WINTER OF CODE

ML BOOTCAMP

PROJECT REPORT

By:
Aditya Gupt
21JE0045

The main objective was to implement Linear regression, Polynomial regression, Logistic regression, KNN, K-means clustering and Neural Network from scratch. All the work has been done in the jupyter notebook. All the codes of models implemented has been pushed in the GitHub repository.

In implementing I have used pandas, NumPy and matplotlib for accessing and reading the file, mathematical calculations and plotting graphs (wherever required), respectively.

GitHub repository link:

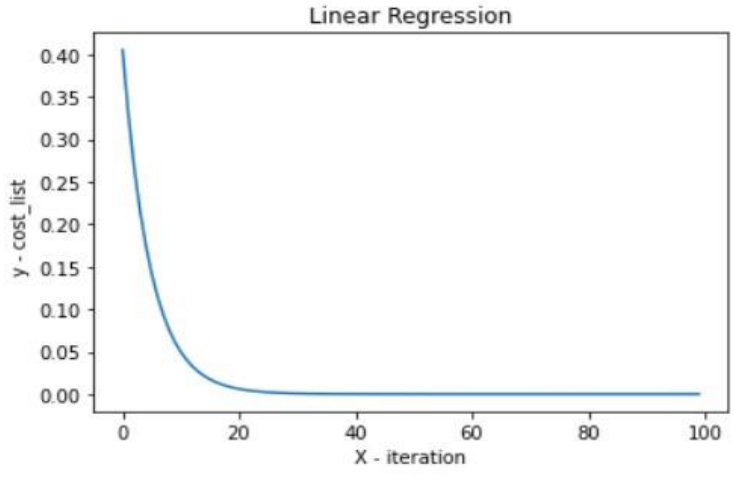
https://github.com/31032004aditya/ML-BOOTCAMP.git

Linear Regression:

Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Linear Regression predicts the value based on independent variables by minimising the error using cost function.

So, this regression technique finds out a linear relationship between x (input) and y(output). Error is minimised by increasing the number of iterations and suitable learning rate by using the concept of gradient and gradient descent.

Visual representation of the cost function against number of iterations:



Polynomial Regression:

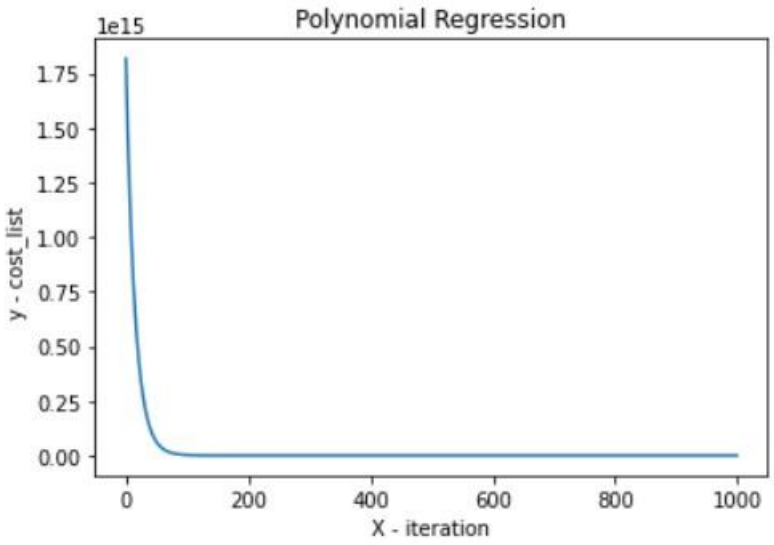
A form of regression analysis in which the relationship between the independent variable x and the dependent variable y is modelled as an nth degree polynomial in x.

Polynomial Regression predicts the value based on independent variables by minimising the error using cost function.

So, this regression technique finds out a polynomial relationship between x (input) and y(output). Error is minimised by increasing the number of iterations and suitable learning rate by using the concept of gradient and gradient descent.

I have used second degree polynomial in the given data set and RMSE for the prediction of how concentrated data is around the polynomial of best fit.

Visual representation of the cost function against number of iterations:

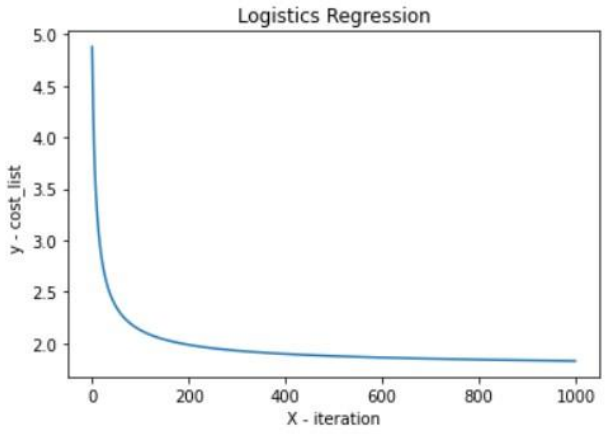


Logistic Regression:

In implementing the logistic regression model, firstly I have defined the basic sigmoid function. The predicted value i.e., the value of Y should lie between 0 and 1, so for that sigmoid function is used.

Then I have defined the error function for the model. After that just like I did in linear regression, similarly, here also I have defined and used the gradient and gradient descent function and updated the variables assumed in the model after each iteration until the error obtained is minimum. After discovering the required variables our model is complete and ready to be implemented.

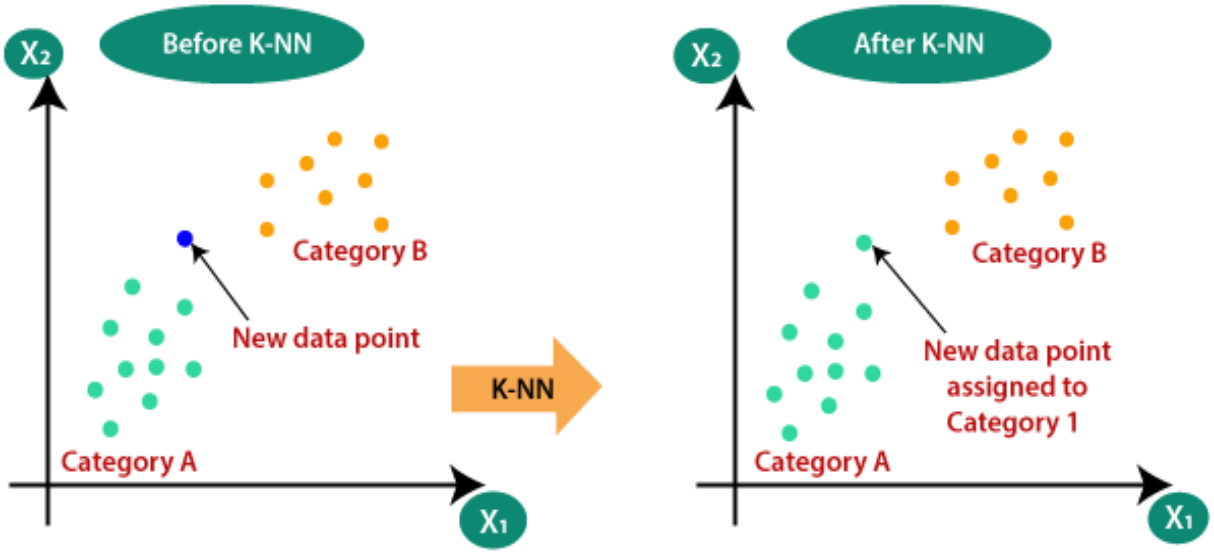
Visual representation of the cost function against number of iterations:



K-Nearest Neighbor:

K-Nearest Neighbour is one of the Machine Learning algorithms based on Supervised Learning technique.

K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using the Euclidian distance between two concerned points.

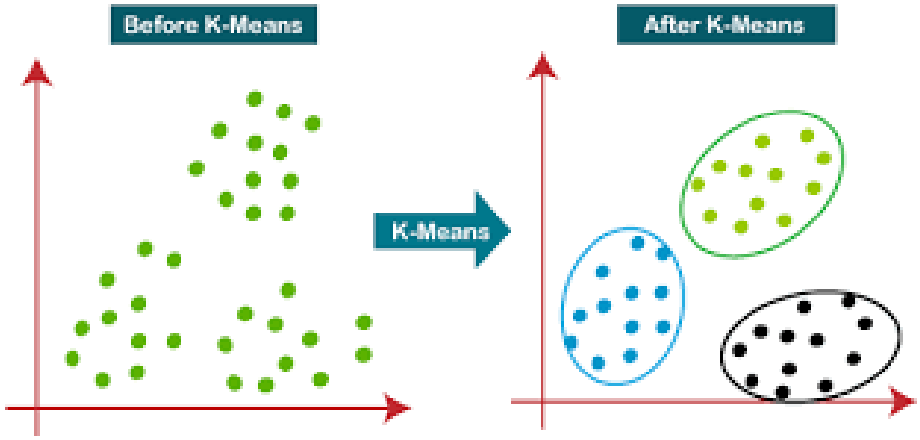


K-Means Clustering:

K-means clustering is one of the unsupervised machine learning algorithms.

My main objective for K-means clustering is to group similar data points together, discover similar patterns and look for a fixed number (*k*) of clusters in a dataset.

To achieve this, I assigned K number of centroids, clustered ones with Euclidian distance from the centroid and then updated it regularly by using mean of all the points lying in the cluster till consistency.



Neural Network:

In the neural network model implementation, I tried a simple neural network including the backpropagation process.

I first defined the basic equation by which the predicted value would come, it was a sigmoid function.

After this I have defined the train function to train the data by using feed forward and back propagation function in which I used gradient and gradient descent for updating weights. I used appropriate number of iterations and suitable learning rate for achieving authentic accuracy.

After all the process of minimizing error and finding of the weights, the model is ready to be implemented.