Tejas G Mahale

Regression and Gradient Descent

IST 597 Foundation of Deep Learning

Problem 1: Linear Regression

a) Curve Fitting

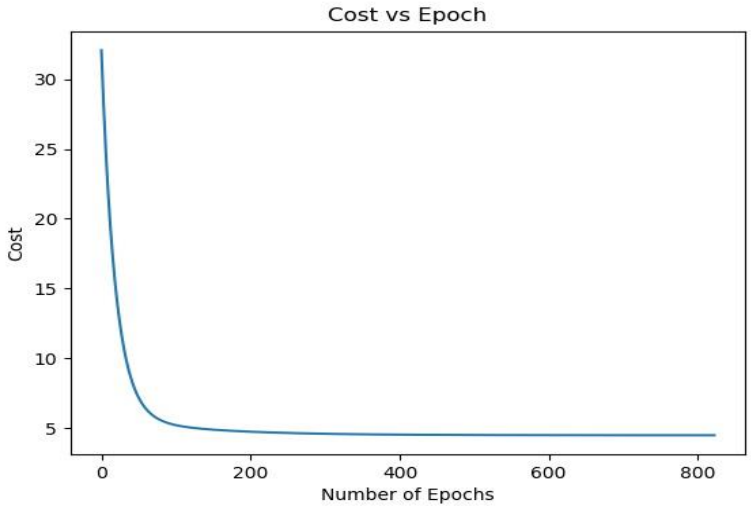


Fig 1.1 Cost vs Number of Epochs

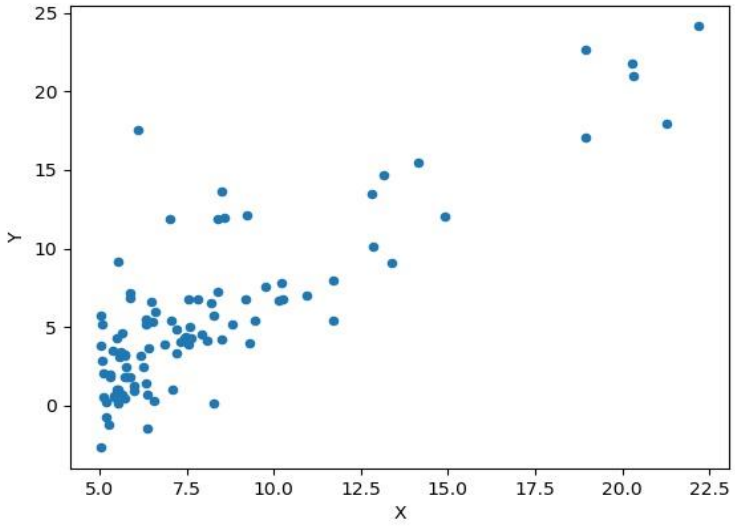


Fig 1.2 Scatter plot of Data

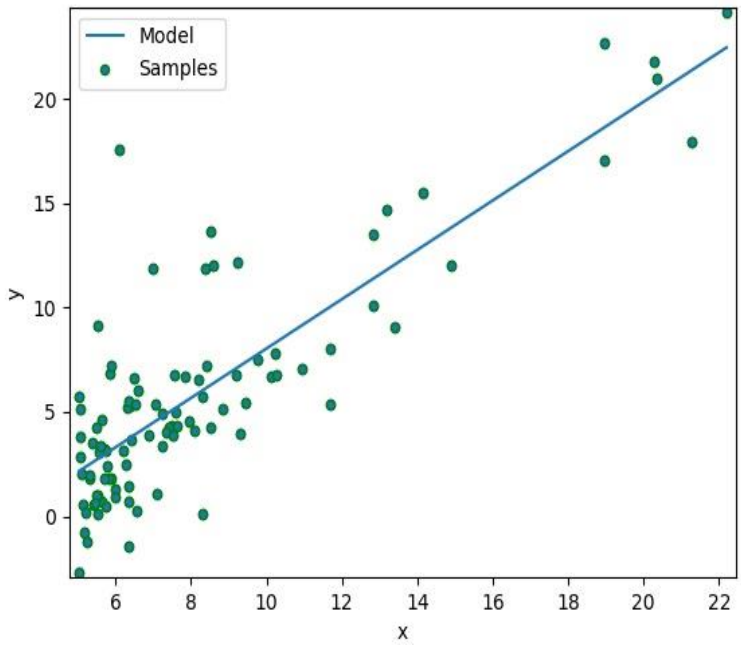


Fig 1.3 Linear Model

b) Parameters:

Model Parameters:

W = 1.1817685

b = -3.7836465

Hyper parameters:

Learning rate (α)= 0.024

Number of Epochs = 822 with convergence criteria of 0.00001

Final Loss = 4.47812608

c) Observation:

- Started with learning rate as 0.0001 but optimization was slow and at every iteration it was reducing loss in small amount.
- Increased learning rate at factor of 10. Increase in learning rate improved rate of optimization and reduced number of epochs required epoch to minimum cost function.

- At certain value of learning rate (0.025), loss started increasing which implies that at this learning rate, gradient descent algorithm will miss the global minima and will diverge.
- After 400 epochs cost epoch, change in curve was not noticeable. So effective number of epoch can be taken as 400.

Learning Rate	Final Loss	Number of Epoch
0.001	4.9826997	2900
0.01	4.4833882	1500
0.1	No Convergence	NA
0.025	No Convergence	NA
0.024	4.47812608	822

Table 1.1

From table 1.1 it is clear that if you increase learning rate, number of epochs require for convergence reduces. From case of learning rate of 0.025 and 0.1 it has been observed that too high learning rate may increases loss rather than reducing it using gradient descent.

If we apply lower limit of eps (convergence criteria) cost may reduce further for higher number of epochs. But as stated above, change in cost function after 400 epochs were negligible and it will not affect model parameter values by far margin.

Problem 2: Polynomial Regression

Parameters:

Hyper parameter	Value
Learning Rate (α)	1.5
Number of Epoch	1130
Regularization Parameter (β)	0.01
Convergence Criteria (eps)	0.00001
Degree of Polynomial	15
Final Loss	0.05681306

Table 2

Advantage in higher order Polynomial curve fitting:

Taking high degree polynomial for least square curve fitting increases capacity of model and make it more flexible. Lot of time if we observe scatter plot of points, we can predict that our linear model is not best fit for given dataset. At such point polynomial curve fitting could be useful.

Problem with Polynomial curve fitting:

One issue with higher order polynomial curve fitting is over-fitting. In over-fitting, curve tries to pass through each and every point in dataset as degree increases. Higher the degree of polynomial, lower the least square error. Such model performs extremely well on points in dataset but this model can fail to estimate on new sample points.

Regularization:

There are multiple ways to tackle over-fitting problem. But one common way to avoid over-fitting in regression is regularization. Higher order polynomial has a greater number of weight parameters. Range of these weight parameters can be erratic which adds more flexibility to curve. In regularization we add another term in least square error loss to penalize weight parameters. This penalty can be controlled using hyperparameter which can be manually adjusted. As regularize parameter value increases, weights tend to be more distributed to avoid over-fitting. Certainly, regularization increases least square error loss in model but it makes model more viable to new data points.

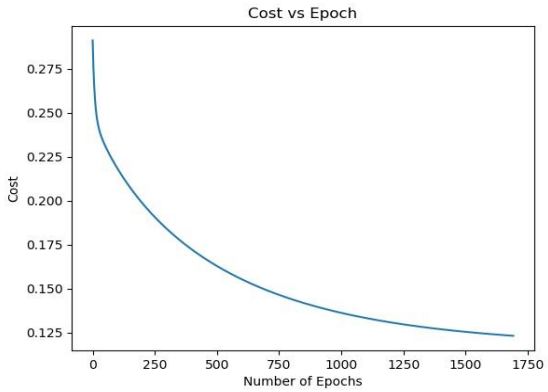
Observations:

a) Finding Learning rate (α) value: Keeping Degree = 15 and Beta = 0.1

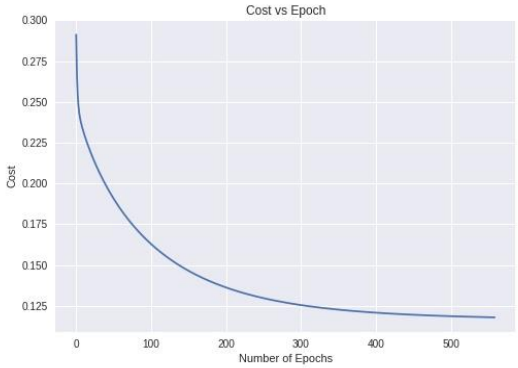
Learning Rate	Final Loss	Number of Epoch
0.001	Too Slow	NA
0.001	Too Slow	NA
0.1	0.12318255	1691
0.5	0.11788995	557
1	0.1170497	337
1.5	0.1167105	252
2	No convergence	NA

Table 2.1

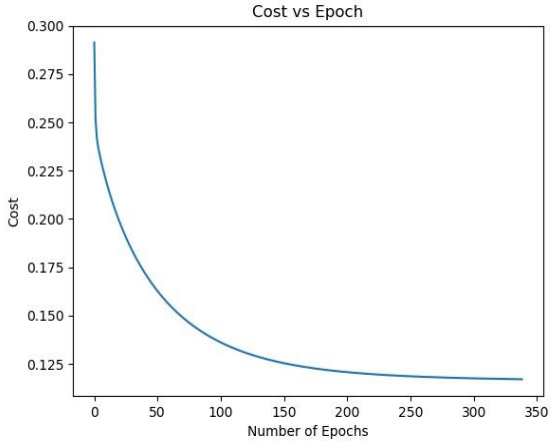
Alpha = 0.1



Alpha =0.5



Alpha = 1



Alpha = 1.5

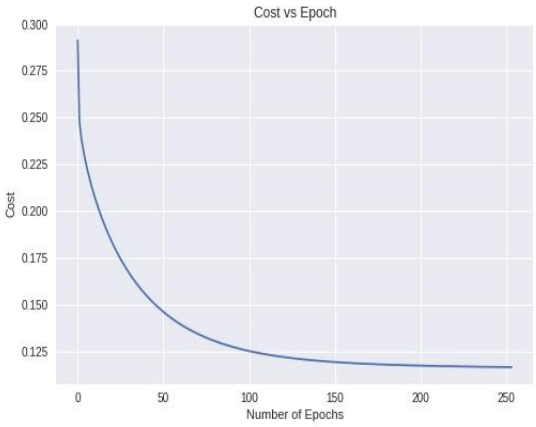


Fig 2.1

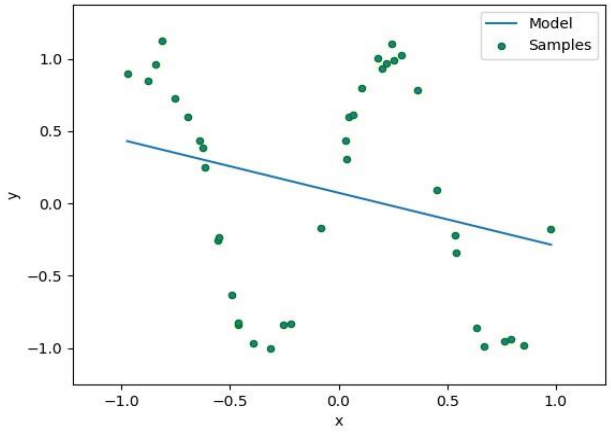
Here learning rate 1.5 is best suitable for model with degree 15. Loss is still high because beat is 0.1 which is pretty high and it is penalizing weight parameters. Lower learning rates like 0.001 are taking so many epochs as well as time for convergence.

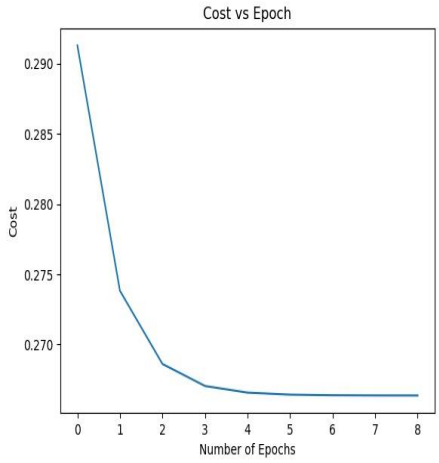
b) Effect of Degree of Polynomial keeping $\alpha = 1.5$ and Beta = 0.001

Degree	Loss	Number of Epochs
Degree 1	0.266361102	7
Degree 3	0.165399098	131
Degree 7	0.036925413	942
Degree 11	0.038787558	1614
Degree 15	0.027208817	2081

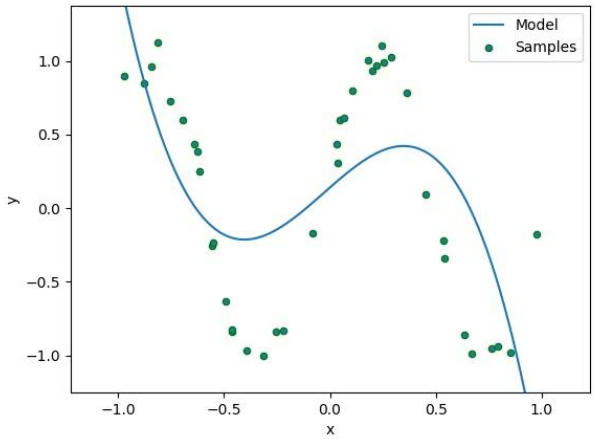
Table 2.2

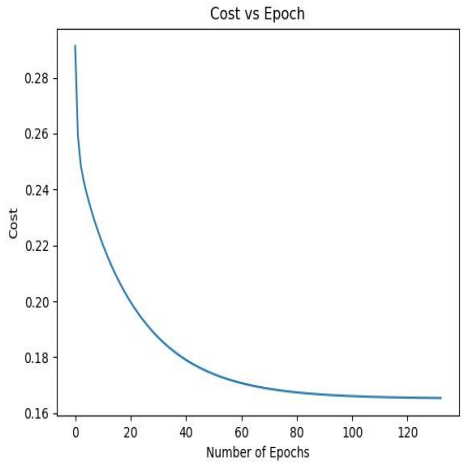
Degree = 1



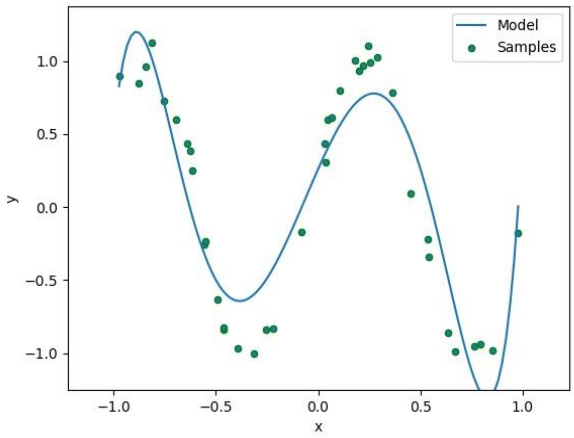


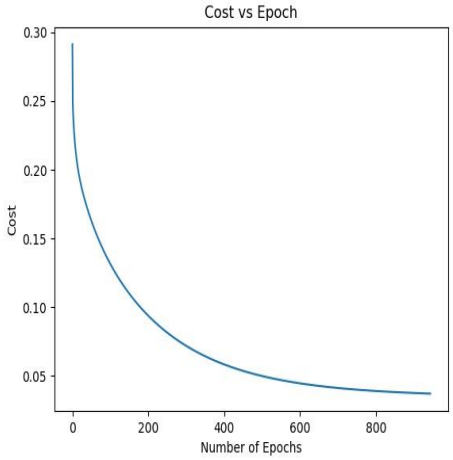
Degree = 3



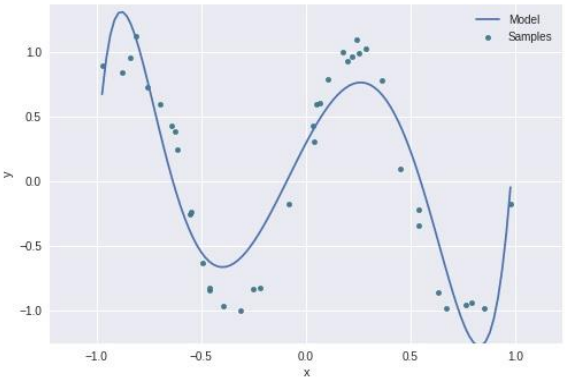


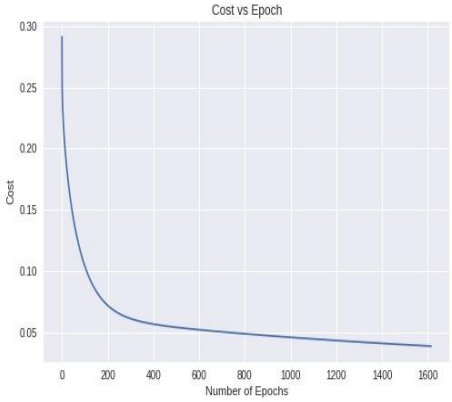
Degree = 7



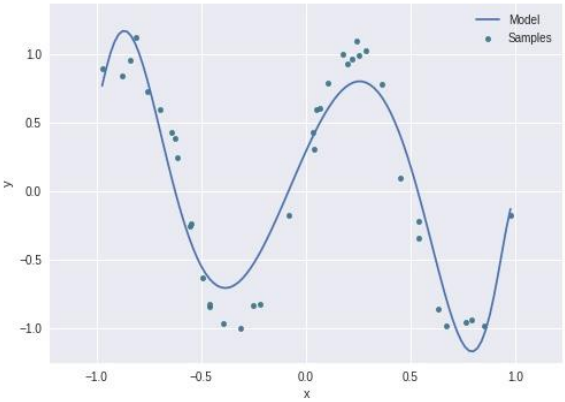


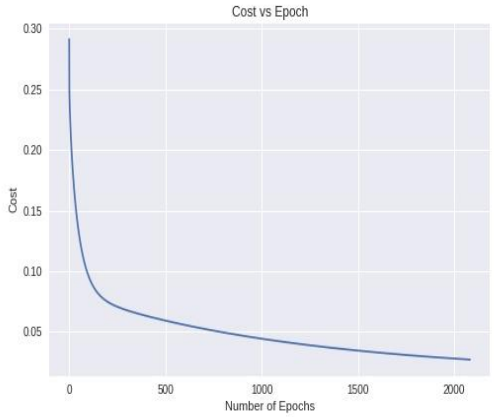
Degree = 11





Degree =15





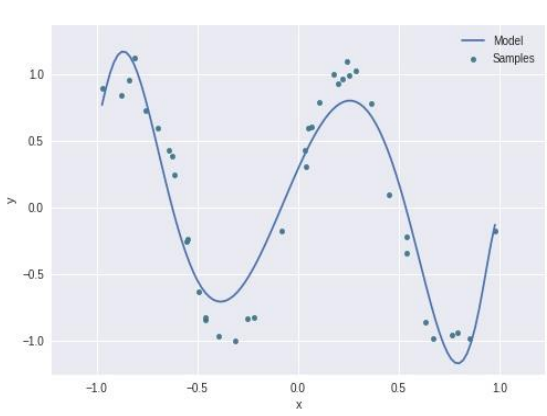
As you increase the degree of polynomial, curve tries to fit according to each and every point in dataset. If dataset has anomalies then it will affect efficiency of model even though loss is minimum. Also, one trend observed from table 2.2 is that number of epochs increases and loss decreases with increase in degree of polynomial. It is clear that higher degree has more parameters to learn and hence model takes more time to converge.

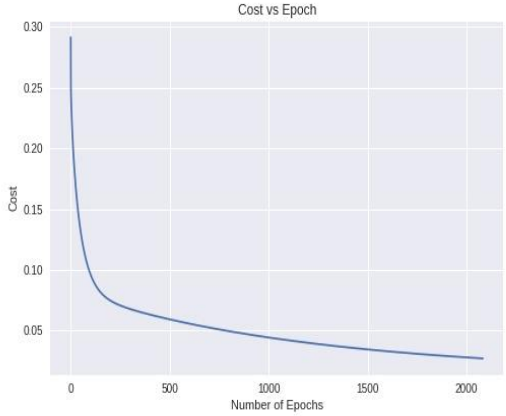
c) Effect of Regularization keeping degree = 15 and learning rate (α) = 1.5

Regularization Parameter (Beta)	Final Loss	Number of Epoch
Beta 0.001	0.027208817	2081
Beta 0.01	0.05681306	1130
Beta 0.1	0.11671056	252
Beta 1	0.20484822	59

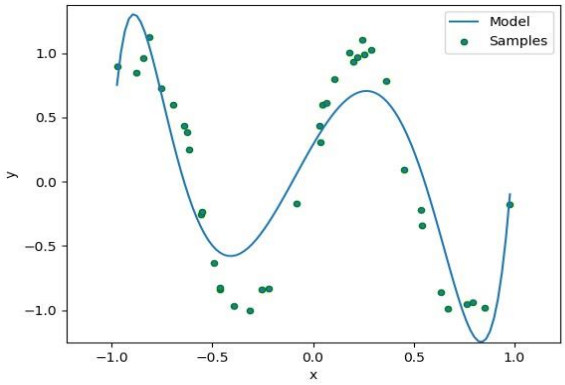
Table 2.3

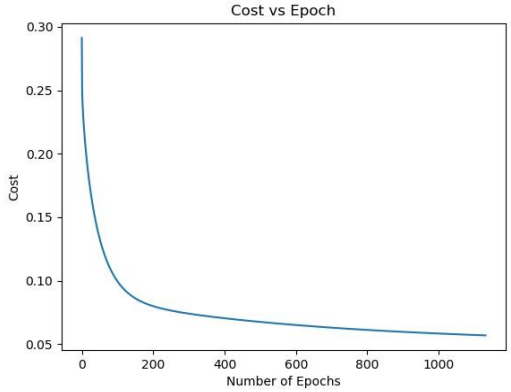
Beta = 0.001



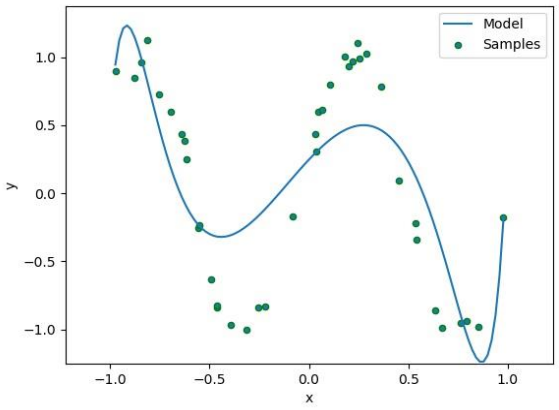


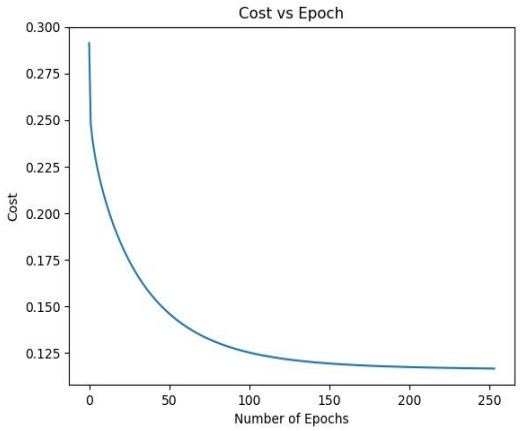
Beta = 0.01



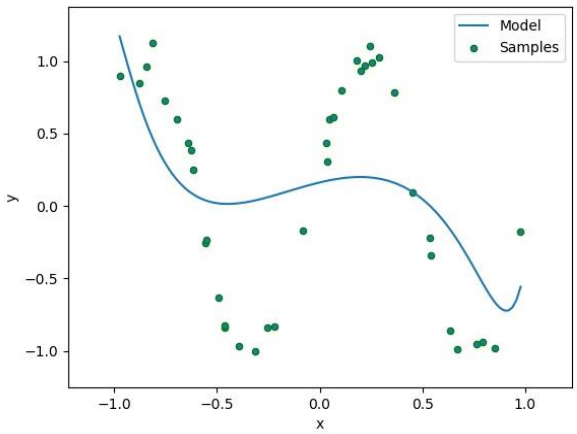


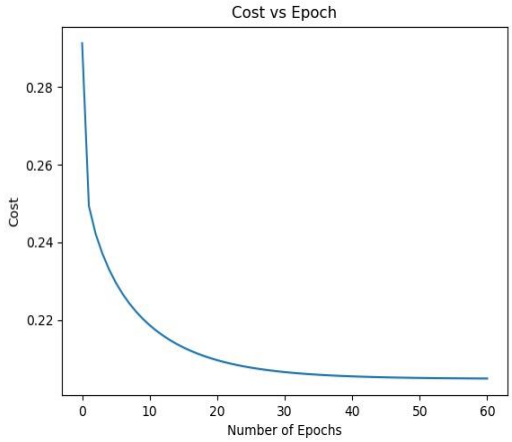
Beta = 0.1





Beta = 1





Effect of regularization can be seen through various values of Beta. As Beta increases, curve becomes less flexible and it actually tries to pull realistic model rather than considering effect of each and every point in dataset. This happens because, regularization coefficient (Beta) reduces variance in distribution of weight parameters of model. It is observed that range of model parameters in case of Beta = 0.001 is way higher than that of Beta= 1. But regularization penalty should not be too high cause it increases final loss value.

Conclusion:

With degree 15 and learning rate 1.5, beta =0.001 gave less possible final loss. Hence these should be one of best possible model parameters. But if you consider time trade off, beta 0.01 curve is nearly same as that of beta 0.001 and it is taking less time (half number of epochs compare to beta 0.001). Hence in my view, beta 0.01, degree 15 and learning rate 1.5 is best solution.

Problem 3: Multivariate Regression

Concept of linear and polynomial regression can be extended to classification task. In such case we can't use regression predicted value, instead we convert them into probabilities in range of 0 and 1. By applying threshold on these probabilities, we can get class labels. In this problem, I kept threshold = 0.5, degree of polynomial = 6 and convergence criteria = 0.00001

Parameters:

Hyper parameter	Value
Learning Rate (α)	1.5
Number of Epoch	140
Regularization Parameter (β)	1
Convergence Criteria (eps)	0.00001
Degree of Polynomial	6
Final Loss	0.5292
Error	16.95 %

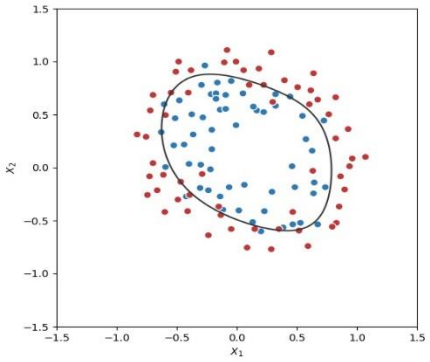
a) Effect on learning rate (α) keeping regularization parameter (Beta) = 0

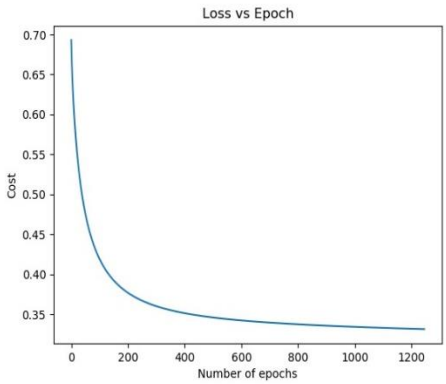
Learning rate	Final Loss	Error (%)	Number of Epochs
0.01	0.6237	NA	1000 (Force Stop)
0.1	0.3864	NA	2500 (Force Stop)
0.5	0.3419	16.10	1837
1	0.3352	16.10	1397
1.5	0.3314	16.10	1244

Table 3

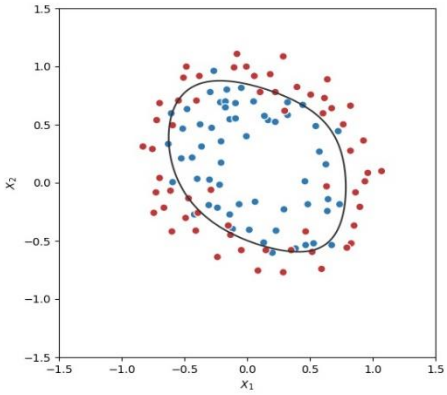
Learning rate 0.01 is too slow. Also learning rate 0.1 took around 2500 epochs (8 hours) to reduce loss into 0.3864. Other higher losses are giving exactly same error. Hence learning rate 1.5 is best suitable for model without regularization parameter as it is taking less epochs.

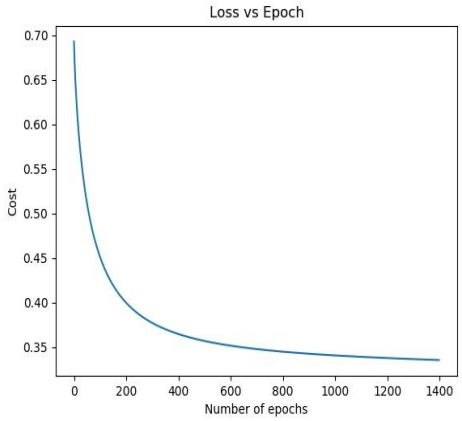
Alpha = 0.5



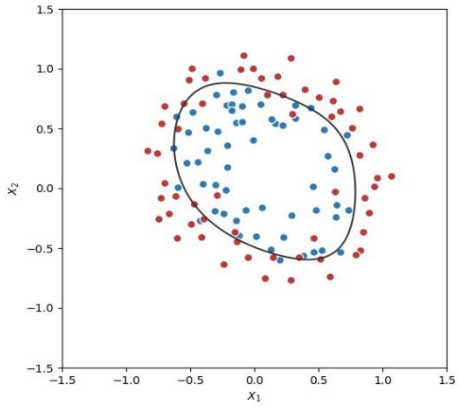


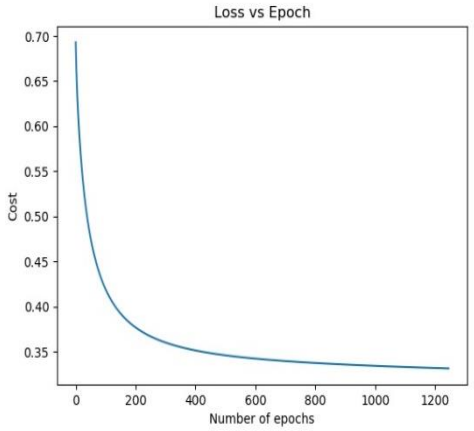
Alpha = 1





Alpha = 1.5



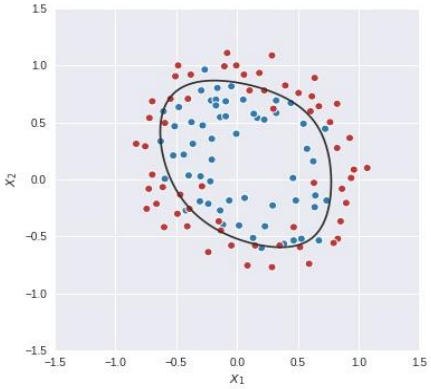


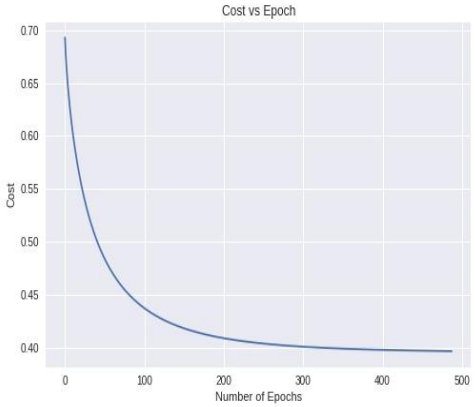
b) Effect of Regularization parameter (Beta) keeping learning rate = 1.5 and Degree= 6

Beta	Final Loss	Error	Number of Epochs
0.1	0.396444173	16.9491525	485
1	0.529263587	16.9491525	140
10	0.648252409	25.4237288	24
100	0.686489749	36.440678	5

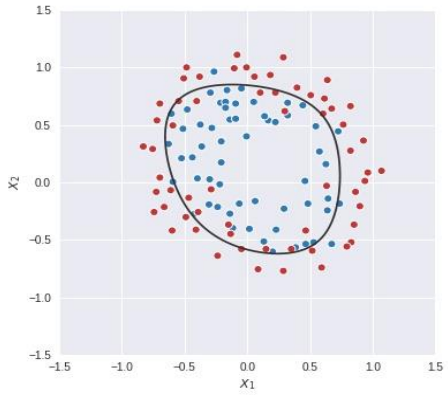
Table 3.2

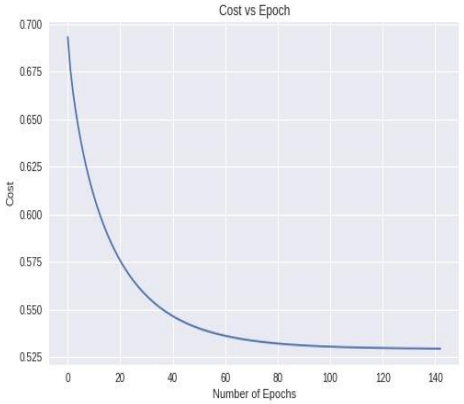
Beta = 0.1



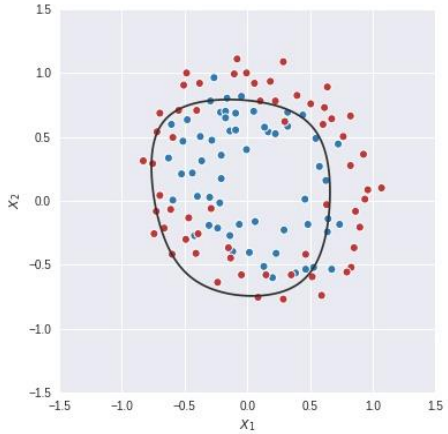


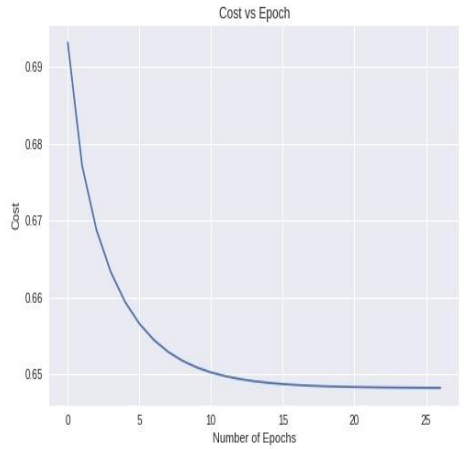
Beta = 1



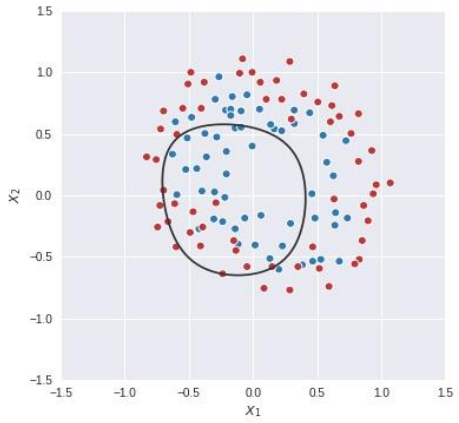


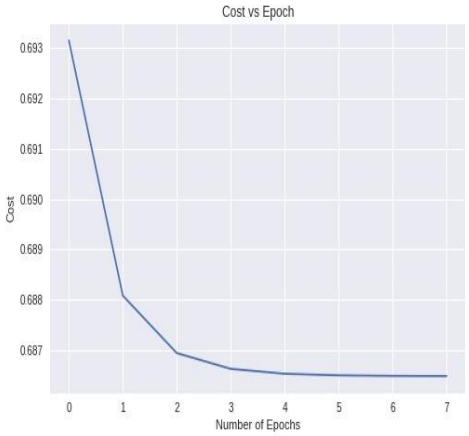
Beta = 10





Beta = 100





c) Observation:

- From table 3.2, it can be seen that beta=0.1 has low loss as compare to beta=1. But scatter plot of beta 0.1 signifies that it is overfitted model.
- On other hand classification error for both beta 0.1 and beta 1 is same. In fact, beta 1 takes less time for convergence as compare to beta 0.1, hence beta 1 is best possible regularization parameter for our model.
- With beta = 0 model have ideal decision boundary. As we increase beta, decision boundary shifted from ideal position.
- With Beta 100, decision boundary covers half of points in both classes which makes no sense in term of classification model as error in this case is too high.

Conclusion:

Overfitting is challenge while applying machine learning techniques. This challenge is significant for multivariate regression because it might end up giving complex decision boundaries to minimize square error. Because of this our classifier can become confident of probabilities it predicts. Over-fitting can be reduced by using regularization (L2) parameter to penalize large coefficient values of model parameters. Adding regularization term will increase loss and model would not be ideal model but it could be practical for new data points.