fits develop a mind of their own irrespective of data, as the penalty term dominate.



Similar to what we did above, we now fit the model on the training set with 4-fold cross validation.

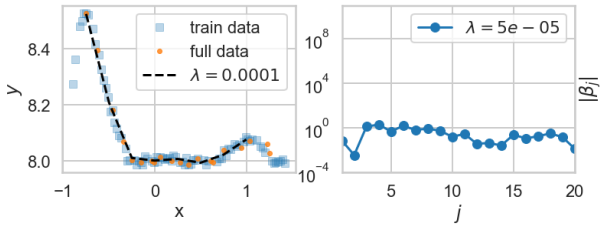
Ridge(alpha=5e-05, copy_X=True, fit_intercept=True, max_iter=None, normalize=False, random_state=None, solver='auto', tol=0.001)
'alpha': 5e-05
score = -0.0001216849807368154 Note: Here we are getting smaller value of λ cum alpha. Also score is less negative. We also output the mean cross-validation error at different λ and Now make a log-log plot of -fit_scores versus fit_lambdas.



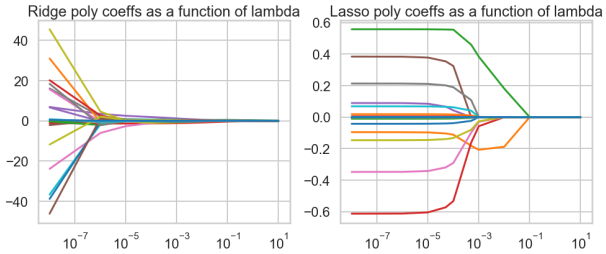
Now we have obtained best ridge model using K-fold validation, we chose k=5.

Best model searched:
alpha = 0.0001
intercept = 8.001101834731967
betas = [0.00000000e+00 7.30338702e-02 3.31495627e-02 -1.32191955e+00 1.85364840e+00 4.86882213e-01 -1.23658273e+00

6.33082488e-01 -1.00697195e+00 5.57064823e-01 -3.14395866e-01 3.63506552e-01 4.51250696e-02 2.99079649e-02 8.16275339e-02 -2.82767522e-01 1.10408079e-01 -2.81929292e-01 3.81263201e-01 -1.27033073e-01 -1.00779617e-03],
similar observation is that some have been pushed to lower values or 0.
coefficients
array([0. , 0.0749809 , 0.00373461, -1.3364602 , 2.03391264, 0.49763223, -1.57627368, 0.67099783, -0.88426336, 0.56955762, -0.1655106 , 0.29719179, 0.03849912, -0.04536453, 0.02712604, -0.25234073, 0.11652172, -0.1839438 , 0.32808903, -0.15247949, 0.01355259])



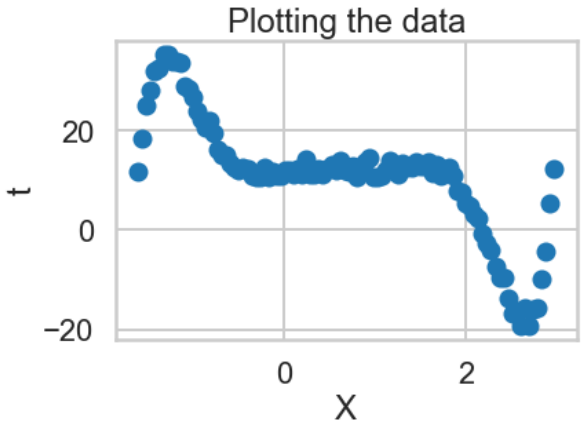
Similar analysis we also did for lasso and generated the below plot for comparison.



Observation: With 20 points, we safely assumed that degree 3 polynomial was fitting the data nicely. When we did analysis for all points, we came to know that is the degree 7 which actually fits nicely.

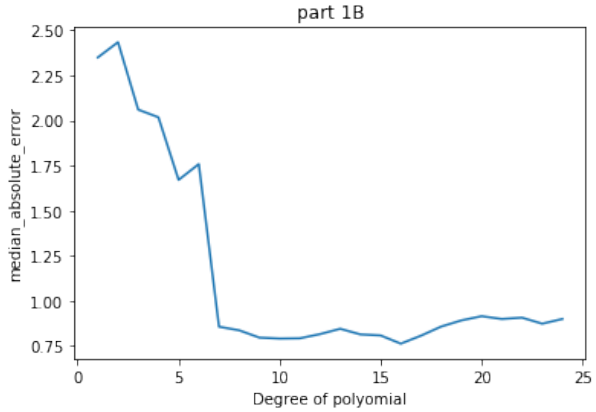
3 Part 1B

visualizing the dataset



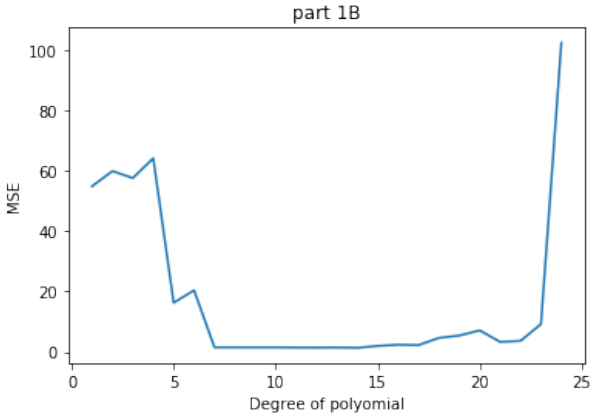
Different error formalisations
Root-Mean-Squared-Error

Best degree 15 RMSE 1.0988167997306735



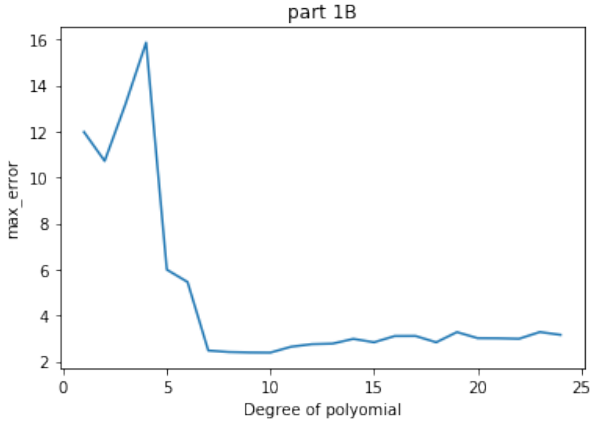
MSE-Mean squared error

Best degree 14 MSE 1.3393979363214055



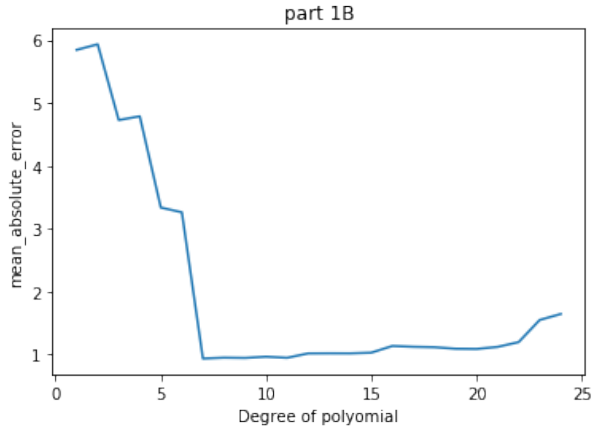
Max-Error

Best degree 10 max_error 2.389283592133385



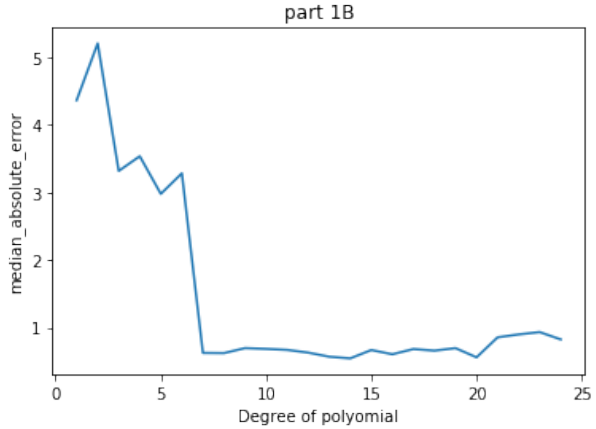
Mean_Absolute_Error

Best degree 7 mean_absolute_error 0.9361986123796445

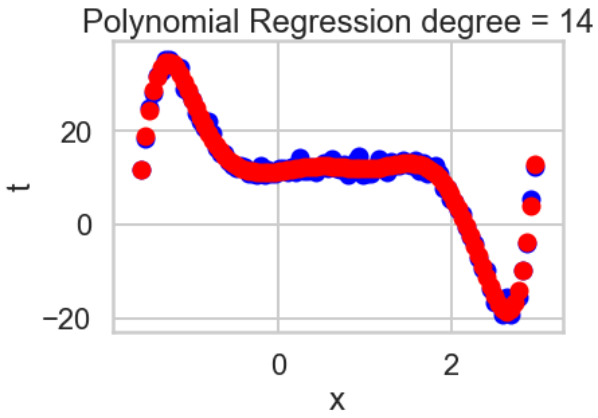


Median_Absolute_Error

Best degree 14 median_absolute_error 0.5461025570189975



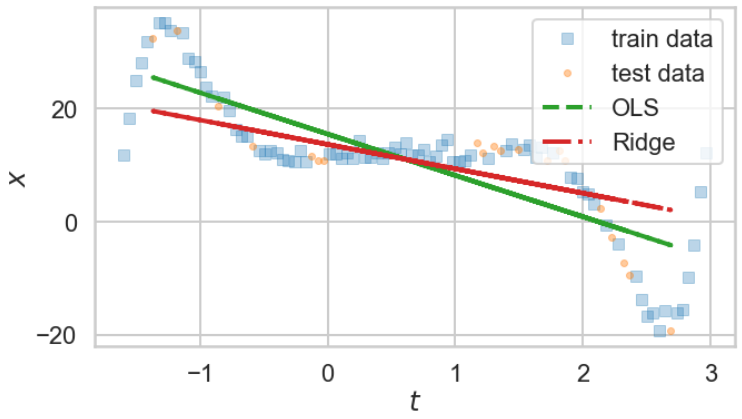
Observation:
Mean_squared_log_error cannot be used because of presence of negative value in data.
On the line of above results,it seems polynomial with degree 14 might be a good fit for this dataset. Let's see.

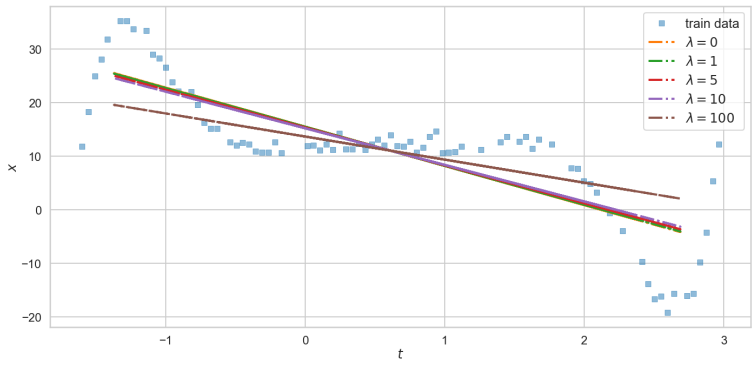


Here we have determined the coefficients and intercept of 14 degree polynomial as follows.
intercept = 11.34323953927949
coefficient= [0. 2.36410495 4.54764445 -9.61340484 -0.77984886 -3.0562964 11.20954851 2.10060978 -9.1739635 1.53298556 2.29842119 -0.7980737 -0.15314098 0.10401358 -0.01265165]
Noise variance = 0.9350343269472999
we see a high value of noise in this case.

Introducing regularisation
Simple ridge regularisation

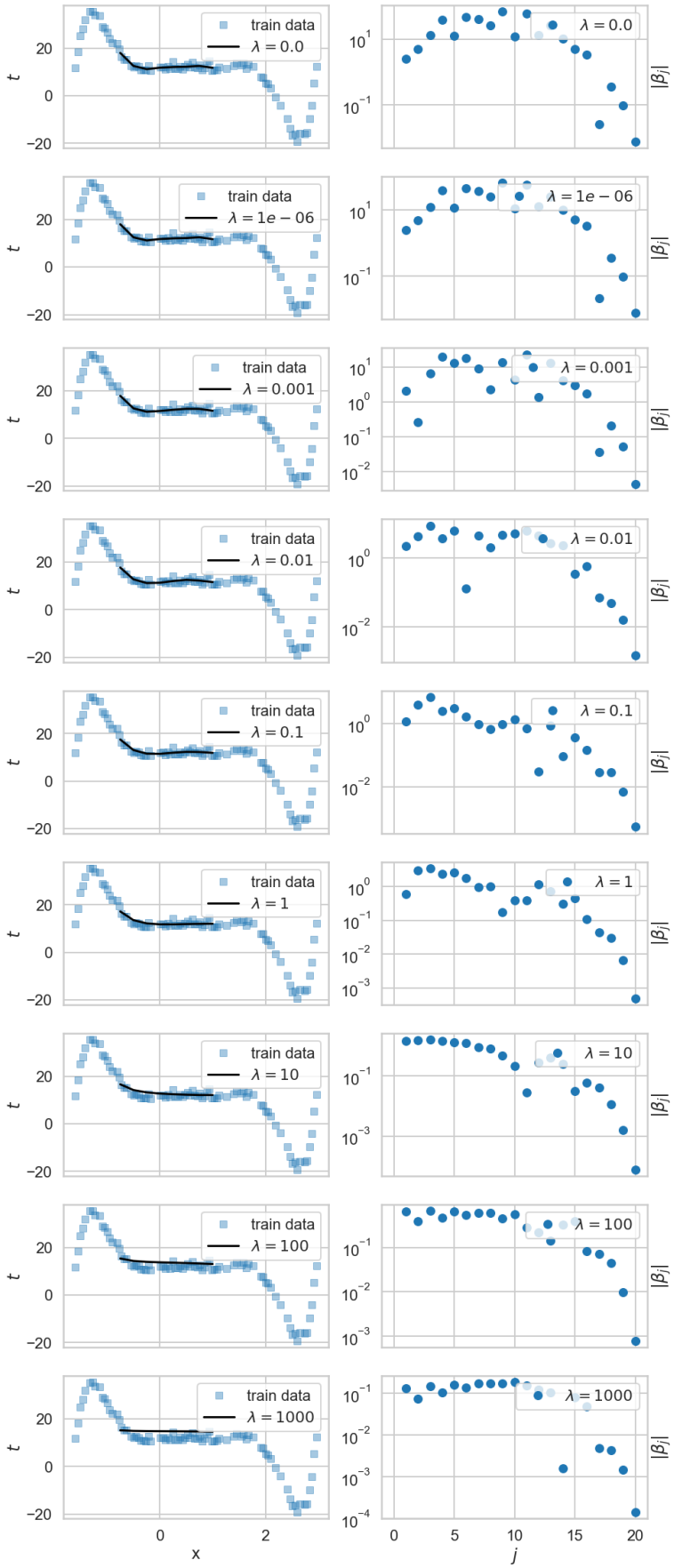
	OLS	Ridge $\lambda = 0$
β_0	15.519864	15.519864
β_1	-7.310003	-7.310003





Polynomial ridge regression

fixing the maximum degree of polynomial to be 25.

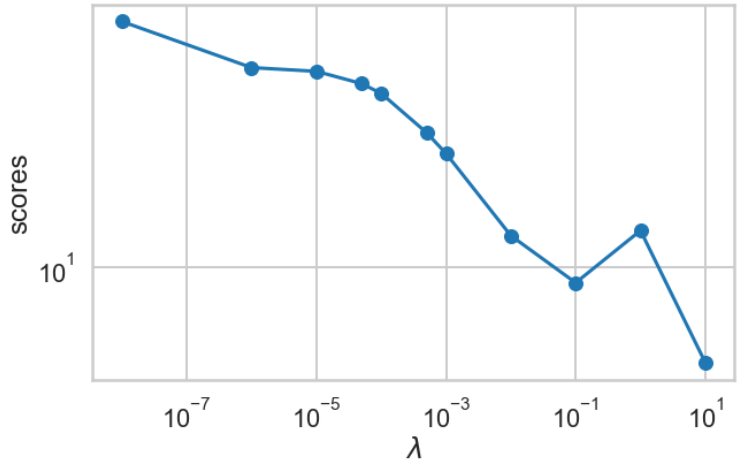


Note: As λ is varied from 0 to 1000, the model starts out overfitting, then doing well, and then our fits develop a mind of their own irrespective of data, as the penalty term dominates.

Using GridSearch and 4-fold cross validation we obtain our best model on this dataset as follows.

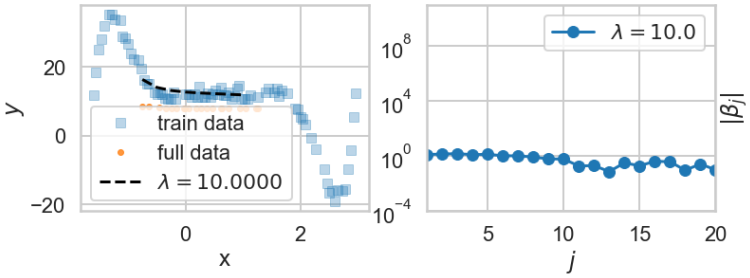
Ridge(alpha=10.0, copy_X=True, fit_intercept=True, max_iter=None, normalize=False, random_state=None, solver='auto', tol=0.001)
'alpha': 10.0
-4.856840310109792 Noticing in this the regularisation parameter λ is high as compared to previous cases.

Now we make a log-log plot of -fit_scores versus fit_lambdas.

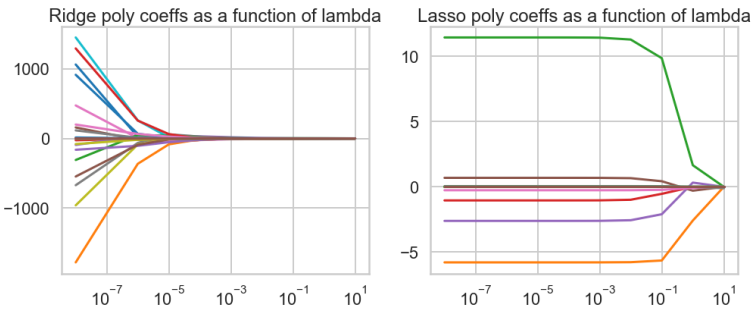


Our regularised model for this dataset
Best model searched: alpha = 10.0
intercept = 12.697290434087275
betas = [-2.77445719e-06 -1.32564283e+00 1.40548381e+00 -1.53249570e+00 1.27501780e+00 -1.36349544e+00 1.06263936e+00 -1.05267424e+00 8.61779666e-01 -6.28783771e-01 6.15767451e-01 -1.92467437e-01 2.05909989e-01 7.52580346e-02 -3.26328057e-01 1.97960759e-01 -4.23957090e-01 3.92184962e-01 9.43182304e-02 -2.70076055e-01 1.00484914e-01 1.96125749e-02 -2.44781218e-02 7.34614946e-03 -9.86240604e-04 5.07946755e-05],
with coefficients as follows

array([-2.77445719e-06, -1.32564283e+00, 1.40548381e+00, -1.53249570e+00, 1.27501780e+00, -1.36349544e+00, 1.06263936e+00, -1.05267424e+00, 8.61779666e-01, -6.28783771e-01, 6.15767451e-01, -1.92467437e-01, 2.05909989e-01, 7.52580346e-02, -3.26328057e-01, 1.97960759e-01, -4.23957090e-01, 3.92184962e-01, 9.43182304e-02, -2.70076055e-01, 1.00484914e-01, 1.96125749e-02, -2.44781218e-02, 7.34614946e-03, -9.86240604e-04, 5.07946755e-05]

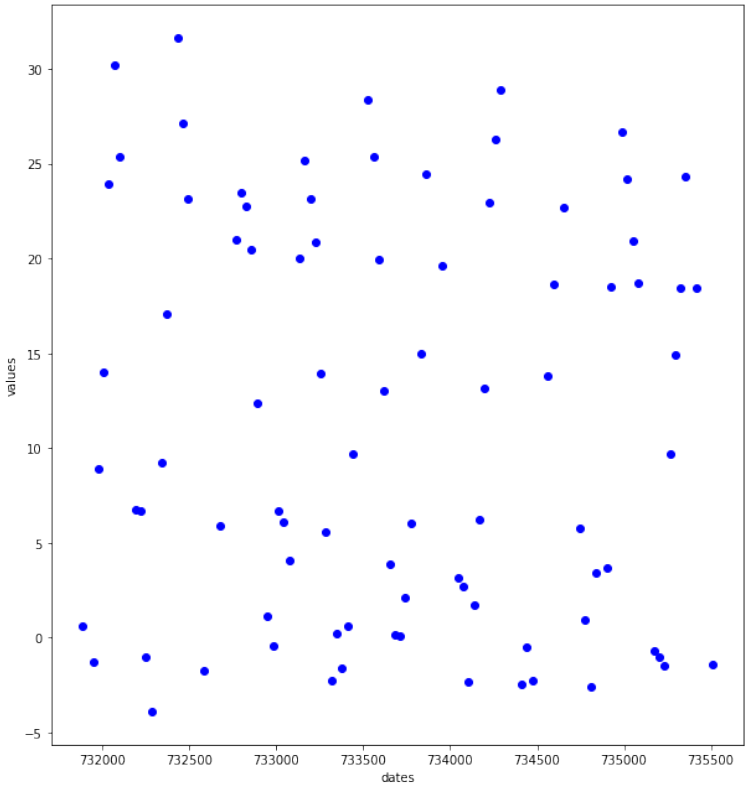


And here is the final plot of Ridge and Lasso's coefficients

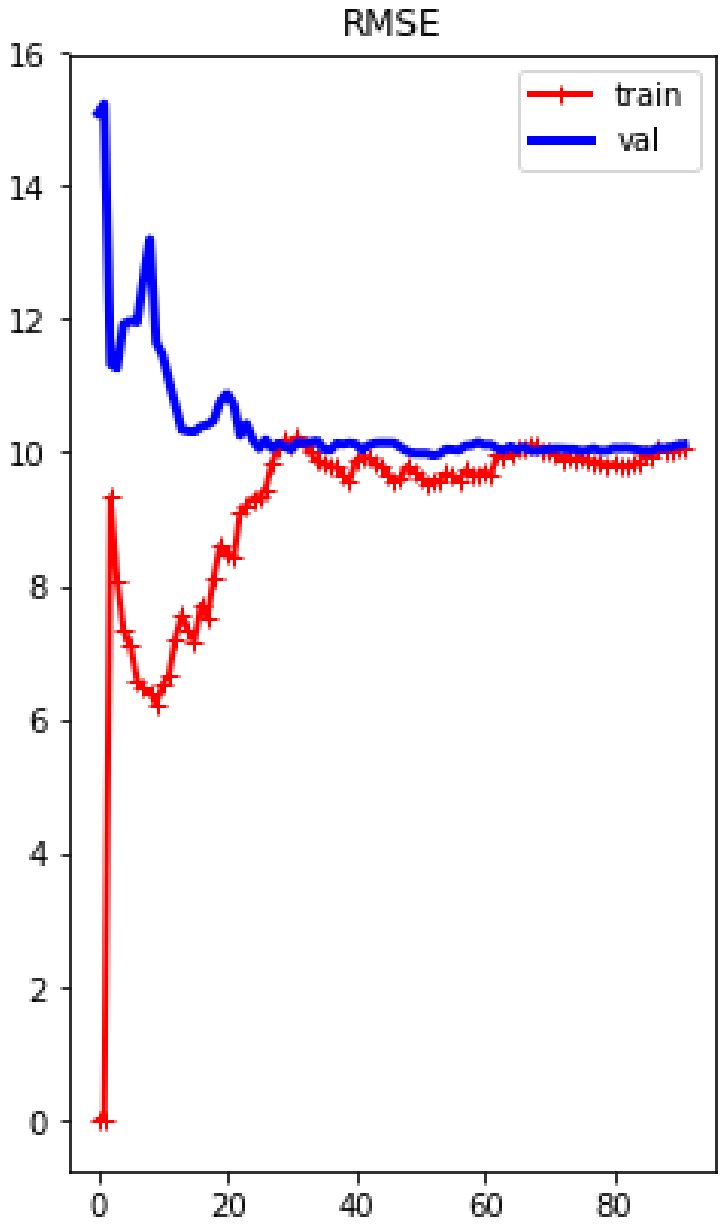


4 Part 2

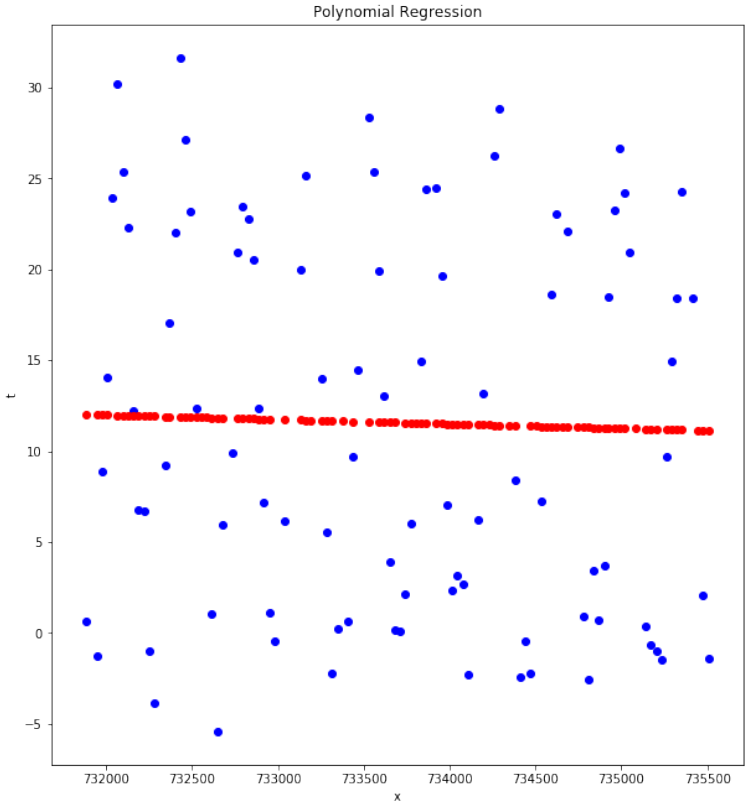
Let's first visualize the data



The data seem randomly distributed.It's very unlikely linear.
Fitting the data using error minimisation. we have used RMSE as error function and cross validation approach.



The 10 values we obtain on test data is as follows:
[[11.52067152] [11.61802371] [11.2202353] [11.91032671] [11.81272807] [11.31783394] [11.15295139] [11.71562235] [12.00792535] [11.41567905]]

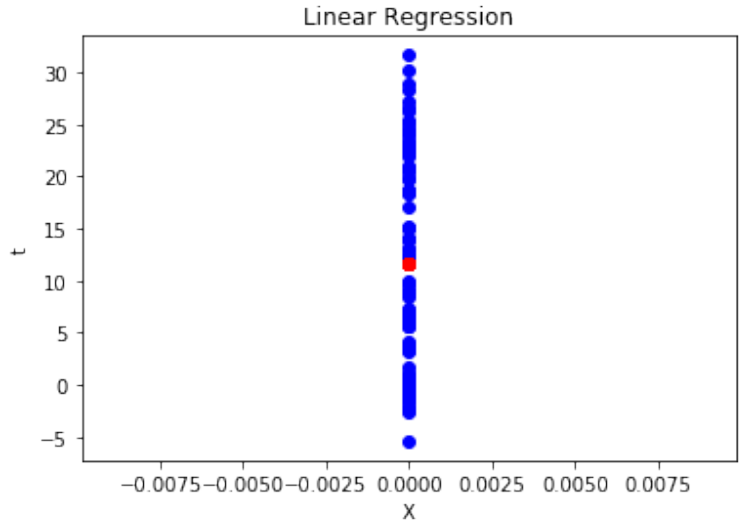


we got out error as: Mean absolute error: 7.30
Residual sum of squares (MSE): 72.72
R2-score: -172.83

Now we have transformed the data using Gaussian basis function
We got coefficient, intercept and score as:

[[0.]] [11.63375161]

score = 0.0
let's plot this fitted data



we got values for test data from Gaussian transformation as follows:

array([[11.63375161], [11.63375161], [11.63375161], [11.63375161], [11.63375161], [11.63375161], [11.63375161], [11.63375161], [11.63375161], [11.63375161]]) Observation:

Gaussian transformation is giving more accurate results.
.

5 References

http://krasserm.github.io/2019/02/23/bayesian-linear-regression/
https://harvard-iacs.github.io/2018-CS109A/labs/lab-5/solutions/