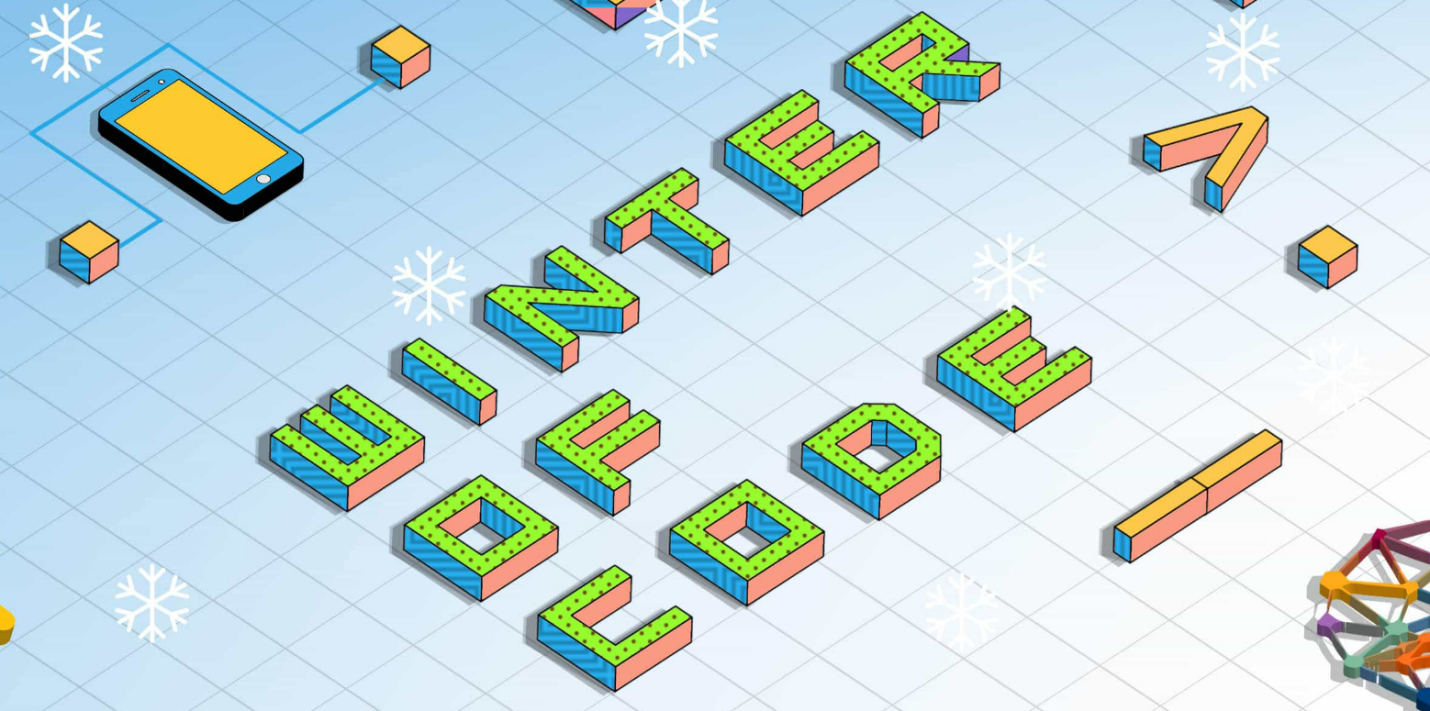
**Machine Learning Report for WOC**

****

1. Linear regression

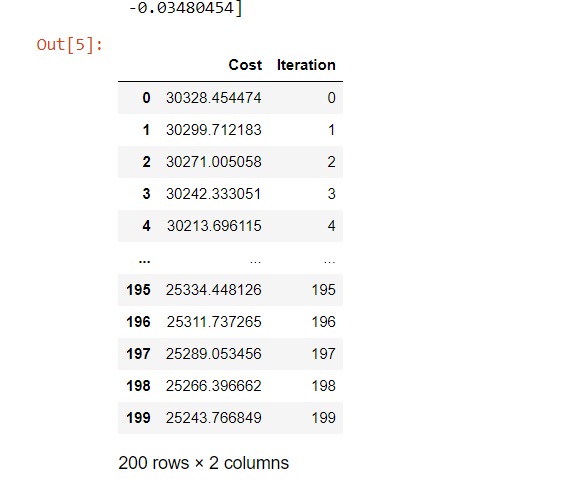
The hyperparameters in these two regressions are:

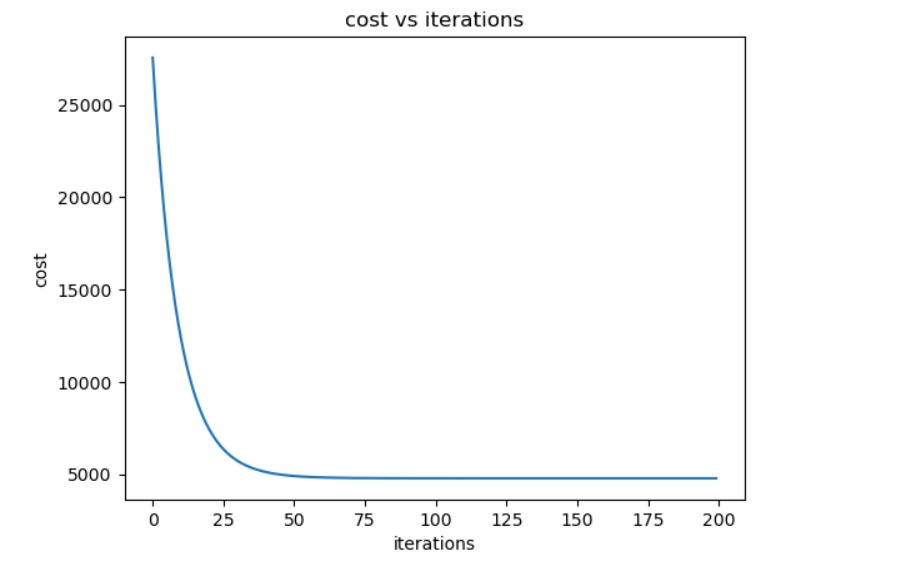
a. Alpha (learning rate)

b. No. of iterations

I have written the code in such a way we can change these parameters according to our choice. So hence looking at the values of cost function and accuracy parameter (R2 square) we can decide what choice of Alpha and No. of Iterations best suits our data set

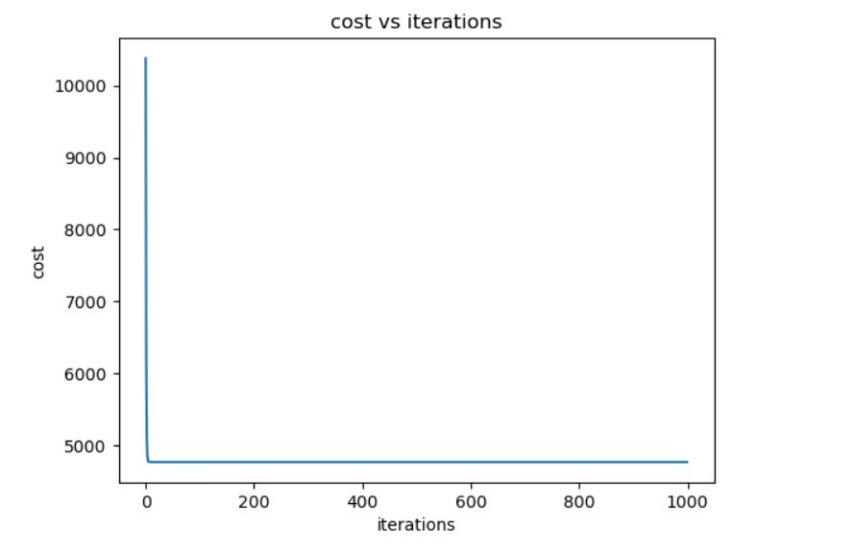
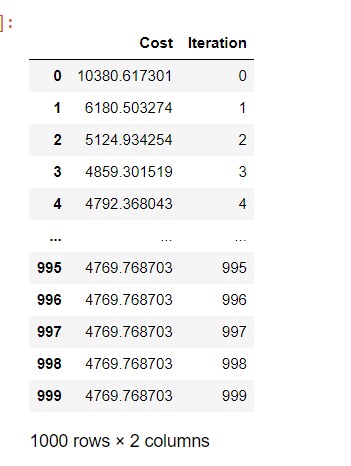
Case 1 : Alpha = 0.0005 and No. of iterations = 200





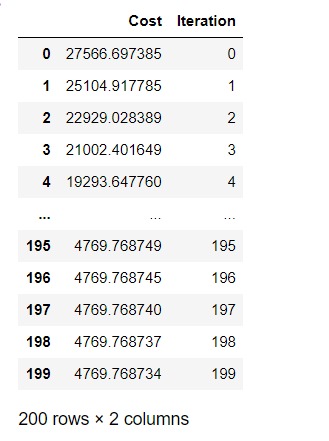
As you can see the cost has not decreased a lot and so I increase the alpha

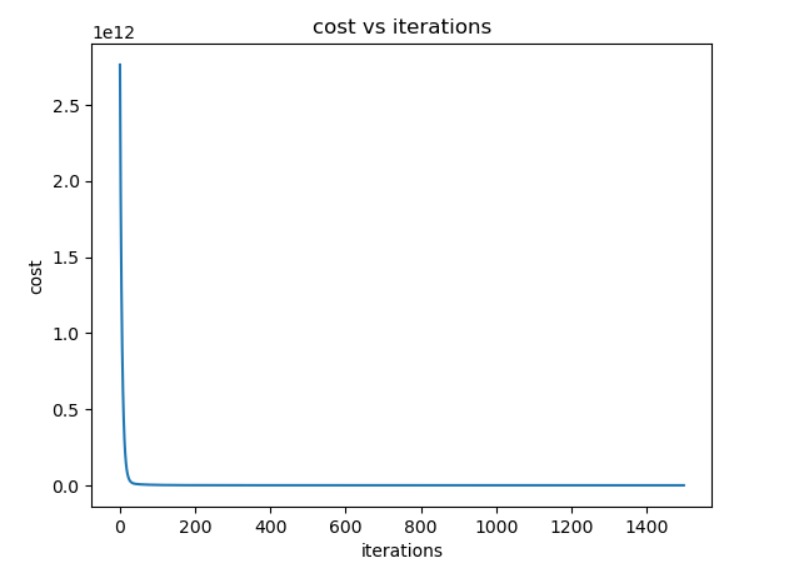
Case 2: Alpha = 0.001 and No. of iterations = 200



I increased alpha but the no. of iterations according the graph looks too much, and in the next run I decreased the no. of iterations.

**Case 3** : Alpha = 0.05 and No. of iterations = 200 after tuning the hyperparameters we get this as the





And the accuracy according to R2 score is 0.824

2)Polynomial Regression

The hyperparameters in this regression are:

a. Alpha (learning rate)

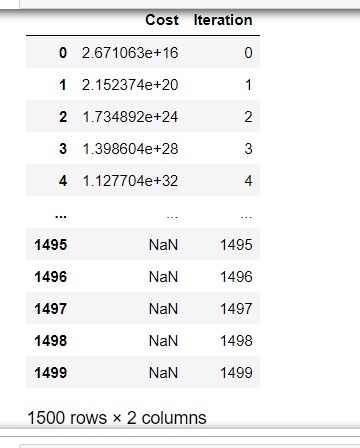
b. No. of iterations

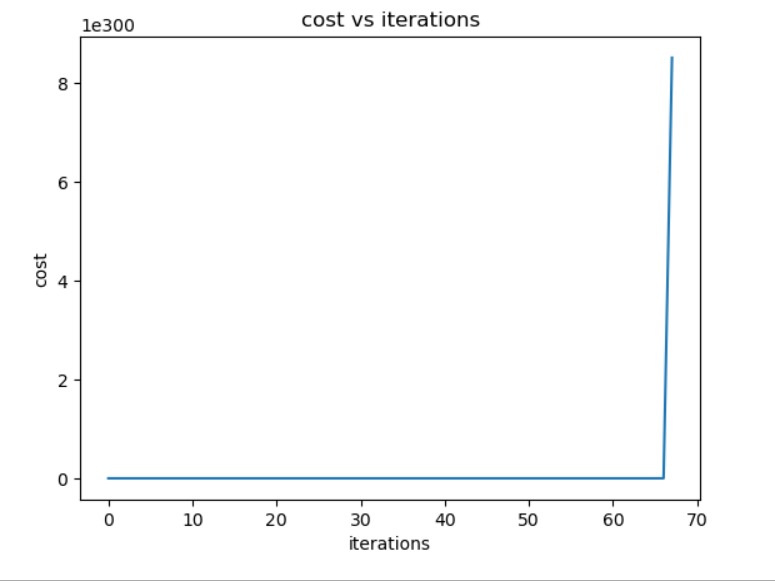
c. power

Case 1: alpha = 0.1 & iterations = 1500 & power = 5

We have seen that from the graph of parameters with y – train is an odd degree polynomial

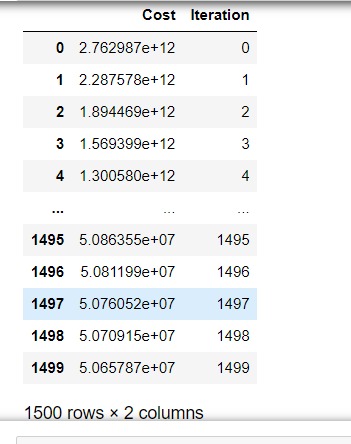
so hence I decided to take power = 5. The accuracy with power 3 is coming out to be 0.78 (R2 square) and with power = 5 it is coming out to be 0.999(r2 square)

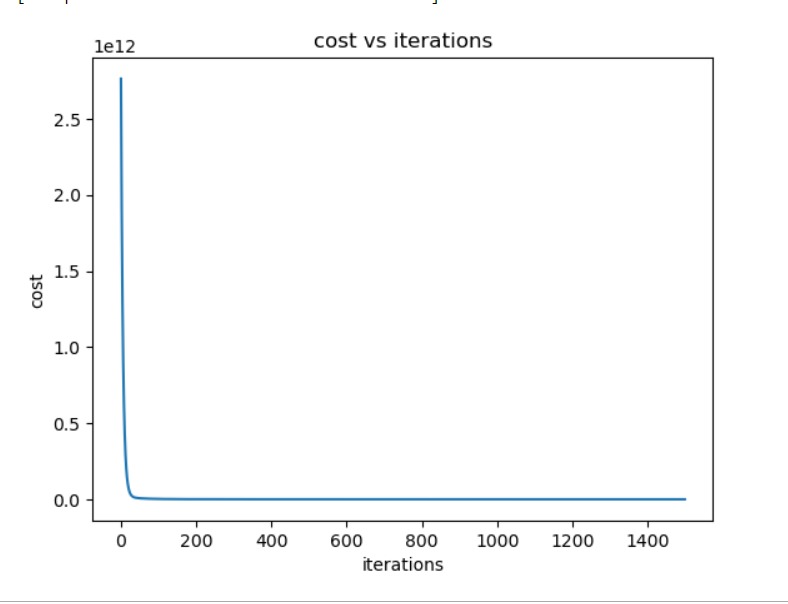




The cost is very high because of very high alpha, so I decided to decrease the alpha

Case 2 : alpha = 0.0001 and no. of iterations = 1500





Accuracy = 0. 99924

3) Logistic Regression

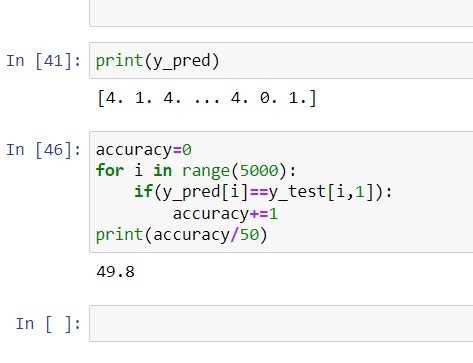
The hyperparameters in this regression is :

1. Activation function
2. Alpha
3. No. of iterations

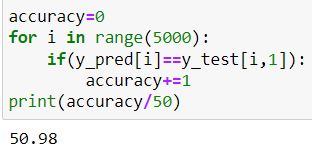
1.Activation function: Of course sigmoid function is the main essence of logistic.

Case 1 : Alpha = 0.01 and No. of Iterations = 200 for each y – label

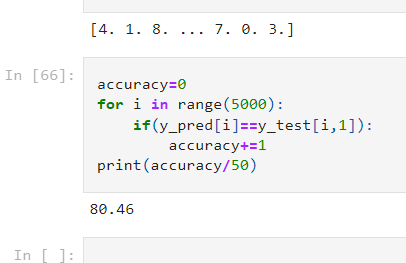
Accuracy = 49.8



Case 2: Alpha = 0.05 for all except 8 th label for which I took alpha = 0.1 and No. of iterations = 200



I had forgot to add z\_score to the training set and once I added it the accuracy increased to 80 percent.

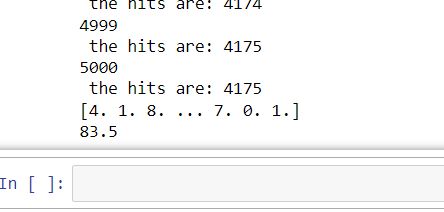


4) K- Nearest Neighbour

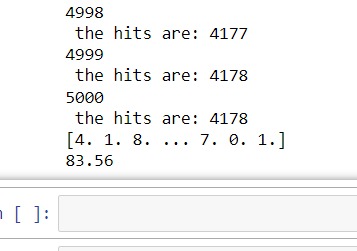
This main hyper-parameters in this algorithm is the number of k-nearest neighbour:

I took the no. of k values randomly.

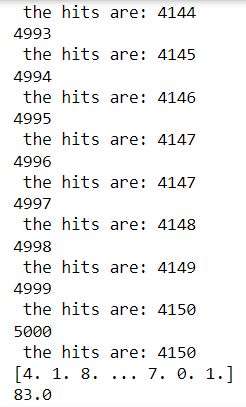
Case 1: k = 5

This is the accuracy

Case2: k = 7



Case 3: k = 10



And so here k = 7 is coming to be the highest so I took k = 7

5) N – layered Neural Network

I tried making it, and I was able to successfully implement forward propagation but backward propagation was difficult to implement but I derived the partial derivatives and tried to implement it and update the parameters and if I was given more time I think I would have been able to implement it.