# Machine Learning Library

## Project Report

Priyanshu Kumar

IIT (ISM) Dhanbad

AIM: To build own machine learning library from scratch using numpy ,pandas and matplotlib.

The algorithm that this library should contain:

1.Linear Regression

2.Polynomial Regression

3.Logistic Regression

4.N-Layers Neural Networks

5.KNN

6.K-Means

## 1.Implementation detail of Linear Regression.

In linear regression, the dataset was having 20 features and 50000 training examples. I divided the training set into 40000 set examples and 10000 cross-validation set.

### Hyperparameters and training logs:

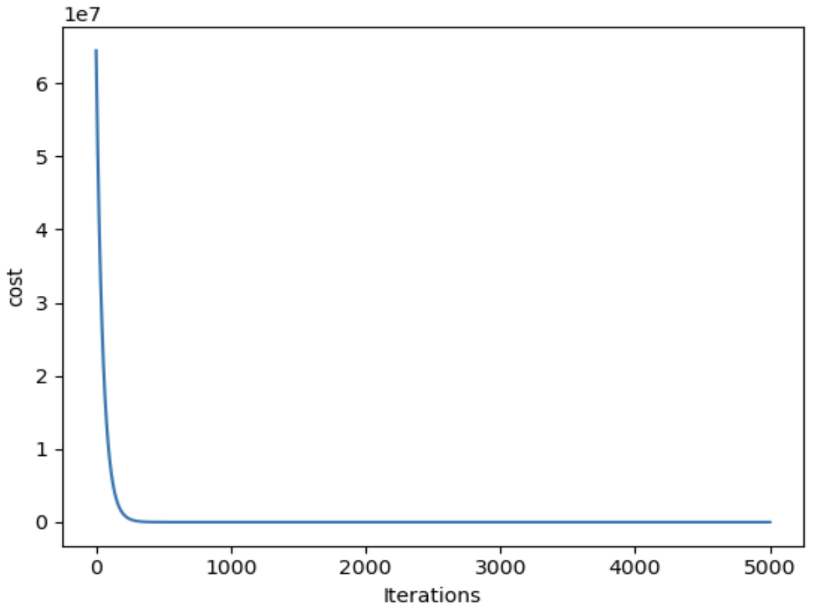
1.Loss Function: The loss function ,I used to calculate model’s error on the training data was Mean Squared Error(MSE).

2.Learning Rate(alpha): The learning rate used in the training process by me was 0.01. For higher value of alpha the cost was overshooting and lower value of alpha was too slow to converge the cost.

After trying few values of learning rate , I arrived to the value of 0.01.

3.Iterations: I used 5000 epochs . but ,the cost was saturated on 1500 epochs only.

I plotted the cost vs iterations.



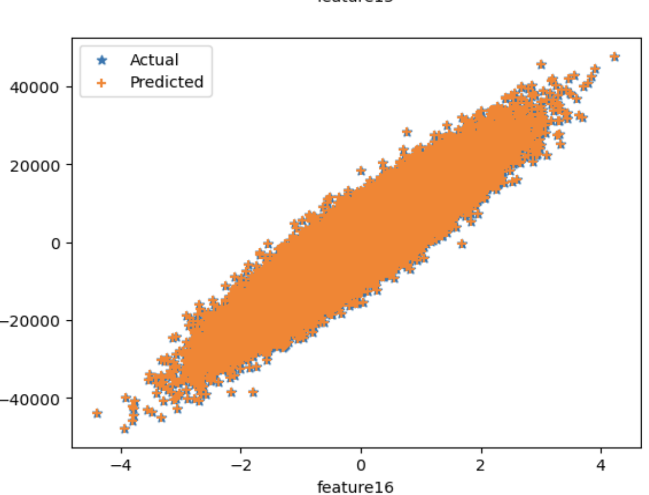
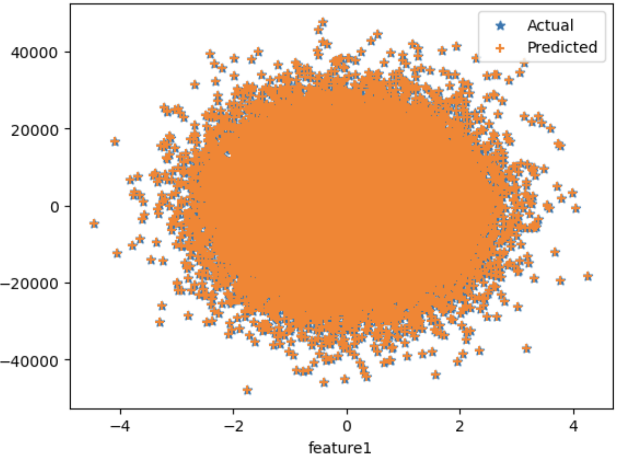
4.Evaluation Metrics: I used R2 score as evaluation metrics. R2 score assess the goodness of fit of a model.

R2 score for training set was :0.9999999999232193

R2 score for cross validation set was : 0.9999999999216167

The above R2 score shows that our model has performed well on training as well as cross validation set. So, we can use this model to predict the test set.

We can see the actual vs predicted graph to get the dea how well our model has performed.



## 2.Impelmentation of Polynomial Regression

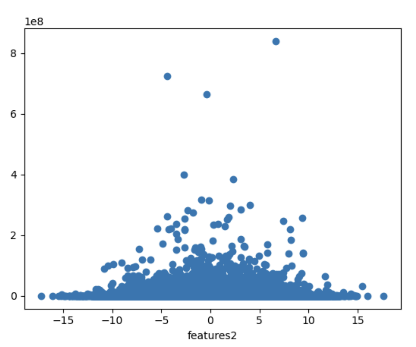
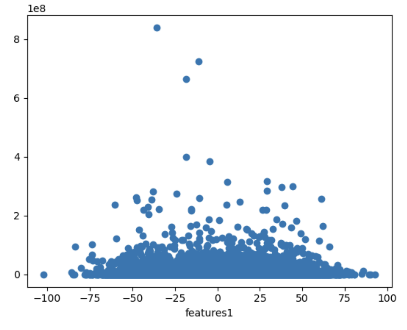
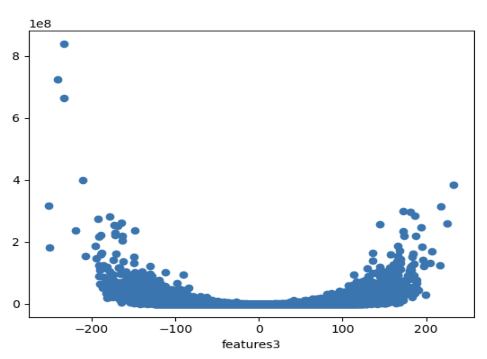
In polynomial regression ,the dataset was having 50000 training example and 3 features.

We need to see the plot of data to visualize the data.

I have chosen the degree 6 for my model as I check the R2score for each degree between 1 to 6.

Best R2 score was at degree 6.

The plots b/w features vs Y labels are shown below:



### Hyperparameters and Training Logs

1. Loss function : The loss function ,I used to calculate model’s error on the training data was Mean Squared Error(MSE).

2.Learning Rate : Learning Rate , I used for training my model is 0.03 because it was overshooting for higher values of learning rate.

3.Iterations: I used 20000 epochs, the cost was saturated at 80000 epochs only.

4.Evaluation Metrics: I have used R2\_score to evaluate model.

R2 score of train data at degree 6: 0.9999999979760448

5.Normalizing data: I have used z\_score normalization to normalize the data which makes the gradient descent faster to reach minima.

## 3.Logistic Regression

For classification task , the data was having 30000 training examples. I divided the training data into 25000 training data and 5000 cross validation data. There were 10 classes and each training example was having 784 pixels(features).

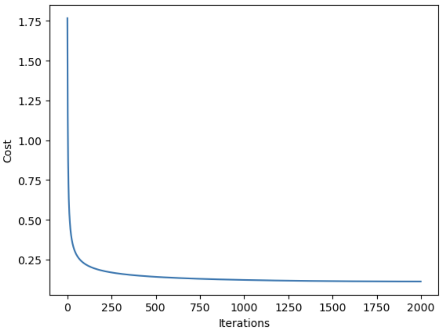
I have used softmax regression for multiclass classification.

## Hyperparameters and Training Logs

1.Loss Function : Loss function used for softmax regression is Categorical cross-entropy.

2.Learning Rate: Learning Rate , I used is 3e-5 as it was overshooting for higher values and was very slow for lower values of learning rate

3.Iterations: I used 2000 epochs .

 We can see that the cost suddenly drop upto 200-250 epochs and suddenly flattens.

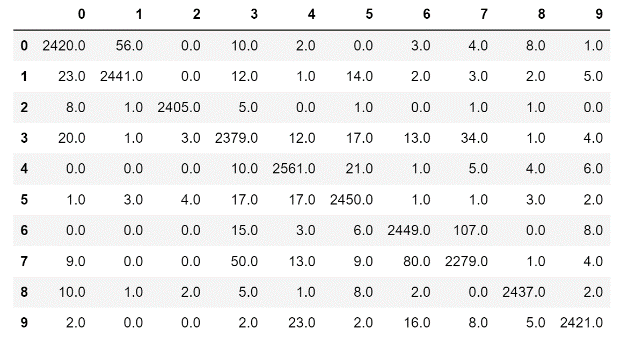
4. Evaluation Metrics: I have used Accuracy and F1 score as evaluation metrics.

Accuracy on Train-data: 0.96968

Accuracy on cross validation set: 0.96

Our model has misclassified 759 out of 25000 training examples.

Confusion Matrix:

 The row specifies the actual class where as column specifies the predicted class .

This matrix tells which class is misclassified and assigned class of misclassified data.

This is also used for calculating F1 score.

F1 score of train-data: 0.9697380320763

5. Normalizing data: I divided training data by maximum value of training data i.e. 255.

## 4.N-layer Neural Network

For classification task , the data was having 30000 training examples. I divided the training data into 25000 training data and 5000 cross validation data. There were 10 classes and each training example was having 784 pixels(features).

### Hyperparameters and Training Logs:

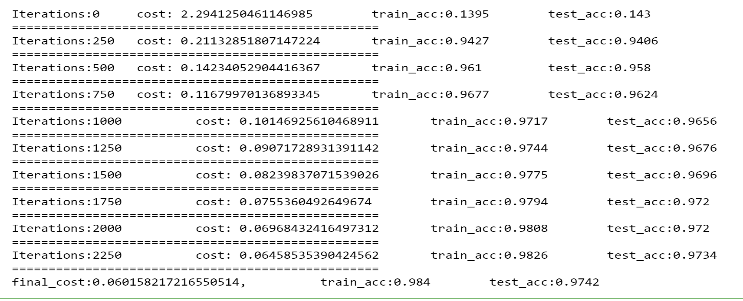
1. Structure and activation functions of neural network: Structure for my neural network is having 2 hidden layer with 16 neurons each with relu activation and the output layer of 10 neurons with softmax activation.

2. Loss function : Cateorical cross-entropy loss.

3.Learning rate : I used 0.1 as my model’s learning rate as it was not overshooting and was converging fast.

4.Iterations: I used 2500 epochs.

5.Evaluation Metrics: I have used accuracy as evaluation metrics.

Accuracy on train data : 0.984

Accuracy on test data : 0.9742

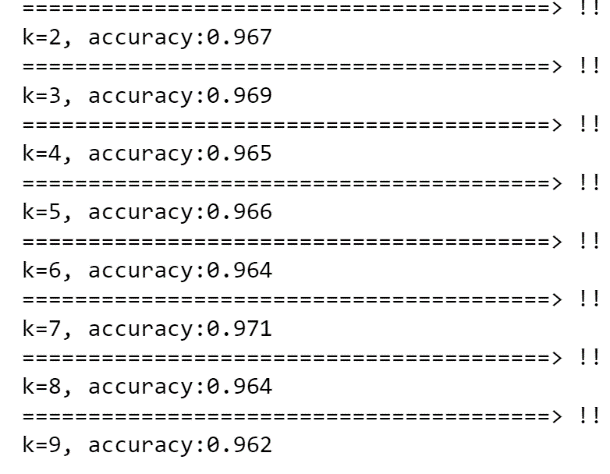
## 5.KNN :

For classification task , the data was having 30000 training examples. I divided the training data into 25000 training data and 5000 cross validation set.

### Hyperprameters and Training logs:

1.Value of K: I selected value of K as 7 because it was giving best accuracy on cross validation set.

I checked the accuracy for values of k between 2 to 10 and find the accuracy and at k=7 accuracy was best on cross validation set.

 At k=7 , accuracy was 0.971, due to which I select optimal k=7.

2.Evaluation Metrics: Evaluation Metric used was accuracy.

Accuracy on the cross validation set : 0.9808 at K=7

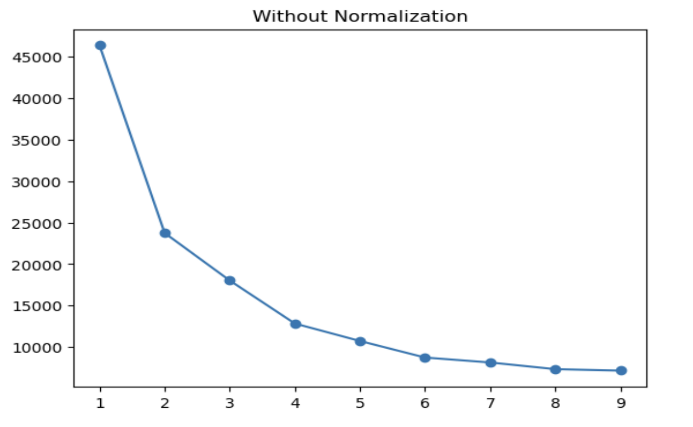
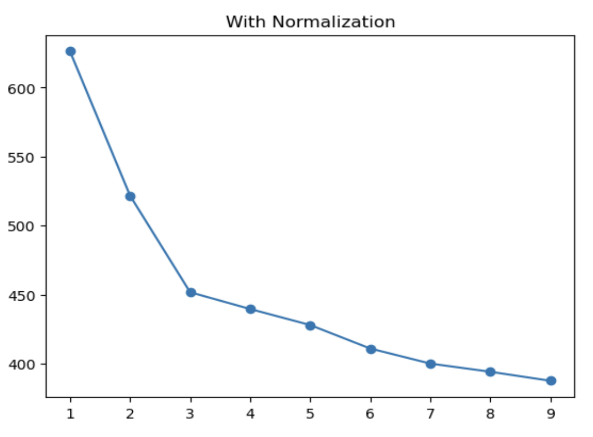
## 6.K-Means:

The provided dataset for k-means was having 178 examples with 13 features. The task was to find the optimal no. of clusters .

### Hyperparameters and training logs:

1.Finding optimal number of K: I have used Elbow method to find the optimal no. of K. There was two different values of k . The value was 2 without normalizing the data and 3 after normalizing data.

I have plotted wcss(within centroid sum of distances) to get elbow point.



We can see that the in fig1. With normalization the elbow is at k=3, whereas k=2 with normalization.

This is because the distribution of data changes when we normalize and hence the no. of clusters are different.