Report

Linear Regression:

For Linear Regression I have done it in two different methods:

- Using Gradient Descent
- Matrix Approach

R2_Score for my Linear Regression Model on Training dataset is coming out be approximately 0.999999999922914 using both the approaches.
R2_score for my Linear Regression Model on Testing Dataset is also coming out to be approximately 0.999999999922914 using both the approaches.
In Gradient Descent Approach, at Learning rate=0.01 and Epochs = 1000, model is optimised and r2_score is maximised.

Polynomial Regression:

In Polynomial Regression I have tested it for 2-degree in which r2_score for Training Dataset is coming out to be approximately 0.3062362964636467 and for testing dataset it is coming out to be 0.9503860368295035.

Further checking it for 3-degree, I have got r2_score for Training data set approximately to be 0.113739 and r2_score for testing dataset approximately to be 0.99041720636767346.
Taking the value of leaning rate to be 0.01 and epochs 1000 the solution was not optimized thus decreasing learning rate to be 0.001 the solution moves towards optimization.
I tried for the 4-degree polynomial also but I was not able to run it as some error occurred. After some manipulation the code runs but it gives more error than the 3-degree polynomial, thus maybe I have done something wrong but I was not able to find it out.
Thus, I wasn't able to do it for the 4^{th} degree polynomial.

Thus, as solution is more optimised for 3-degree polynomial rejecting the 2-degree polynomial.

Logistic Regression:

For learning_rate=0.1 and epochs=500 the value of r2_score is comin g out to be 0.776754

For learning_rate=0.1 and epochs=1000 the value of r2_score is foun d to be 0.697243

For learning_rate=0.01 and epochs=500 the value of r2_score is comi ng out to be approximately 0.89453

For learning_rate =0.01 and epochs=1000 the value of r2_score is co ming out to be 0.9372215035329756. This is the maximum r2_score I have reached out of which I tried.

Since the code is taking a lot of time to run, I was able to do only the above experiments.

KNearestNeighbors:

The most crucial part in KNN is to choose the value of K.
I have a run a for loop in which K varies from 1 to 10 and calculating t he value of r2_score which again takes a lot of time so I ended the co de and do it manually.
But till k=3 for loop runs and in which r2_score for k=3 is maximum a nd.
When I was doing manually, I started with the value of k to be 4 and after that k=5 in which r2_score remains constant and same when k= 3.Then I take the value of k=sqrt (30000) (i.e. no. of data points in dat aset) and value of r2_score is found to be the same.
Thus, after k=3 value of r2_score remains constant.

NNeuralNetwork:

In Neural Network I have taken one hidden layer in which there are 5 12 neurons and 10 neurons in output layer. Thus, for this test accura cy is coming to be 88%.

88% accuracy is coming out when epochs=30 and learning_rate=0.00 1

When epochs=20 and learning_rate =0.01 accuracy is coming to be 9 5% and r2_score to be 0.93511.

I have tried no. of neurons in hidden layer to be 128 for which the tes t accuracy is coming out to be approximately 71%.

I have taken no. of neurons in the Input layer to be 512 along with th e 128 in hidden layer and 10 in output layer for which the test accura cy is coming out to be approximately 74%

For one hidden layer such that: input_size=784(28*28)

hidden_size=512

output_size=10

I have got the 88% test accuracy.

Since the code is taking a lot of time to run (not less than 7-6 hrs) I w as able to test it only on the above different parameters.

KMeansClustering:

Since KMeans Clustering is an unsupervised Machine Learning Algorit
hm, the model I have prepared will classify the data points in differen
t cluster based on the no. of cluster given to the model.