

## HOMWORK1 Report

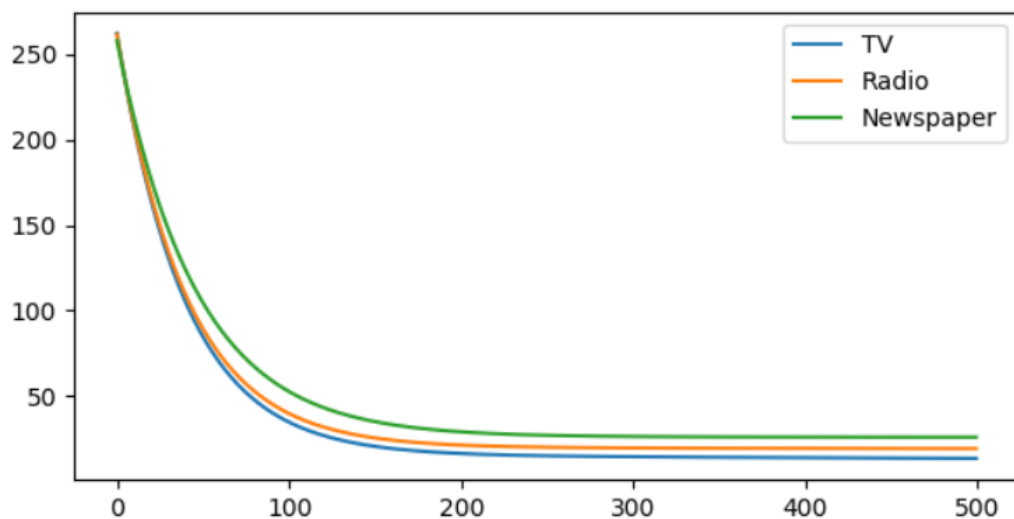
1. You have to report the  $\theta$  parameters in step 3 when you are using TV feature.

Answer:  $\theta_0 = 10.11283406777275$

$\theta_1 = 8.271831294479673$

2. A plot, which visualizing the change in cost function  $J(\theta)$  at each iteration.

Answer:



3. RMSE for your training set when you use TV feature.

Answer: RMSE(TV training set) = 3.640 (3 digits after the decimal point)

4. RMSE for test set, when you use TV feature.

Answer: RMSE(TV testing set) = 3.909 (3 digits after the decimal point)

5. RMSE for test set, when you use Radio feature.

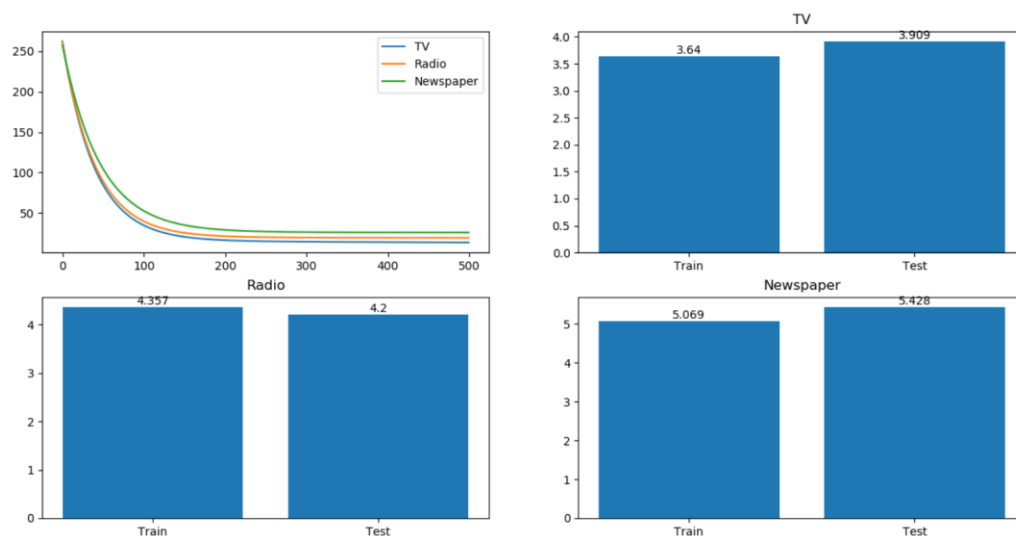
Answer: RMSE(Radio test set) = 4.200 (3 digits after the decimal point)

6. RMSE for test set, when you use newspaper feature.

**Answer: RMSE(Newspaper test set) = 5.428 (3 digits after the decimal point)**

**7. Compare the performance of your three models and rank them accordingly.**

**Answer:**



**I compared three models using the graph above. The RMSE of TV is the smallest.**

**Rank: (with loss) TV Radio Newspaper**

**(with RMSE) TV Radio Newspaper**

**Code:**

```
1. import csv
2. import math
3. import matplotlib.pyplot as plt
4. x1 = []
5. x2 = []
6. x3 = []
7. y = []
8. with open('Advertising.csv','r') as csvfile:
9.     reader = csv.DictReader(csvfile)
10.     for row in reader:
```

```

11.         x1.append(float(row['TV']))
12.         x2.append(float(row['Radio']))
13.         x3.append(float(row['Newspaper']))
14.         y.append(float(row['Sales']))
15.
16. #pre-processing
17. def normalisation(xlist):
18.     xmin = min(xlist)
19.     xmax = max(xlist)
20.     for i in range(len(xlist)):
21.         xlist[i] = (xlist[i] - xmin)/(xmax - xmin)
22.     return xlist
23.
24. x1 = normalisation(x1)
25. x2 = normalisation(x2)
26. x3 = normalisation(x3)
27.
28. #creating test and training set
29. training_list = [x1[:190],x2[:190],x3[:190],y[:190]]
30. test_list = [x1[190:201],x2[190:201],x3[190:201],y[190:201]]
31.
32. #Gradient descent
33. def gradientDescent(theta0,theta1,xlist,ylist):
34.     x0 = 1
35.     alpha = 0.01
36.     num = 0
37.     J_theta = []
38.     while(num<500):
39.         new_list0 = []
40.         new_list1 = []
41.         new_list2 = []
42.         h_x = []
43.         for i in xlist:
44.             h_x.append(theta0 * x0 + theta1 * i)
45.         for j in range(len(xlist)):
46.             new_list0.append((ylist[j] - h_x[j]))
47.             new_list1.append((ylist[j] - h_x[j])*xlist[j])
48.             new_list2.append((ylist[j] - h_x[j])**2)
49.         theta0 = theta0 + (sum(new_list0)/190)*alpha
50.         theta1 = theta1 + (sum(new_list1)/190)*alpha
51.         J_theta.append(sum(new_list2)/190)
52.         num = num + 1
53.     return theta0,theta1,J_theta
54.

```

```

55. theta0 = -1
56. theta1 = -0.5
57. theta0_0,theta0_1,graph1 = gradientDescent(theta0,theta1,training_list[0],tr
    aining_list[3])
58. theta1_0,theta1_1,graph2 = gradientDescent(theta0,theta1,training_list[1],tr
    aining_list[3])
59. theta2_0,theta2_1,graph3 = gradientDescent(theta0,theta1,training_list[2],tr
    aining_list[3])
60.
61. #visualization
62. plt.figure()
63. plt.subplot(2,2,1)
64. plt.plot(graph1,label = 'TV')
65. plt.plot(graph2,label = 'Radio')
66. plt.plot(graph3,label = 'Newspaper')
67. plot.legend()
68. #plt.show()
69.
70. #evaluation
71. def RMSE(xlist,ylist,theta0,theta1):
72.     h_theta = []
73.     L = []
74.     for i in xlist:
75.         h_theta.append(theta0 + theta1 * i)
76.         #print(h_theta)
77.         length = len(xlist)
78.         for j in range(length):
79.             L.append((ylist[j] - h_theta[j])**2)
80.         rmse = round(math.sqrt(sum(L)/length),3)
81.         return rmse
82.
83. rmse0 = [RMSE(training_list[0],training_list[3],theta0_0,theta0_1),RMSE(test
    _list[0],test_list[3],theta0_0,theta0_1)]
84. rmse1 = [RMSE(training_list[1],training_list[3],theta1_0,theta1_1),RMSE(test
    _list[1],test_list[3],theta1_0,theta1_1)]
85. rmse2 = [RMSE(training_list[2],training_list[3],theta2_0,theta2_1),RMSE(test
    _list[2],test_list[3],theta2_0,theta2_1)]
86.
87. print(rmse0)
88. print(rmse1)
89. print(rmse2)
90.
91. namelist = ['Train','Test']
92. a = plt.subplot(222)

```

```
93. a.set_title('TV')
94. plt.bar(x = namelist,height = rmse0)
95. for x, y in enumerate(rmse0):
96.     plt.text(x, y, '%s' % y, ha='center', va='bottom')
97. b = plt.subplot(223)
98. b.set_title('Radio')
99. plt.bar(x = namelist,height = rmse1)
100. for x, y in enumerate(rmse1):
101.     plt.text(x, y, '%s' % y, ha='center', va='bottom')
102. c = plt.subplot(224)
103. c.set_title('Newspaper')
104. plt.bar(x = namelist,height = rmse2)
105. for x, y in enumerate(rmse2):
106.     plt.text(x, y, '%s' % y, ha='center', va='bottom')
107. plt.show()
```