Alperen ÜGÜŞ
150160158

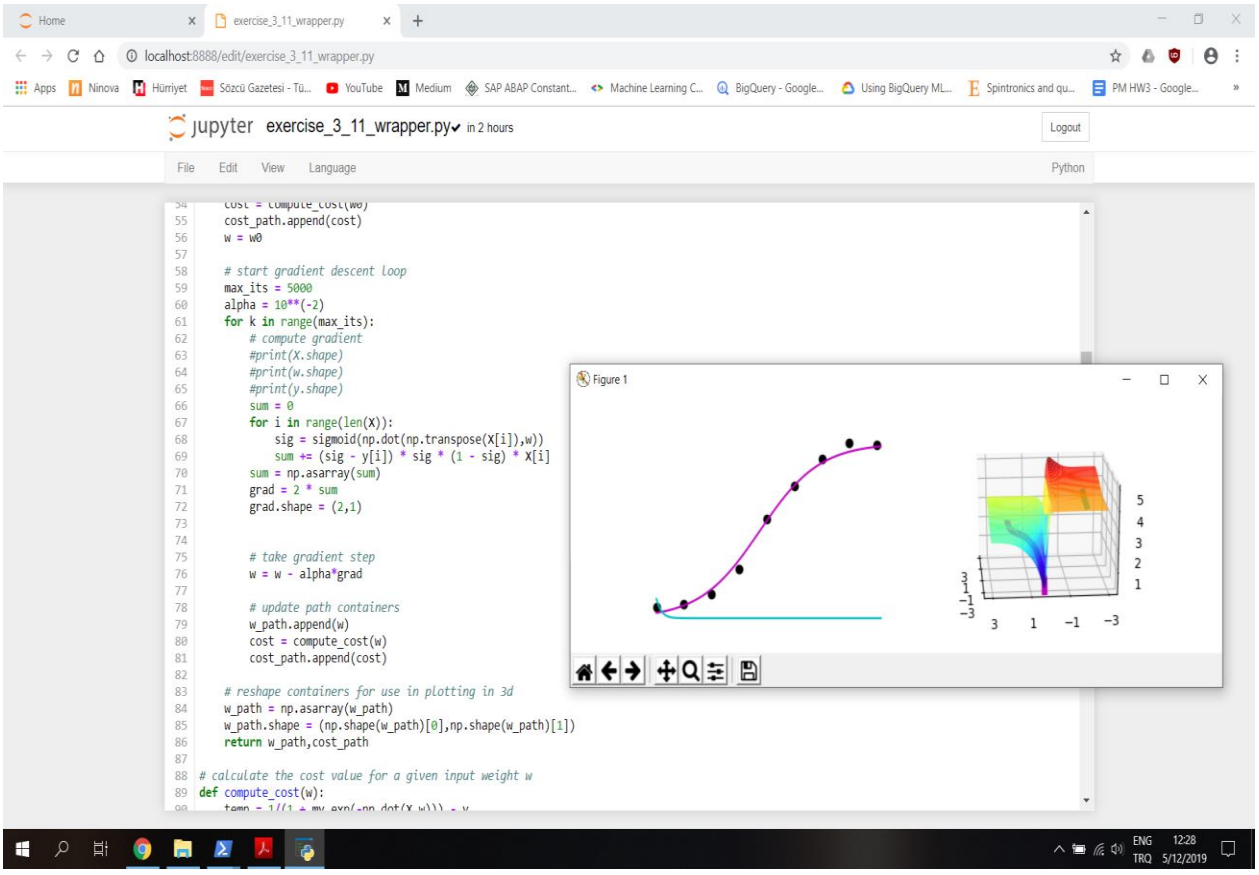
Learning from Data

Assignment 2

Question 1:

The Added/Changed Code Statement	Explanation
def sigmoid(x): return 1 / (1 + math.exp(-x))	This is a function that returns the sigmoidal value of a parameter passed to it.

sum = 0
for i in range(len(X)):
  sig = sigmoid(np.dot(np.transpose(X[i]),w))
  sum += (sig - y[i]) * sig * (1 - sig) * X[i]
sum = np.asarray(sum)
grad = 2 * sum
grad.shape = (2,1)
	This is where the gradient is calculated by using the logistic regression gradient descent calculation formula. For each subarrays of X and y, calculate gradient, make the result a numpy array, reshape gradient and use it for the iterations.



Alperen ÜGÜŞ
150160158

Question 2:

The Added/Changed Code Statement	Explanation
def sigmoid(x): return 1 / (1 + math.exp(-x))	This is a function that returns the sigmoidal value of a parameter passed to it.

sum = 0
for i in range(len(X)):
  sig = sigmoid(np.dot(np.transpose(X[i]),w))
  sum += (sig - y[i]) * sig * (1 - sig) * X[i]
 	
ar = []
ar.append(0)
ar.append(w[1][0])
ar = np.asarray(ar)
ar = np.reshape(ar, (2,1))
sum = np.reshape(sum, (2,1))
# regularize it 	
sum += 2 * lam * ar
	This is where the gradient is calculated by using the logistic regression gradient descent calculation formula. The difference between the above one is the regularization term.