Project 1

Q1.

A and B Part

Program in: Python 2.7

Program Name: ML\_1(Training)

ML\_1(Test)

I have written program in python to get input from user for degree/order of equation as required. I have provided alpha as 0.03 (r in program) which can be changed as per requirement. It plots the curves in 3d space. The color bar in the graph represents value of prediction by the program. The other two axes are x1 and x2 values.

I have used numpy library just to take input and especially to do matrix calculations.

C Part

For high learning rate polynomial of order 1 is best and fittest. For better curves the learning rate has to be decreased. Higher the order, lower will be the learning rate.

For ideal and steady results Polynomial of degree 2 is best.

Some of the observations are:

Enter order: 1

Enter alpha: 0.03

Cost: 0.241336930909

Error = -1.7318730805e-06

Enter order: 2

Enter alpha: 0.03

Cost: 3177.57922665

Error = -76325.1422974

Enter order: 3

Enter alpha: 0.03

Cost: 614552.150934

Error = -77266530.7391

Enter order: 4

Enter alpha: 0.03

Overflow Error!

Enter order: 4

Enter alpha: 0.0003

Cost: 64235779.9137

Error = -597213792.572

Q2

Program Name: ML\_P2

B part

Logistic regression yields well than the other two because it optimizes for one of the output (value =1).Moreover the function never reaches 1 or 0 so there is always scope of optimization based on the alpha value. The curve has space for future data too so chances of overfitting are less in this case on optimization.

KNN predicts on the basis of Euclidean distance between the data points. Similarly naïve baiyes too predicts data on the basis of probabilities in training data given. Both this algorithms cannot optimize as the logistic regression does.