ML REPORT (WOC 6.0)

PROBLEM STATEMENT : To implement the machine learning models : Linear Regression , Polynomial Regression , Logistic Regression , KNN(K-Nearest Neighbours) , NNN(N-Layered-Neural-Network) , K-means Clustering.

LINEAR REGRESSION:-

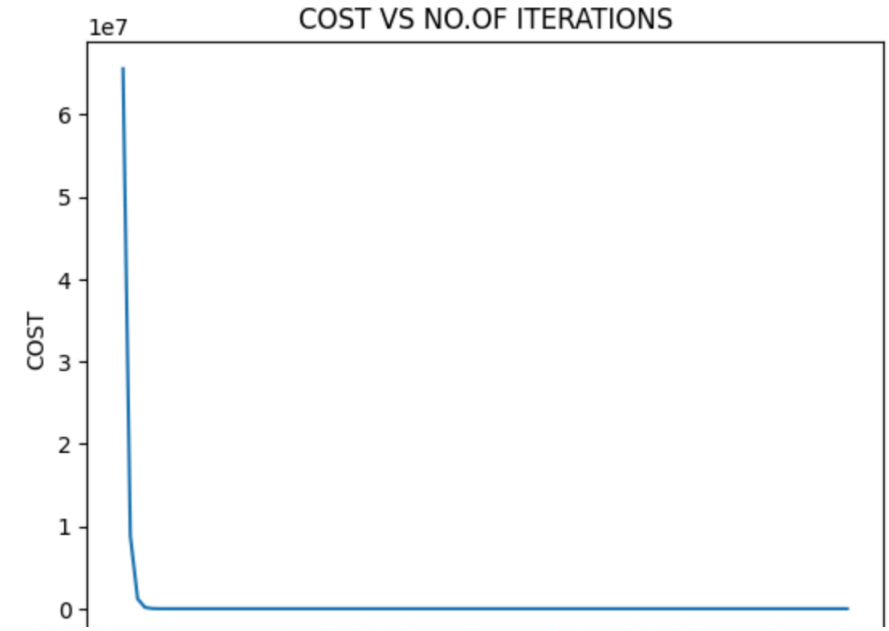
The training set had 50000 trainee examples each described by 20 features.Firstly, input and output numpy arrays were formed as **x_train** and **y_train** , hence we create the hyper parameters: weights and biases (w and b) of suitable sizes. The values of w and b are so as to give the most suitable fitting linear function. Thus we form the predicted array of the input data set so as to minimise the cost function or error, it being a two degree polynomial gives a unique minima.

Here, we use random values of learning rate and number of iterations and put it through the gradient decent function to have a simultaneous change in the values of the weights and biases so as the cost function gradually approaches

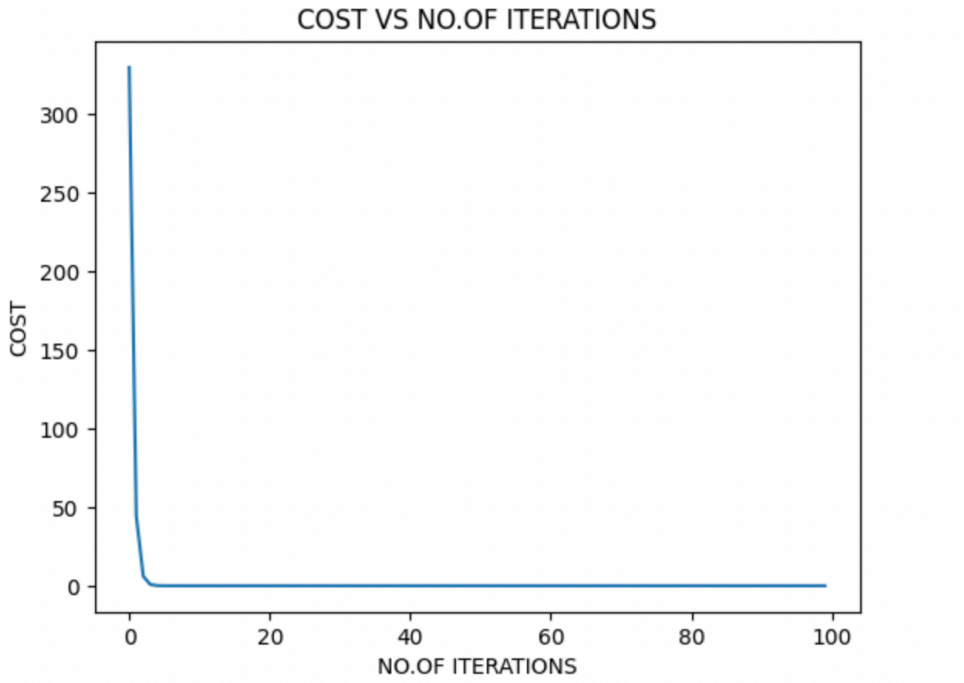
the minima with a significantly small value of alpha to prevent overshoot of the value of weights and biases. Finally learning rate=0.001

To check the model's efficacy, we train it on 80% training data set and thus use the obtained values of the weights and biases on the remaining 20% data set used as the cross-validation set.

```
0.005050994574172633
```




```
0.0050478845815012325
```



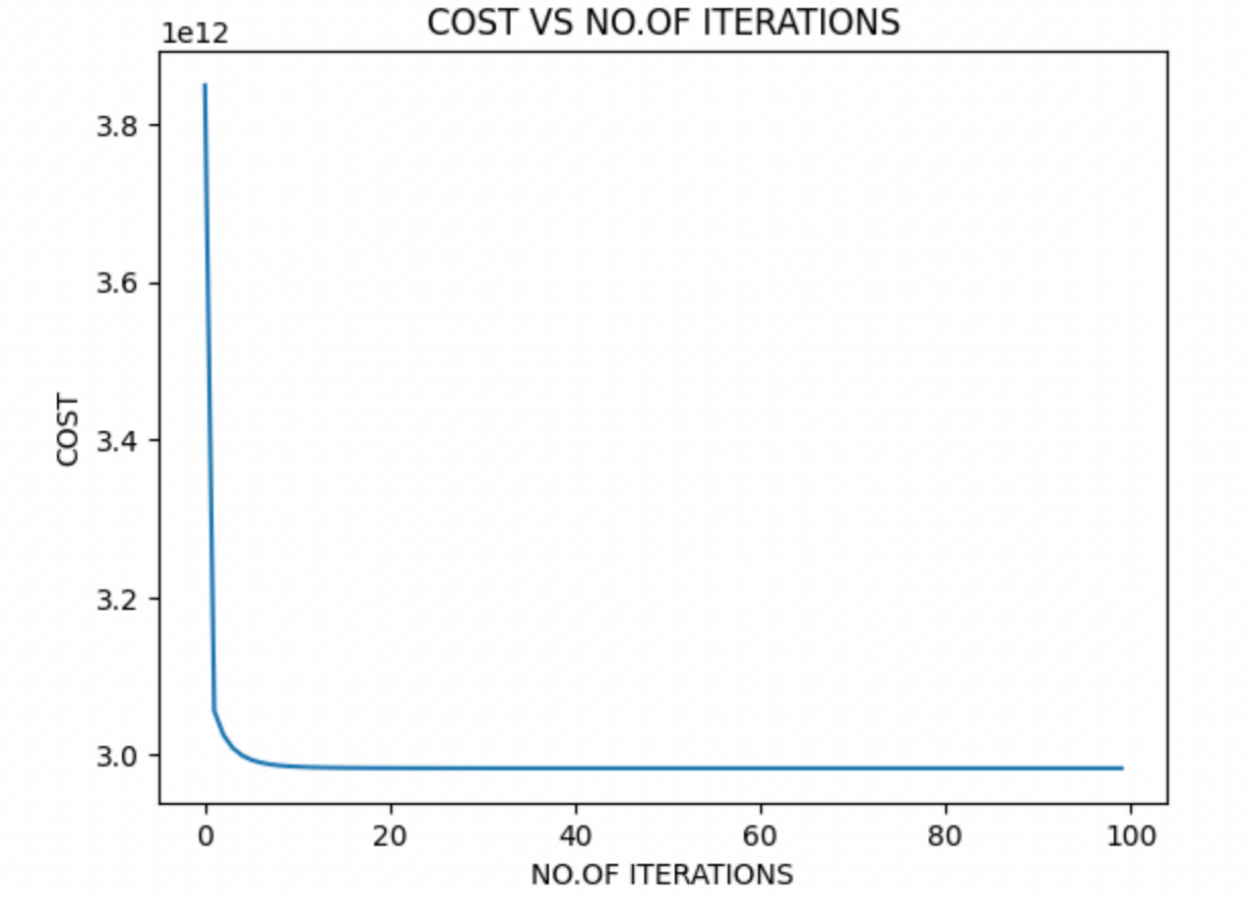

```
y_predcv=(np.matmul(w_cv,x_cv.T)).T+b_cv
print(R2score(y_predcv,y_cv))
```

```
99.99949809858576
```

Polynomial Regression:-

The training set had 50000 trainee examples with 3 features provided. Features are added using a recursive function so as to make such a n-degree polynomial which would minimise cost by giving a perfect fit.The new features are each a polynomial term of degree n of all possible combination of the three input features.

The value of n is decided after trying with certain values of learning rate and hence **5th** degree polynomial with **0.001** as the learning rate to offer minimum cost.With larger values of learning rate overshooting is observed.
Efficacy of the model is checked using test set and the cross validation sets.

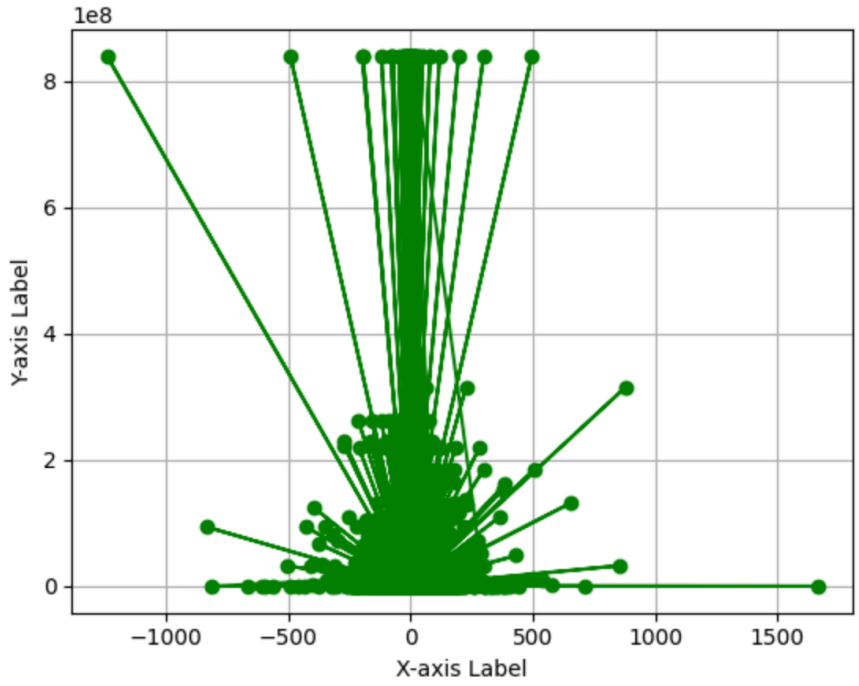



```
]: print(R2score(y_predcv,y_cv))
```

```
96.50524001287913
```

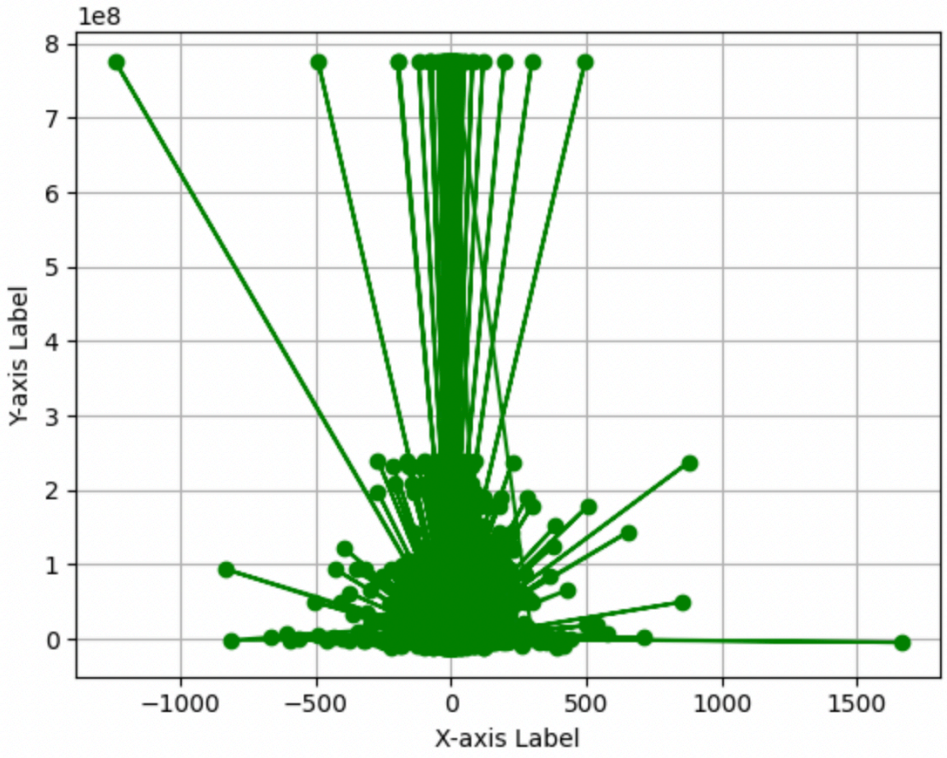
Plot of testing data with the model :

```
plt.plot(x_cv, y_cv, marker='o', color='g')
plt.xlabel('X-axis Label')
plt.ylabel('Y-axis Label')
plt.grid(True)
plt.show()
```



Plot of Predicted values of the above testing data:

```
plt.plot(x_cv, y_predcv, marker='o', color='g')
plt.xlabel('X-axis Label')
plt.ylabel('Y-axis Label')
plt.grid(True)
plt.show()
```



LOGISTIC REGRESSION:-

The given data set had 30000 trainee examples with 784 features given as pixels describing them. This shows that the problem involves classification of images.

The model makes use of **sigmoid activation** gives multiple minimal values of cost functions due to non-linearity.Thus, **binary cross-entropy***(Log based loss function)* is used to calculate the loss or cost value which is hence minimised to obtain the specific values for weights and biases at minimum loss value.

Methods of gradient and gradient decent calculations remain the same with changed cost function from the last two models.Here, the labels are “one-hot encoded” for smooth cost calculation.

“One vs All“ approach has been used for the classification task with varied values of learning rates and iterations are tried out.

The model showed very optimistic accuracy of **96.133%** for **learning rate=0.01 and 1000 iterations.**

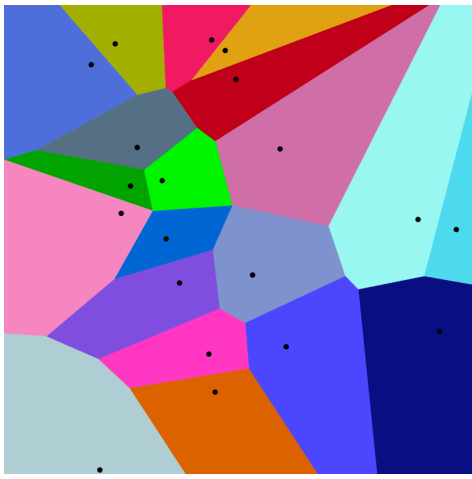
```
prob_check(y_check_test,y_cv)
```

```
array([96.13333333])
```

KNN(K-Nearest Neighbours):-

The given training set had 30000 trainee examples , with pixels of features, which implies pictures form the training set.
In this model, a particular point in vector space is either assigned a value or classified according to the ‘K’ number of points closest to it. It’s a lazy learner for this model doesn’t learn at all while training.

The mathematical concept of **Voronoi diagram** is implemented by this model for the given case of classification type of problem.
*A **Voronoi diagram** consists of a set of adjacent straight-sided polygons that divide the plane into regions of closeness to a given node.*



Several values of ‘K’ are tried upon to predict the values and hence accuracy was measured , the most modest accuracy was observed for K= 3 with a predicting accuracy of 99.12 .

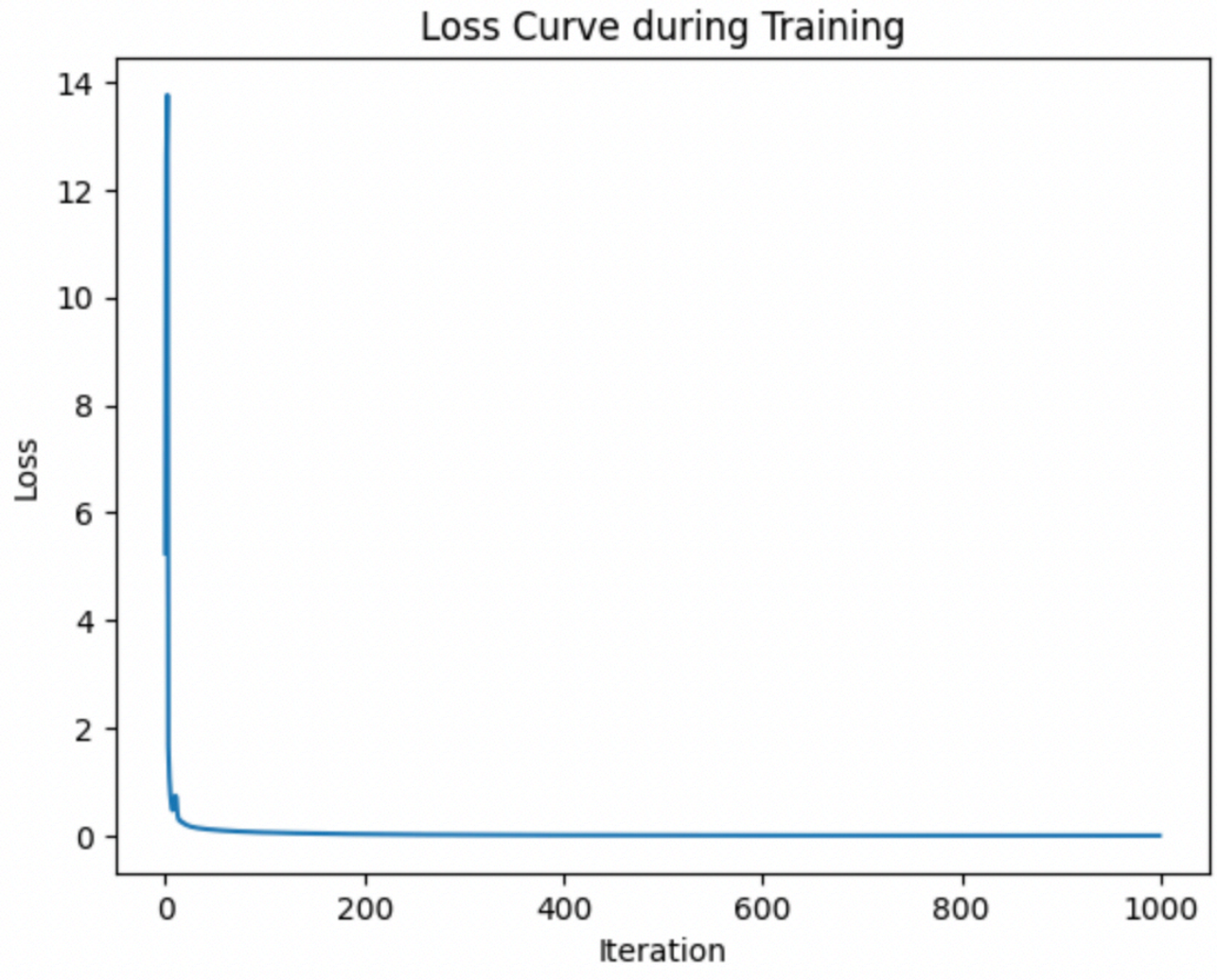
NNN(N layered-Neural_Network):-

The training set has 30000 trainee examples.The model makes use of one hidden layer with standard input and output layers. The method of classification applied is **softmax** with **relu** activation, for greater accuracy and faster execution of the model.

The loss function is calculated using **binary cross entropy**. Here, B**ack propagation** is applied to obtain gradient decent which is a much faster method to obtain the most appropriate values of the parameters (w,b) by using **chain rule** as the fundamental working processing. Back propagation is a much faster and efficient way to calculate gradient decent rather the ones applied in the previous models.

The perfect value of learning rate is obtained so as to minimise the value of loss function by hit and trial with progressive iterations. For learning rate = 0.01 , accuracy on training and cross-validation sets are 100% and **98.37%.**

```
Iteration 0, Loss: 5.249254335717199
Iteration 100, Loss: 0.06117374216127106
Iteration 200, Loss: 0.029413016201858107
Iteration 300, Loss: 0.016758144362461123
Iteration 400, Loss: 0.010607938132671996
Iteration 500, Loss: 0.0073561698889974955
Iteration 600, Loss: 0.005476509550506684
Iteration 700, Loss: 0.00429007755789554
Iteration 800, Loss: 0.0034873054985528236
Iteration 900, Loss: 0.002915258806171216
Training Accuracy: 100.0%
Cross-Validation Accuracy: 98.37777777777778%
```



K-means Clustering:-

This model assumes a value of K, which specifies number of clusters to be formed for the entire data set. Here a point in space is assigned to a cluster with minimum distance from the cluster centroid.
After clusters have been assigned, new cluster centroids are reinitialised with the mean value of all the points present in that cluster and the process continues for the specified number of iterations or the terminal condition where on reinitialisation the clusters centroids do not change
Several values are tried for K, showing changes in cost as well as accuracy of the model.It is observed **"Elbow-Curve"** is not purely followed but on an irregular interval elbow curve is obeyed.
Value of 'K' is chosen so as to maintain a low cost with minor variations in accuracy.
For K= 3 with cost as 143.84 , has been considered after tuning the model .