**ML BOOTCAMP**

**Description-**Build your own machine learning library

**LINEAR REGRESSION**

**This is the first model in the library built .**

**First we split the training data set provided into two four separate arrays ,X\_train,y\_train,X\_test,y\_test.**

**X\_train**

**COST FUNCTION:-**

**The cost function is basically the error between the actual data and the predicted data.**

**We use hyper parameters such as W and b.**

**We start by initializing the hyper parameters to zero and then use gradient descent on them to get to the value of W and b where the cost function converges or is the least**

**HYPER PARAMETERS:**

**W(optimal) b(optimal)**

[[2.19752012e+01] [3.17287776e+03] 8.30116145405199

[3.32587218e+01] [2.30066751e+02]]

[9.97194643e+00]

[6.05587238e+00]

[1.77200369e+02]

[3.29675733e+02]

[2.53110919e+02]

[8.98042262e+02]

[4.34536242e+02]

[8.40428002e+01]

[1.21535358e+02]

[1.60195445e+03]

[1.67874941e+03]

[1.31936615e+03]

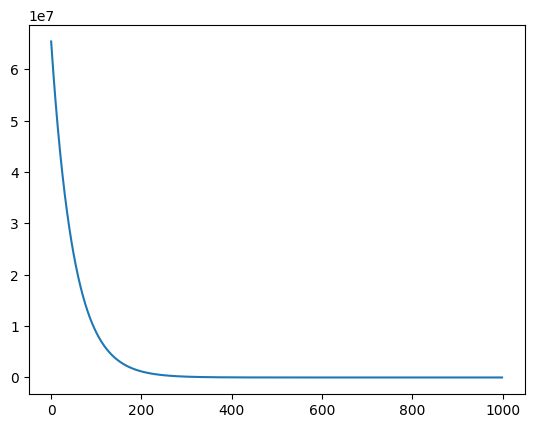
[5.99746393e+02]

[1.05470877e+04]

[5.03838424e+02]

[6.78209249e+02]

**This below is the plot of cost function versus the number of iterations.It is a decreasing curve which shows that with each iteration the cost function rapidly decreases .**



**TESTING:-**

**We get the optimal hyper parameters after running the gradient descent on the initial hyper parameters . We use these hyper parameters to calculate the predicted y and then use a formula called r2score to calculate basically the accuracy of the developed/predicted values in comparison to the original value of y .**

**r2score=0.999851,this is the value of r2score for my implementation of the linear regression model.**

**POLYNOMIAL REGRESSION**

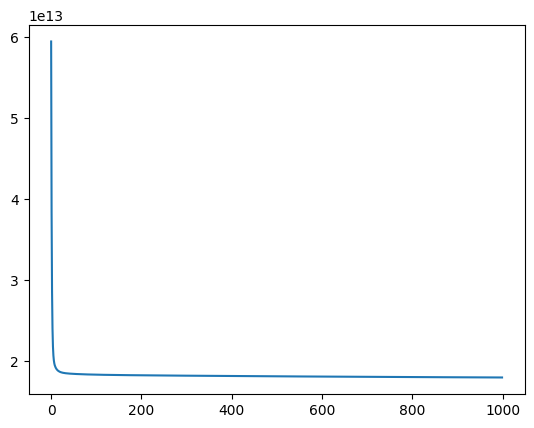
**Here we again split the given data set into a training and testing data set.**

**Description:**

**In polynomial regression we create a new X\_train array by appending the square,cube,… of the given X\_train data set and hence it basically becomes y=w1x+w2x^2+w3x^3+…..+b.**

**Here I am doing this because I don’t know what polynomial fits the given data set best.Hence we give all the polynomials up to a certain degree and then run the model The hyper parameter corresponding to the polynomial feature which bests suits the data given has a greater value when compared to other polynomial features.**

**Here again the gradient descent is same as that in linear regression.**

**The following is the plot of cost function versus the number of iterations .The cost function is rapidly decreasing with the increase in number of iterations.**  
  
  


**R2SCORE:**

**We obtain the optimum hyper parameters by training the model on the training data set .Upon using these hyper parameters to test the model we obtain a predicted y.**

**Hence the r2score obtained was:0.595513**

**LOGISTIC REGRESSION:-**

