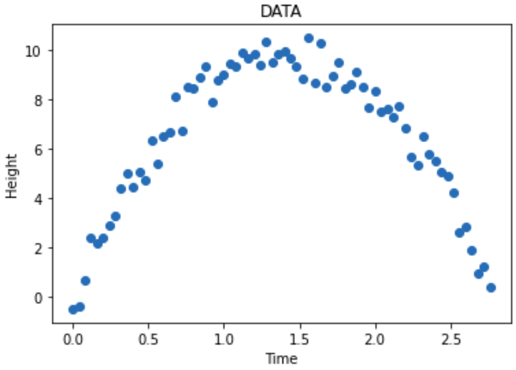
ASSIGNMENT-2

1 Question 1

1.1 Question 1.1(Plot of training data) :



- Since, it is a trajectory of freely moving body in air, we can say that it of the model, h = $w_0 + w_1 t + w_2 t^2$. We can also say that by seeing the plot(as it is so much similar to a parabola)

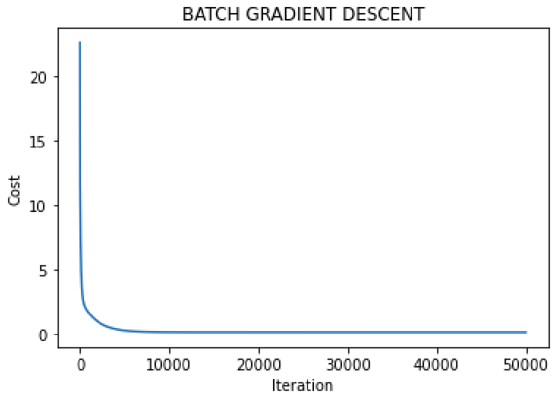
1.2 Question 1.2

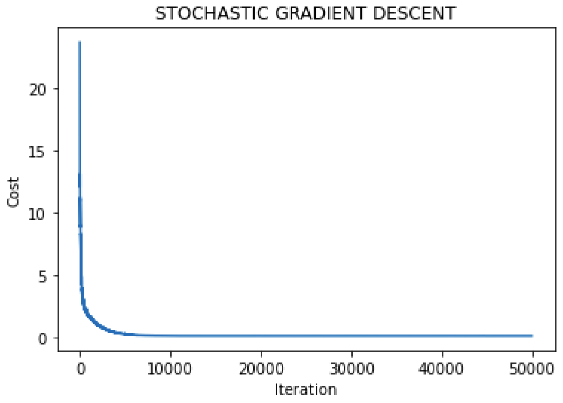
1.2.1 Stochastic gradient descent

```
def SgradientDescent(self, x, y, w1, alpha, iters):
    m =  len(x)
    p = len(w1)
    w = [0.0]*p
    J_history_s = []   # Use a python list to save cost in every iteration
    n= int(m/3)
    n = 100 if n>100 else n
    for i in range(iters):
      r = np.random.randint(m)
      for k in range(p):
        J=0
        for i in range(r,r+n):
          temp = 0
          for j in range(p):
            temp+= w[j]*x[(i)%m][j]
          temp-= y[(i)%m]
          J+= temp*x[(i)%m][k]
        w[k] -= alpha*J/n
      J_history_s.append(self.computeCost(x, y, w))
    return w, J_history_s
```

- **Weights :** [-0.14153782259205086, 14.20959532819735, -5.048589701805952]
  Iterations performed : 50000, alpha : 0.01

1.2.2 Plots of Cost vs Iterations





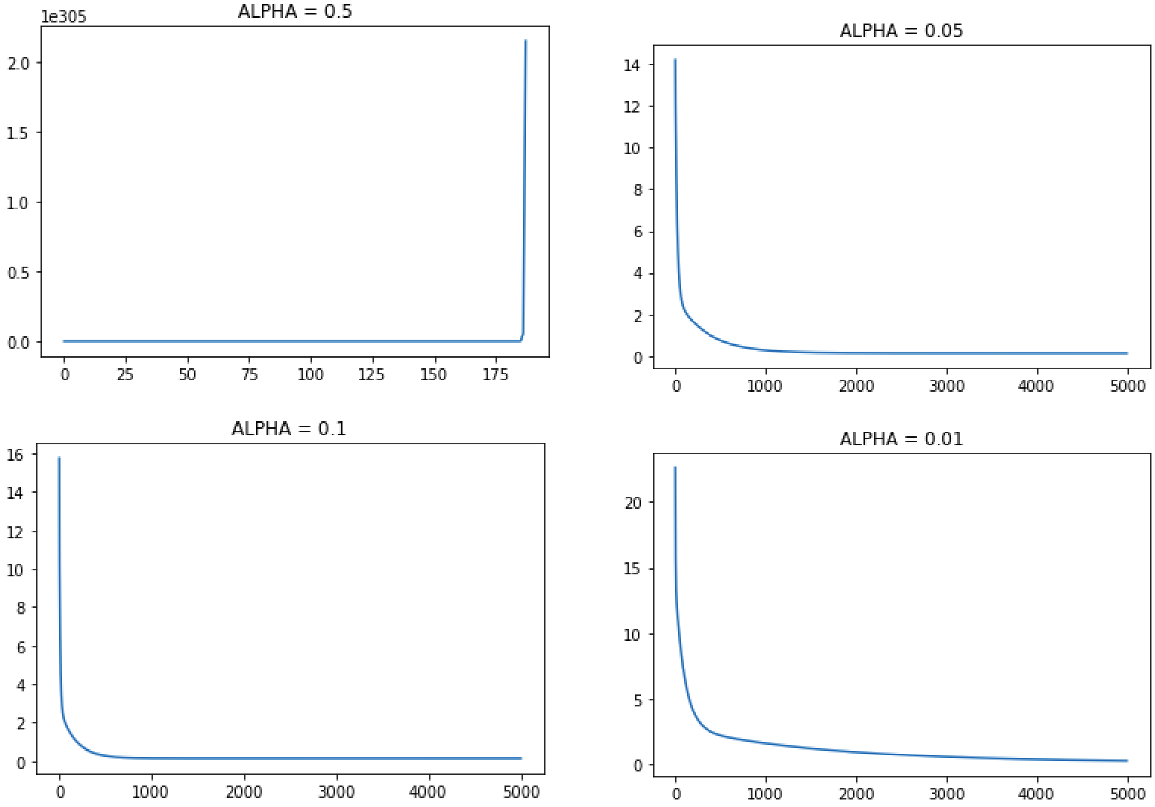
1.2.3 Analysis of the above plots

- There exists a lot of disturbance(rapid zig-zag movement) at the starting in the stochastic Gradient, where as Batch Gradient is very smooth and uniform. This difference is due to random selection of a part of data for each iteration in optimising the vector of weights. The absence of disturbance at higher number of iterations represents that the model is very good, as every random data used in each iteration have very less or no deviation from the suggested model.

- We also see a faster and better results in Stochastic gradient when sufficiently high number of iterations are performed.

1.3 Question 1.3

1.3.1 Plots of Cost(Y-axis) vs Iterations for different Alphas(X-axis)


ALPHA = 0.5

ALPHA = 0.05

ALPHA = 0.1

ALPHA = 0.01

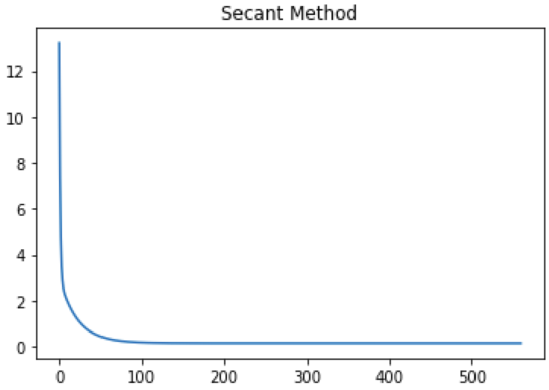
1.3.2 Analysis of Above plots

- When we take alpha as 0.5, the cost function increases with each iteration and explodes towards infinity. This is because of its large value. Later, we see that the output(weights) contains a value of absolute value around 0.135. As part of my observation, for every alpha below 0.1343(including it), the cost function is decreasing with the increase in interactions, where as it increases along with iterations for the value of alpha starting around 0.1344.

- The rate of decrease in cost function increases with increase in alpha, but we get more accurate output with lower alpha value.

1.4 Question 1.4

1.4.1 Cost(Y-axis) vs Iterations(X-axis)

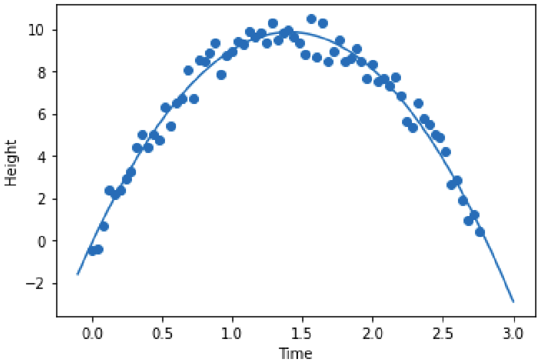


1.4.2 Analysis

- In this method, we get the optimised value of weights by performing very less number of iterations because of calculation of optimised value of alpha at each iteration. This is method is better to use in cases where we are unaware of good range of alpha to take(as we saw before that cost function will increase for bad value of alpha).

1.5 Plot of Trained Model and Training data

Weights : [-0.13575299423570797, 14.200395903323264, -5.041700050986729]
Trained data : scattered, Trained Model : curve



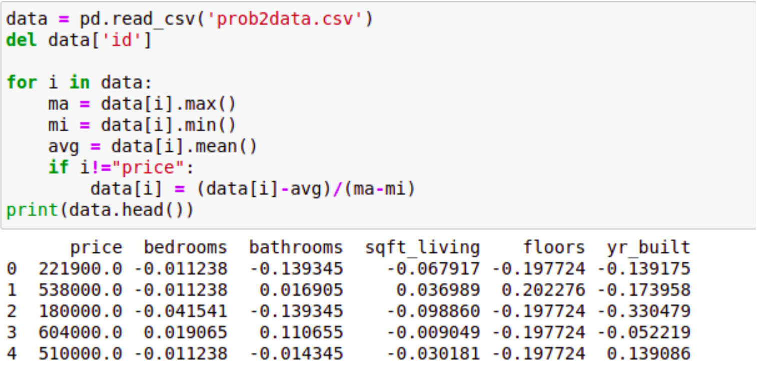
2 Question 2

2.1 Question 2.1

2.1.1 Code

```
data = pd.read_csv('prob2data.csv')
del data['id']

for i in data:
    ma = data[i].max()
    mi = data[i].min()
    avg = data[i].mean()
    if i!="price":
        data[i] = (data[i]-avg)/(ma-mi)
```



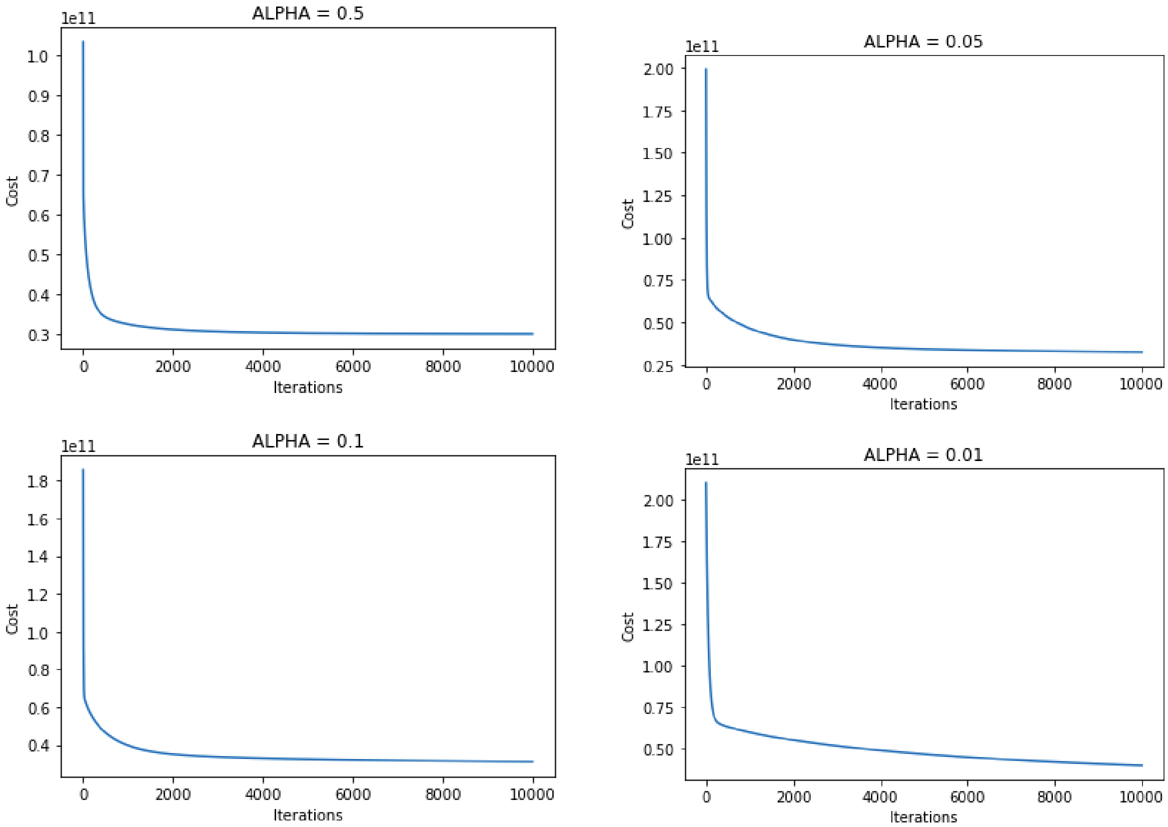
```
data = pd.read_csv('prob2data.csv')
del data['id']

for i in data:
    ma = data[i].max()
    mi = data[i].min()
    avg = data[i].mean()
    if i!="price":
        data[i] = (data[i]-avg)/(ma-mi)
print(data.head())
```

```
      price  bedrooms  bathrooms  sqft_living    floors  yr_built
0  221900.0 -0.011238  -0.139345    -0.067917 -0.197724 -0.139175
1  538000.0 -0.011238   0.016905     0.036989  0.202276 -0.173958
2  180000.0 -0.041541  -0.139345    -0.098860 -0.197724 -0.330479
3  604000.0  0.019065   0.110655    -0.009049 -0.197724 -0.052219
4  510000.0 -0.011238  -0.014345    -0.030181 -0.197724  0.139086
```

2.2 Question 2.4

2.2.1 Cost vs Iterations



2.2.2 Analysis

- Since, the prices are high, Cost function range is high.
- It is clear that graphs with lesser value of alpha takes more iterations to attain saturate values of weight.
- We can see that saturate value of cost function is little far from zero, showing that there could be a better model for this data, or that there is little more randomness(or absence of other features) in the data compared to question 1.