ML BOOTCAMP

DESCRIPTION:

Building a machine learning algorithm library which involves

implementation of Linear (and Polynomial) Regression, Logistic

Regression, KNN, K-Means Clustering and an n-layer Neural

Networks from scratch using python, Training and Testing dataset.

LINEAR REGRESSION:

The goal of linear regression is to find the values of the coefficients that minimize the difference the predicted and actual values of the dependent variable.

First we will split the above dataset in two parts ‘x’ and ‘x\_test’ in which ‘x’ is

used for training and ‘x\_test’ is used for testing the code.

The code includes normalization ,calculation of cost function and gradient

Descent.

NORMALIZATION:

Normalization is uesd to bring all the features to a similar scale, preventing some features from dominating others during the process in machine learning algorithms.

The common normalization technique involves transforming the values of each feature so that they have a mean of 0 and a standard deviation of 1.

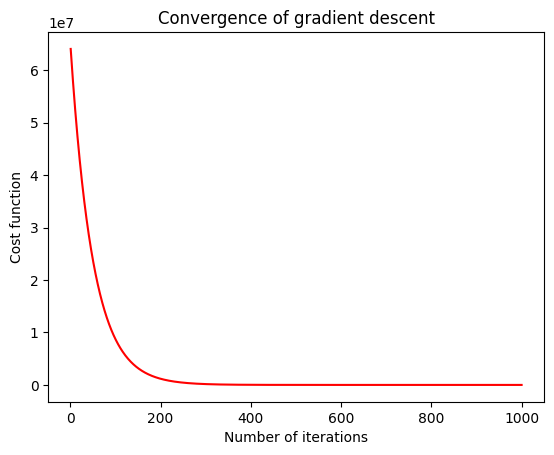
COST FUNCTION:

Cost function is nothing but the mean squared error.The mean squared is calculated as the average of the of the squared difference between the predicted values and the actual values.The goal during the training of the MLR model is to adjust the model parameters to minimize the cost function.

GRADIENT DESCENT:

Gradient descent is used to minimize the cost function.

At each iteration the gradient of the cost function with respect to the model parameters is calculated.The gradient points in the direction of the steepest increase in cost function.The parameters are then updated by moving them in the opposite direction of the gradient. This process is repeated until convergence.The graph between number of iterations and cost function for a learning rate of 0.01 is given below

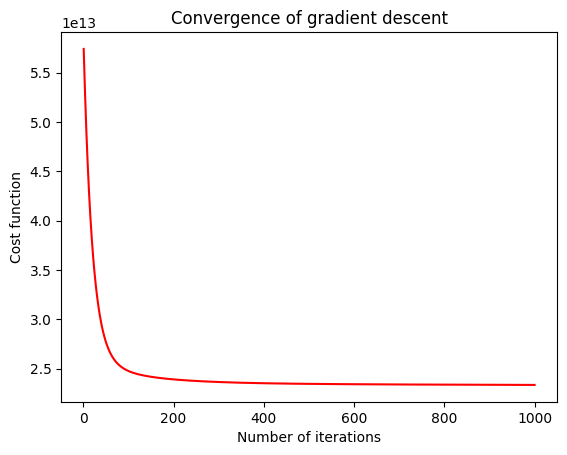
R2score=0.9998516125074495 

**R2score=0.9998516125074495**

POLYNOMIAL REGRESSION:

The idea of polynomial regression is same as linear regression but in this we not only use linear terms but also there squares,cubes,etc.

The graph between number of iterations and cost function for a learning rate of 0.01 is given below



**R2score=0.64039**

KNN:

KNN is used for classification.KNN is based on the principle that similar data points tend to have similar outcomes.

In this we calculate k nearest points from the required points and then we check the category where maximum number of points lie.

**R2score=0.972867** for k=6