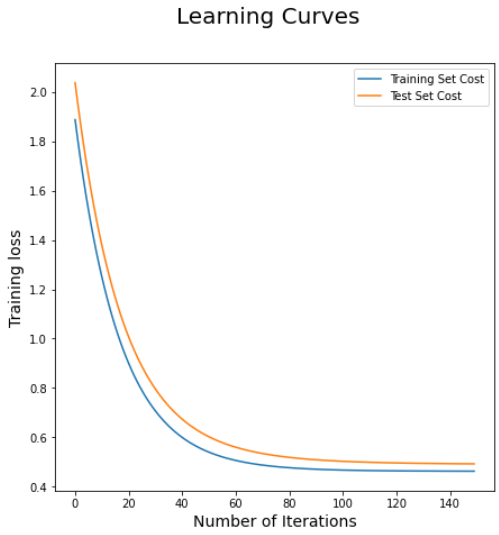
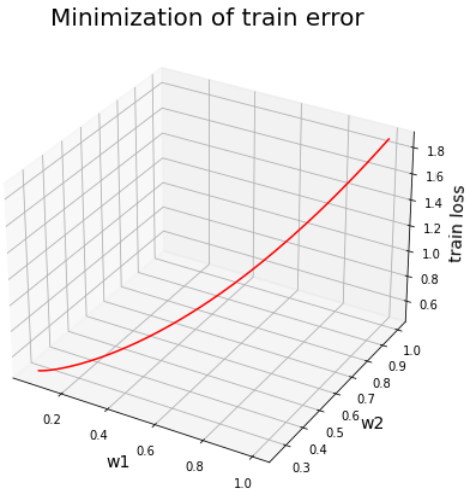
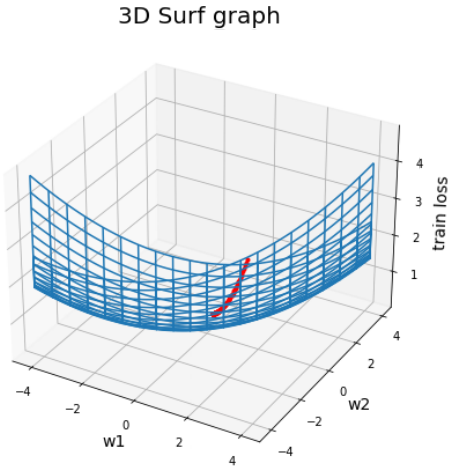
Assignment 1 (Linear regression, regularization, logistic regression, multi-class classification, probabilistic classifiers)

Q1.Implement the linear regression algorithm to estimate the weight parameters. You can use batch gradient descent algorithm for the implementation. (a) Plot the cost function vs the number of iterations. (b) Plot the cost function (J) vs w1 and w2 in a contour or 3D surf graph (w= [w0 w1 w2]). (c) Evaluate the mean square error between the actual test output and predicted test output. Please use the files such as Training feature matrix, training output vector, test feature matrix, test output vector for this question (Use for or while loop for the implementation of linear regression).

a)



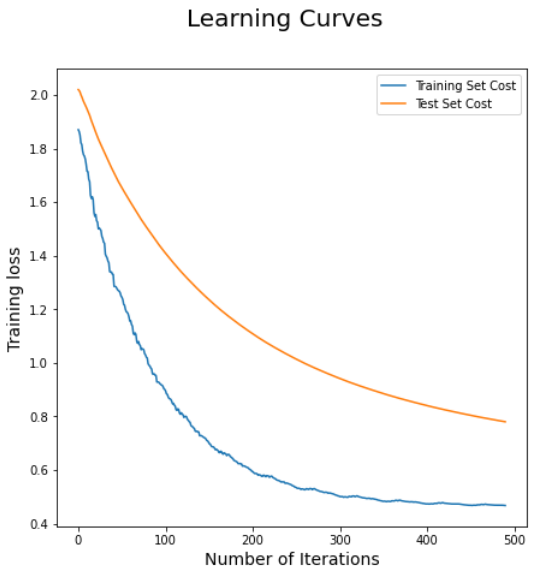


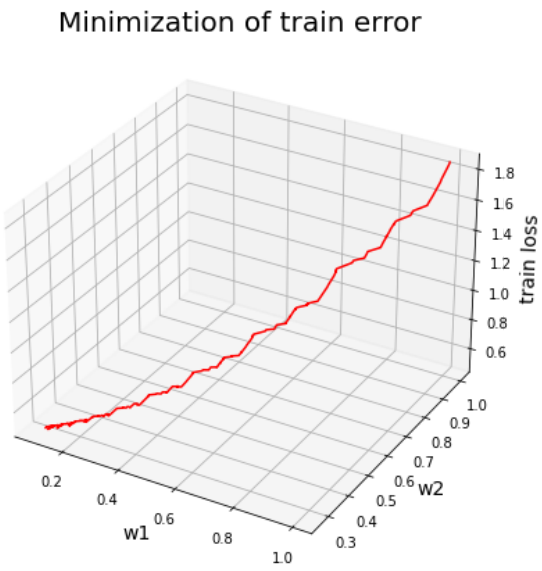


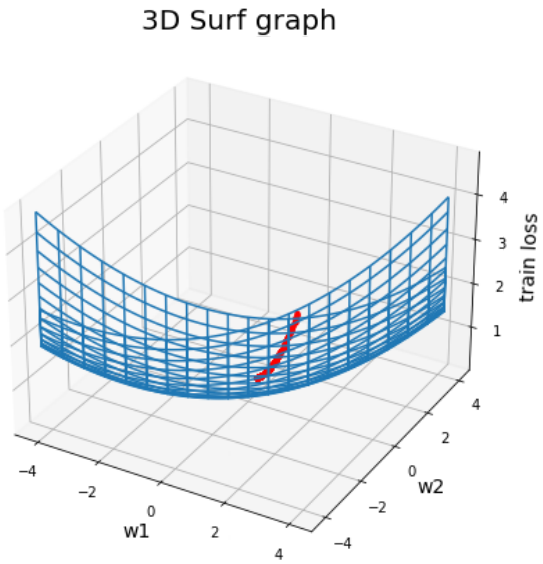
c)mean square error between the actual test output and predicted test output is 0.4925

2)Implement mini-batch and stochastic gradient descent algorithms for the linear regression problem in question number 1. (a) Plot the cost function vs the number of iterations. (b) Plot the cost function vs w1 and w2. (c) Evaluate the mean square error between the actual test output and predicted test output. Please use the same data files as in question number 1 (Use for or while loop for the implementation).

a)Mini-Batch Gradient Descent-

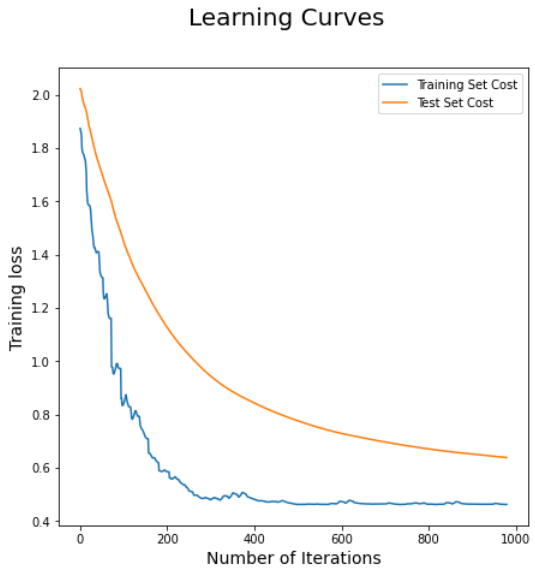


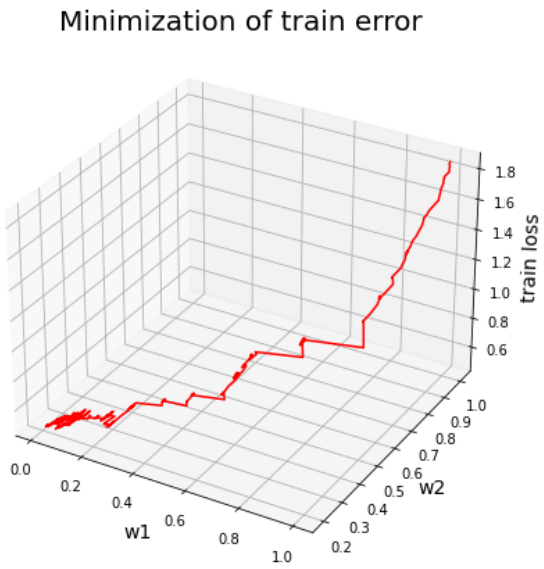


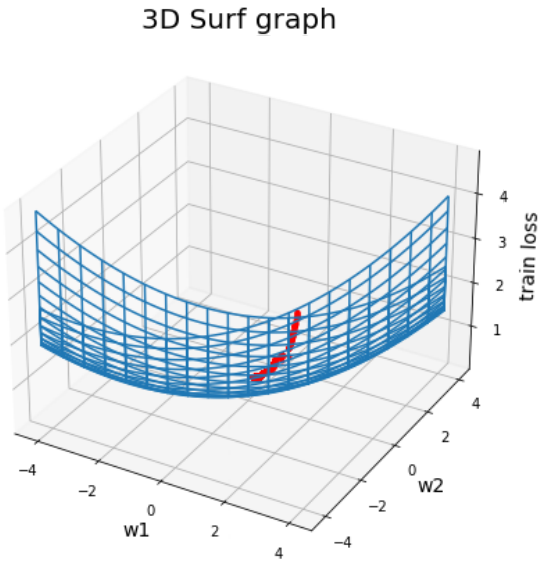


mean square error between the actual test output and predicted test output is 0.78029

b)Stochastic Gradient descent



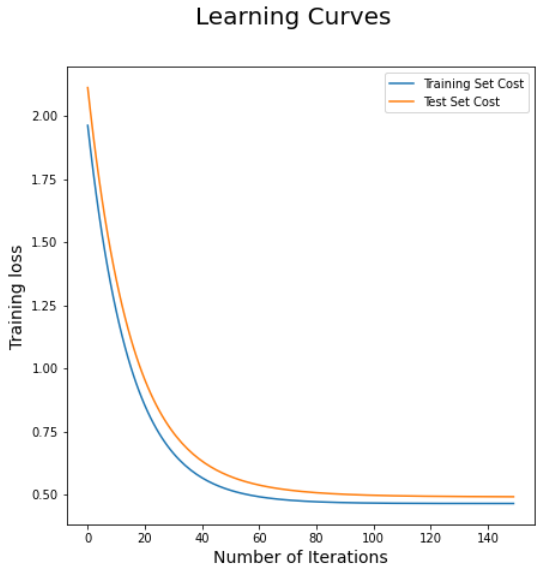


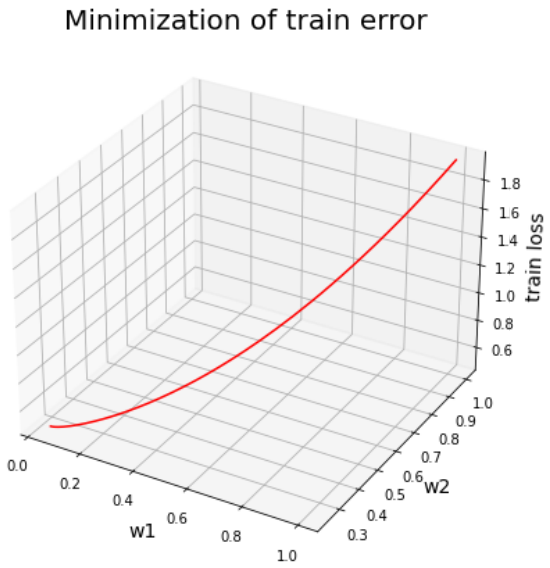


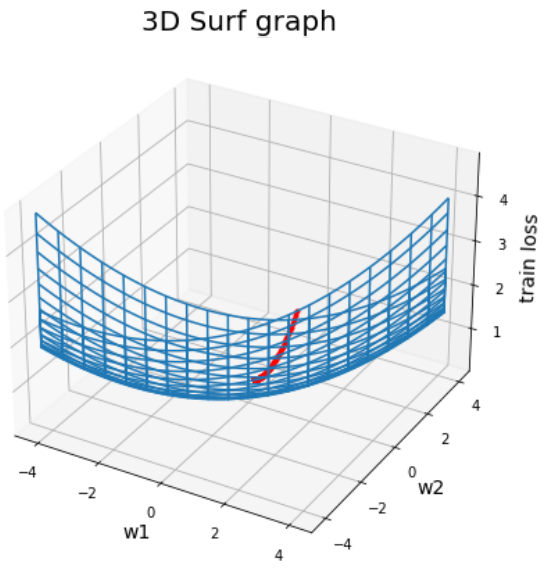
mean square error between the actual test output and predicted test output is 0.638841

3)Implement the ridge regression problem by considering batch, mini-batch and stochastic gradient descent algorithms. (a) Plot the cost function vs the number of iterations for all three cases. (b) Plot the cost function (J) vs w1 and w2 in a contour or 3D surf graph for all three cases. (c) Evaluate the mean square error between the actual test output and predicted test output. Please use the same data files as in question number 1 (Use for or while loop for the implementation).

a)Ridge Regression with Batch Gradient Descent

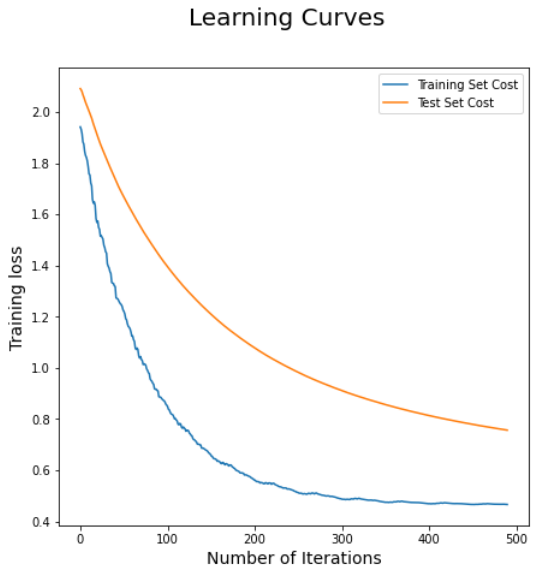


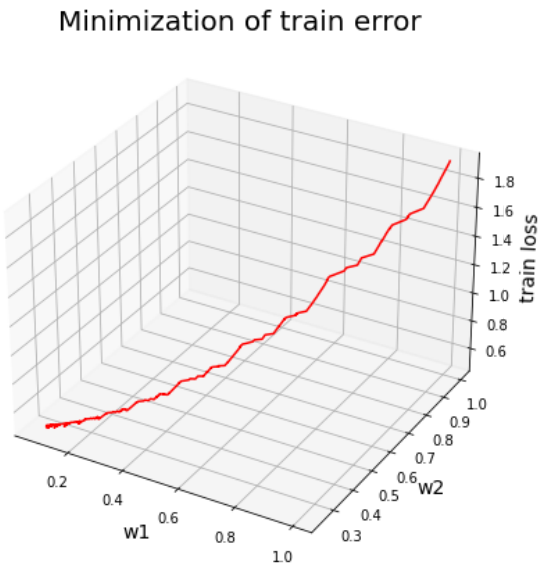


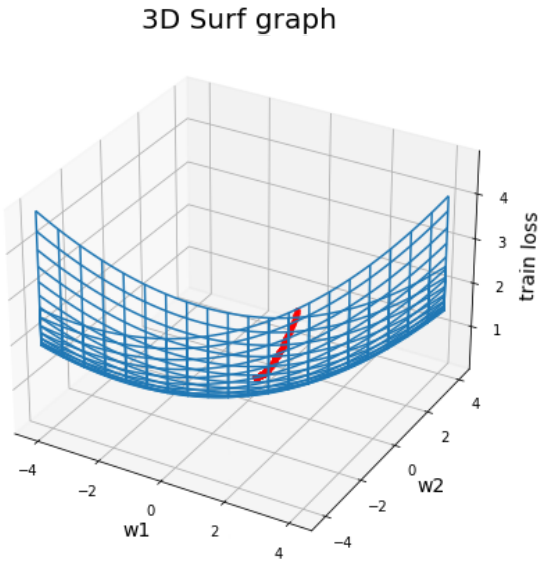


mean square error between the actual test output and predicted test output is 0.490866

b)Ridge regression with mini batch gradient descent

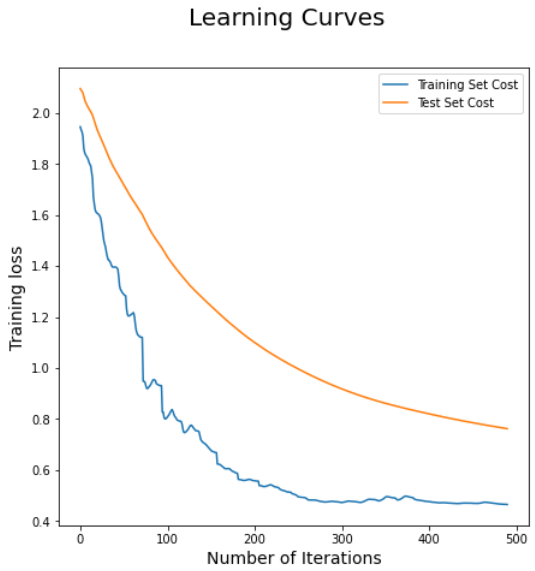


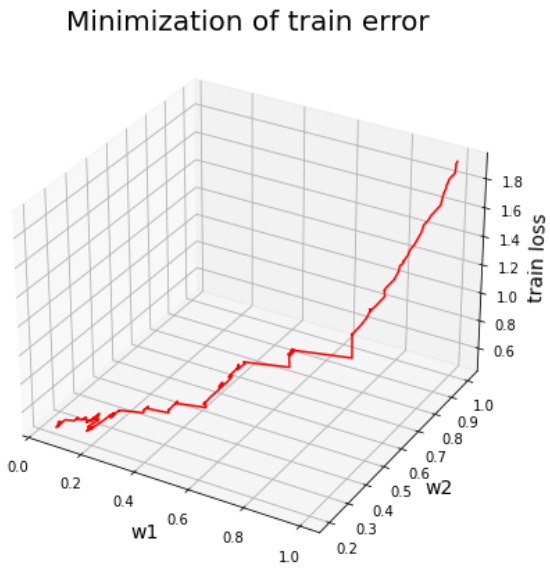


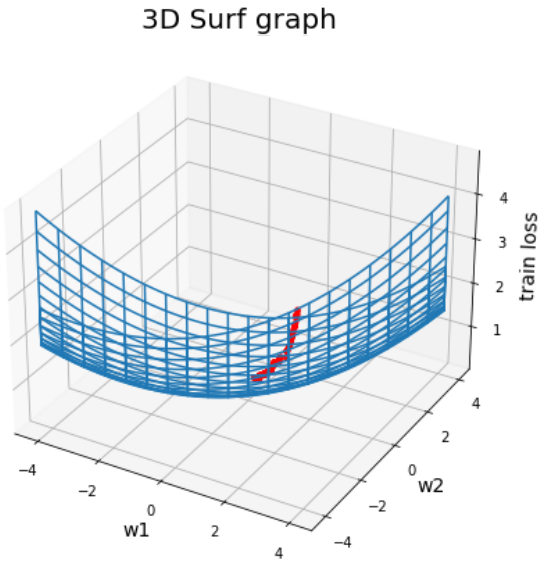


mean square error between the actual test output and predicted test output is 0.756503

c)Ridge regression with stochastic gradient descent

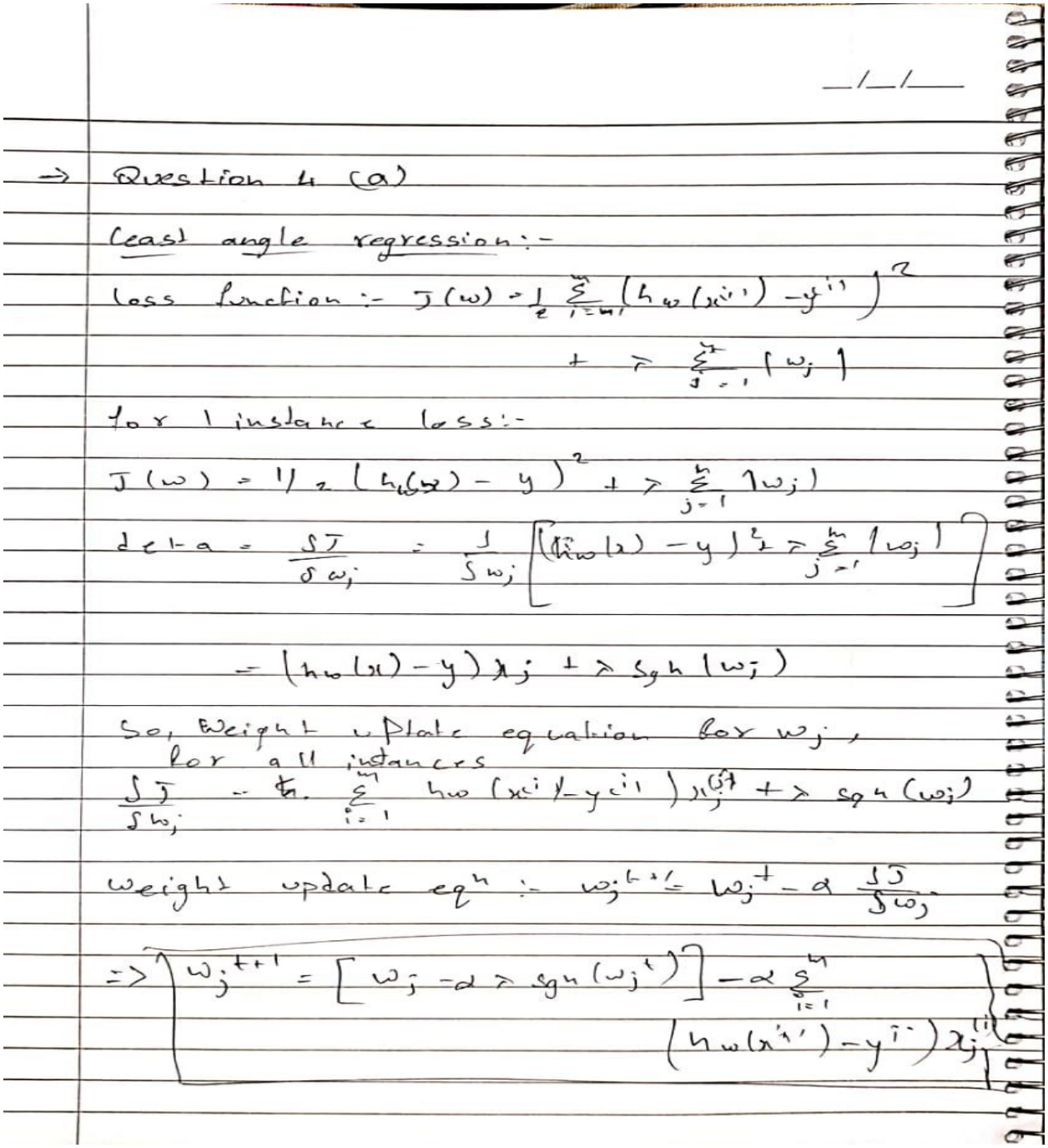






mean square error between the actual test output and predicted test output is 0.761930

4. Implement Least angle regression to estimate the weight parameters for the feature matrix (X) and the class label vector (y) by considering batch, mini-batch and stochastic gradient descent-based algorithms. (a) Plot the cost function vs the number of iterations for all three cases. (b) Plot the cost function (J) vs w1 and w2 in a contour or 3D surf graph for all three cases. (c) Evaluate the mean square error between the actual test output and predicted test output. Please use the same data files as in question number 1 (Use for or while loop for the implementation).



→ Question 4 (a)

Least angle regression :-

Loss function :- $J(w) = \frac{1}{2} \sum_{i=1}^{m} \left(h_w(x^{(i)}) - y^{(i)} \right)^2 + \lambda \sum_{j=1}^{n} |w_j|$

for 1 instance loss :-

$J(w) = 1/2 \left(h_w(x) - y \right)^2 + \lambda \sum_{j=1}^{n} |w_j|$

$delta = \frac{\partial J}{\partial w_j} = \frac{1}{\partial w_j} \left[(h_w(x) - y)^2 + \lambda \sum_{j=1}^{n} |w_j| \right]$

$= (h_w(x) - y) x_j + \lambda \, sgn(w_j)$

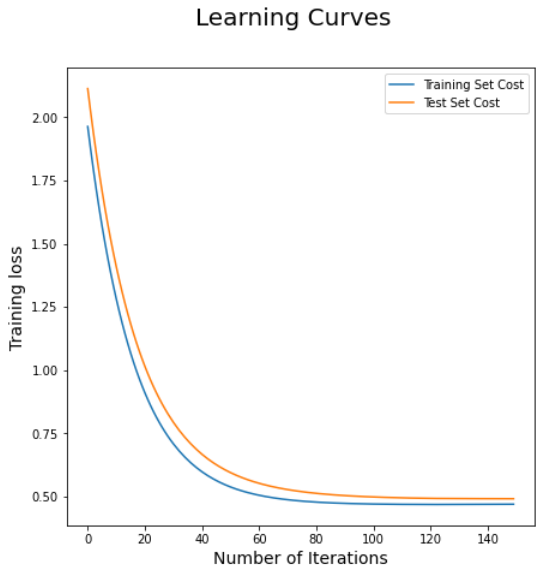
So, Weight update equation for w_j, for all instances

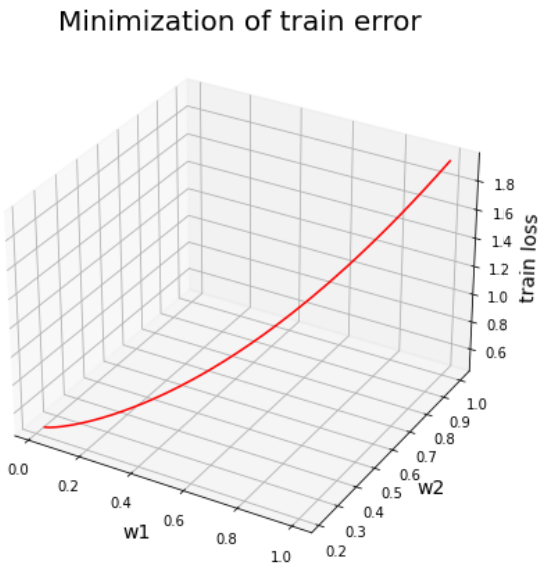
$\frac{\partial J}{\partial w_j} = \sum_{i=1}^{m} h_w(x^{(i)} - y^{(i)}) x_j^{(i)} + \lambda \, sgn(w_j)$

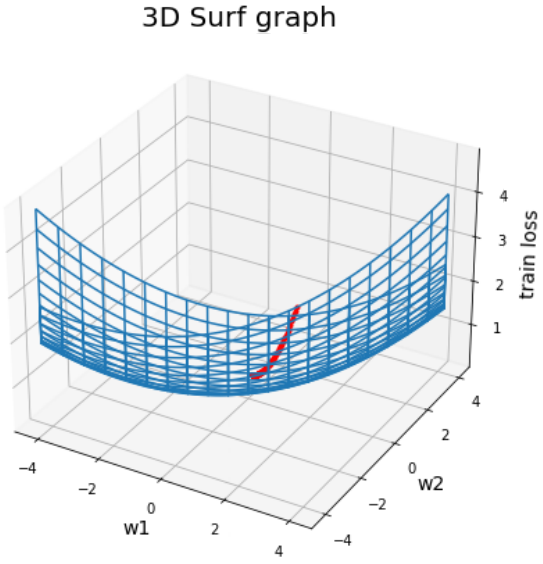
Weight update eqn : $w_j^{t+1} = w_j^t - \alpha \frac{\partial J}{\partial w_j}$

$$\Rightarrow \boxed{w_j^{t+1} = \left[w_j - \alpha \lambda \, sgn(w_j^t) \right] - \alpha \sum_{i=1}^{m} \left(h_w(x^{(i)}) - y^{(i)} \right) x_j^{(i)}}$$

a)Least angle regression with batch gradient descent

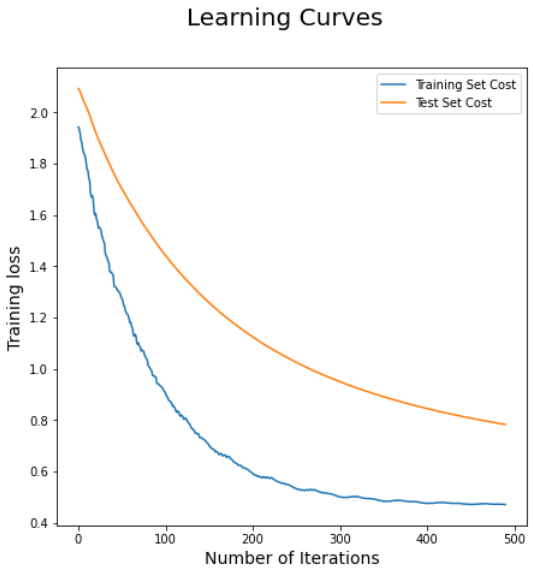


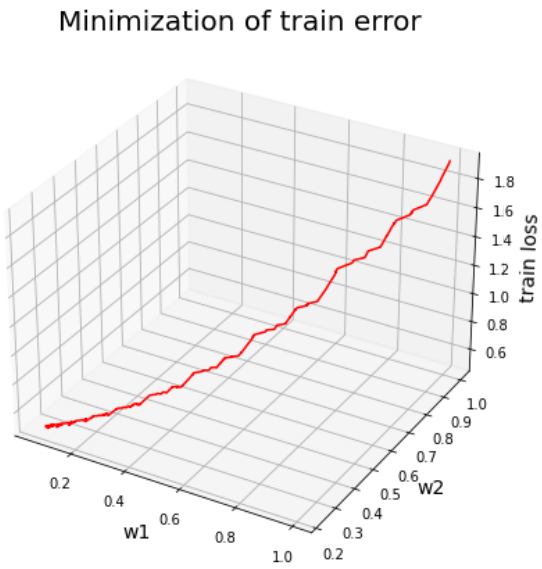


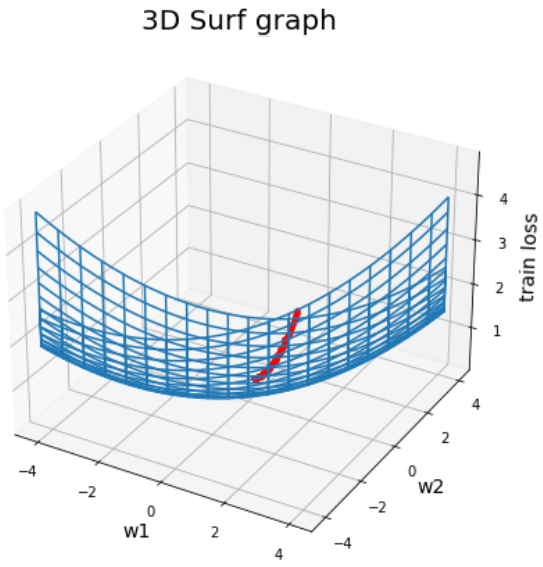


mean square error between the actual test output and predicted test output is 0.492350

b)Least angle factor with mini batch gradient descent

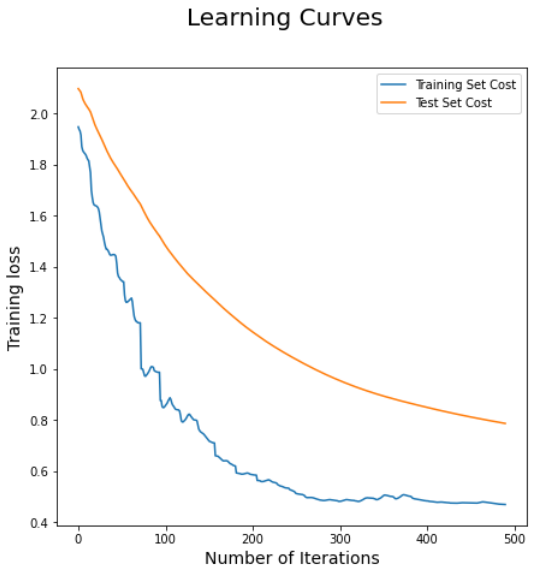


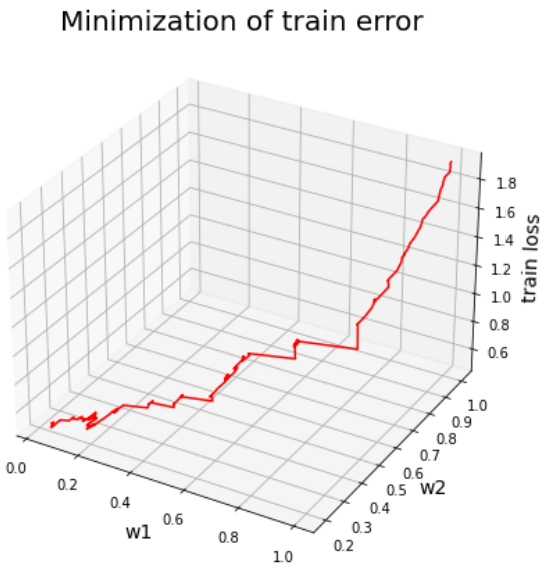


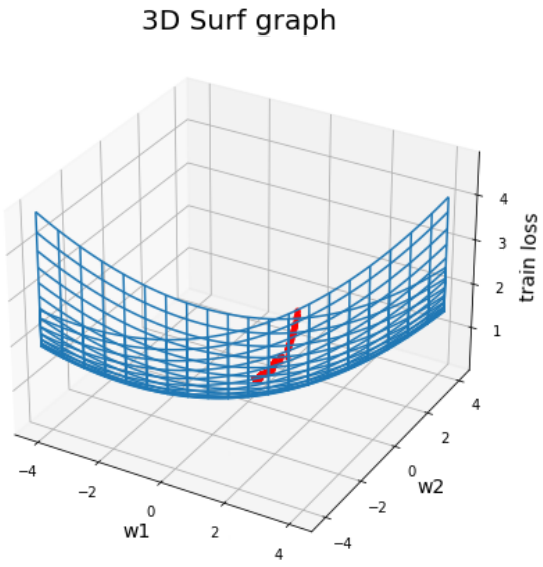


mean square error between the actual test output and predicted test output is 0.787378

c)least angle factor with stochastic regression







mean square error between the actual test output and predicted test output is 0.7873786

5)Implement the vectorized linear regression, ridge regression and least angle regression problems to evaluate the weight parameters for question number 1. Compare the weight parameters of vectorization-based methods with the weights obtained using batch, mini-batch and stochastic gradient descent-based algorithms. Please use the same data files as in question number 1. Evaluate the mean square error between the actual test output and predicted test output.

a)Weight values of vectorized linear regression [[-2.02962647e-16][4.30528649e-02]
[2.58988001e-01]]
b) Weight values of vectorized ridge regression
[[-2.02962647e-16][4.30528649e-02] [2.58988001e-01]]

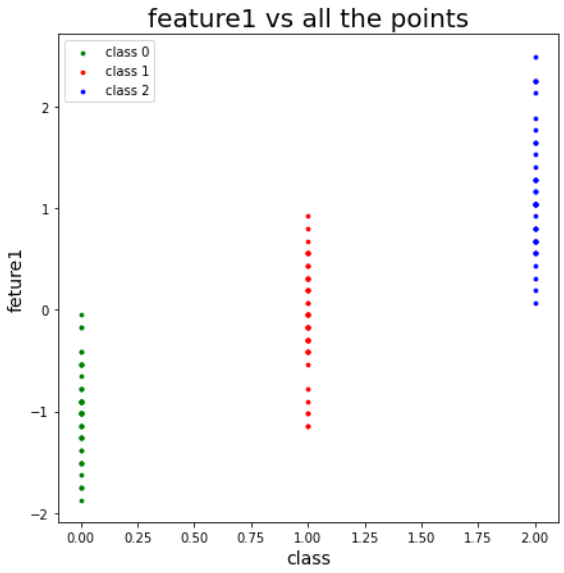
c) Weight values using batch gradient descent [[0.02242293][0.05658221][0.25880947]]

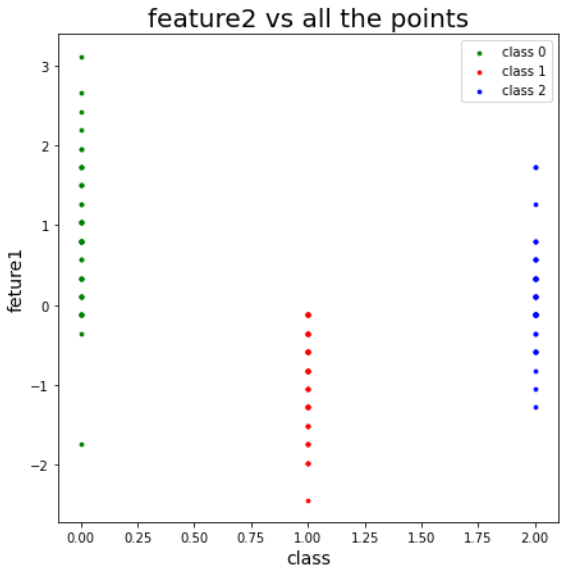
d)Weight values using mini batch gradient descent
[0.08518338 0.09338971 0.2791136]

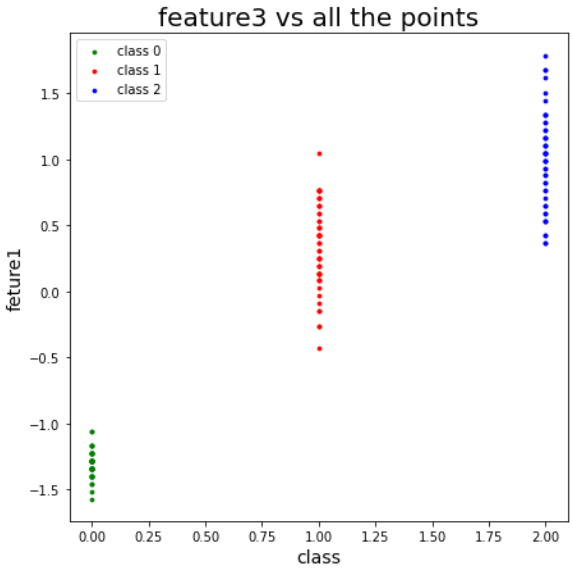
e) weight values of stochastic gradient descent
[-0.00034923 0.01935004 0.25028439]

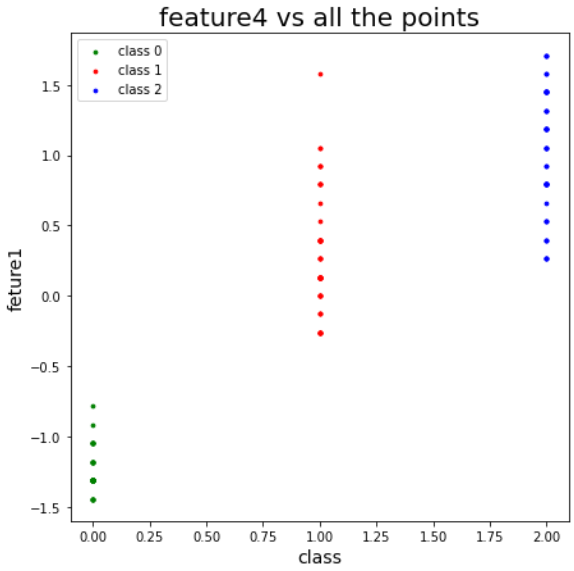
6)Implement K-means clustering based unsupervised learning algorithm for the dataset (“data2.xlsx”). Plot the estimated class labels vs features. Use the number of clusters as K=3.

A)









Question - 13

I've learnt the following things while solving assignment 1:

- Building basic ML models like, linear regression, logistic regression from scratch.
- Using L1 and L2 normalisation and demonstrating smoother minimisation of cost function.
- Implementing Vectorized linear regression.
- Implementing k-means clustering from scratch
- Using k-fold cross validation to validate the model performance.
- Implementing one-Vs-one and one-Vs-all multi class classification.
- Using probabilistic models like maximum a posteriori (MAP) and Maximum likelihood (ML).

● Using scientific computing libraries like NumPy,matplotlib and SciPy.

All the required plots are pasted above for the codes of all the questions and results, go to the following link
Code files(.ipynb files)