Abigail Dawn P. Davocol
BS in Computer Science IV
Machine Learning

Questions:
Note that the weights are generated randomly and the results may vary across execution. While you have a function for generating random weights, assume that the initial weights are all 0s.

init_w = np.array([0.0, 0.0, 0.0, 0.0])

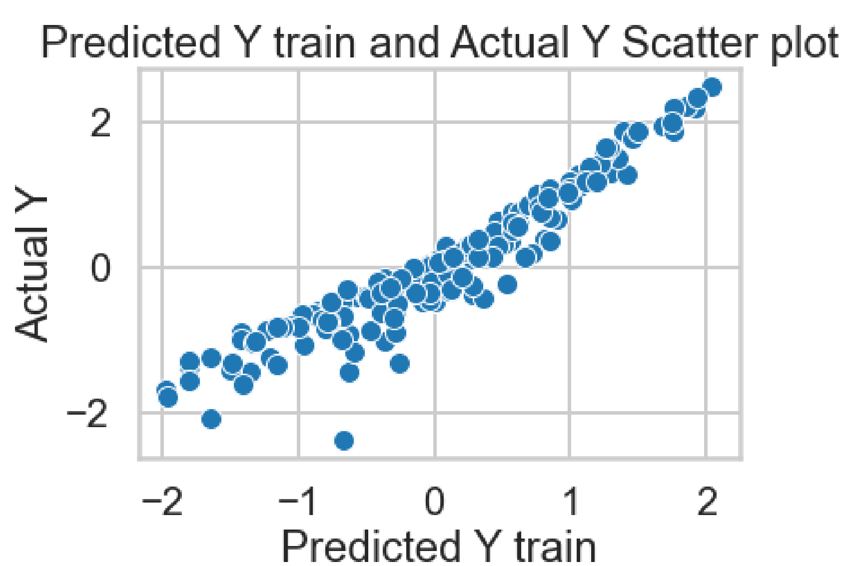
1. What are the optimal weights found by your implemented gradient descent? Plug it into the linear model: h! (x) = θ" + θ# TV + $\theta\$$ $Radio$ + θ6 $Newspaper$ What are your interpretations regarding the formed linear model?

The linear model is: $h\theta(x)$ = 4.35589486e-04 + 7.37383146e-01 * TV + 5.36307180e-01 * $Radio$ + 3.14254020e-03 * $Newspaper$

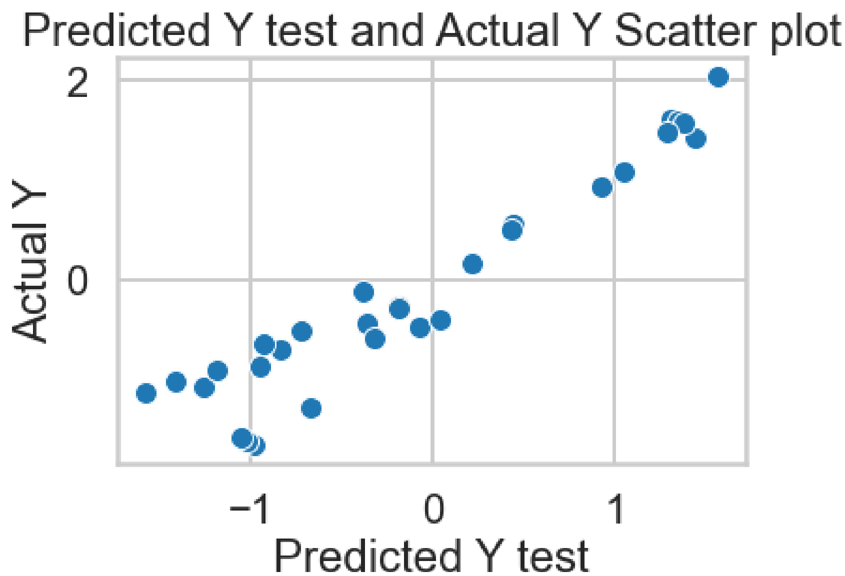
Looking at the linear model, we can observe that the TV is the greatest predictor compared to Radio and Newspaper. Next to TV is Radio then followed by Newspaper.

2. Provide a scatter plot of the 1y (,)5 2 and y (() for both the train and test set. Is there a trend? Provide an r2 score (also available in sklearn).

```
The R2 score using train set is: 0.8936793584593189
```



```
The R2 score using test set is: 0.911027570209169
```



3. What happens to the error, r2, and cost as the number of iterations increase? Show your data and proof. You can alternatively plot your result data for visualization and check until 50000 iterations or more (actually).

For **500 Iteration:**
The **Cost** for 500 iteration is: **0.05118539064304306**
The **r2** score for 500 iteration is : **0.896415695070808**
The **MAE** for 500 iteration is : **0.2413453935318982**7

For **1000 iteration:**
The **Cost** for 1000 iteration is: **0.051079831065374885**
The **r2** score for 1000 iteration is : **0.8969142691949464**
The **MAE** for 1000 iteration is : **0.2399596550148806**

For **10000 iteration:**
The **Cost** for 10000 iteration is: **0.051079554729786925**
The **r2** score for 10000 iteration is : **0.8969225203073516**
The **MAE** for 10000 iteration is : **0.2398886365772415**

It can be observed base on the results of cost, r2 score and MAE that the higher the number of iteration, the r2 score increases while the cost and MAE decreases. However, these differences are really small.

4. Once you determine the optimal number of iterations, check the effect on the cost and error as you change the learning rate. The common learning rates in

machine learning include 0.1, 0.01, 0.001, 0.0001, 0.2 but you have the option to include others. Visualize the cost function (vs the optimal number of iterations) of each learning rate in ONLY ONE PLOT. Provide your analysis.

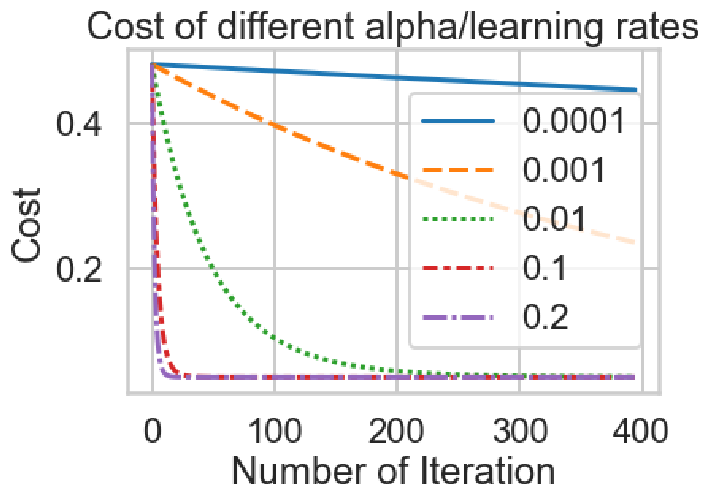
When alpha rate is: 0.1
Cost: 0.05107955472978691
R2: 0.8969225203072294
MAE: 0.2398886365786446

When alpha rate is: 0.01
Cost: 0.051500250039664096
R2: 0.8953717254069102
MAE: 0.24272053055351145

When alpha rate is: 0.001
Cost: 0.23622673852442644
R2: 0.5060374784393623
MAE: 0.563782776791368

When alpha rate is: 0.0001
Cost: 0.44536428998765176
R2: 0.07286799176886316
MAE: 0.7890090157139464

When alpha rate is: 0.2
Cost: 0.051079554729786925
R2: 0.8969225203073514
MAE: 0.239888636577241



It can be observed from the graph that the 0.2 reached the lowest cost the fastest and it was followed by 0.1 then followed by 0.01. While, 0.001 is linear and lastly, 0.0001 did not achieve the lowest cost at 395 iteration. Both 0.001 and 0.0001 did not reach lowest cost within the optimal iteration.

5. Is there a relationship on the learning rate and the number of iterations?
Base on the previous graph, if the learning rate is higher, then the number of iteration to get the optimal cost will decrease. Hence, the learning rate and the number of iteration has an inversely proporional relationship.

6. Compare the results with the results of ordinary least squares function.

R2 score when using **linear regression**:

0.9110275702091712

R2 score when using the **weights of the optimal iteration of the gradient descent**:

0.906288920753653

The difference between the R2 score are not that really big. But the linear regression is much closer to one so compared to the r2 score which usedthe optimal iteration weights the linear regression model is better fitted.