Project -1

Task:

Part1, part2, and part3 are performed as part of the code
Initial linear regression equation

```
> betas <- solve(t(x)%*%x)%*%t(x)%*%y
> betas
                    [,1]
x0          -0.009534068
season       0.119641553
mnth         0.001981801
hr           0.279093205
holiday     -0.016837240
weekday      0.020270827
workingday  -0.006439391
weathersit   0.019616607
temp        -0.014079171
atemp        0.327709262
hum         -0.239564842
windspeed    0.031775625
```

Experimentation:

1. Experiment with various values of learning rate ∝ and report on your findings as how the error varies for train and test sets with varying ∝. Plot the results. Report your best ∝ and why you picked it.

I have experimented with various values of alpha(∝) and noted down all Error for each value of alpha and have created a Experimental_Dataset_train dataset with Iterations, MSE_new, Converged, conv_threshold, max_iter and Dataset from each value of Alpha and threshold.

<u>Alpha Vs MSE for Train Dataset:</u>

Filter

	iterations	MSE_new	converged	alpha	conv_threshold	max_iter	Dataset
1	469	22108.93	Yes	0.010	1e-01	10000	Train
2	721	22100.81	Yes	0.010	1e-02	10000	Train
3	1405	22098.39	Yes	0.010	1e-03	10000	Train
4	2417	22098.01	Yes	0.010	1e-04	10000	Train
5	10000	22097.78	No	0.010	1e-05	10000	Train
6	3363	22159.36	Yes	0.001	1e-01	10000	Train
7	4689	22108.73	Yes	0.001	1e-02	10000	Train
8	7366	22100.82	Yes	0.001	1e-03	10000	Train
9	10000	22099.20	No	0.001	1e-04	10000	Train
10	10000	22099.17	No	0.001	1e-05	10000	Train
11	72	22100.67	Yes	0.100	1e-01	10000	Train
12	141	22098.37	Yes	0.100	1e-02	10000	Train
13	244	22098.01	Yes	0.100	1e-03	10000	Train

Showing 1 to 14 of 25 entries

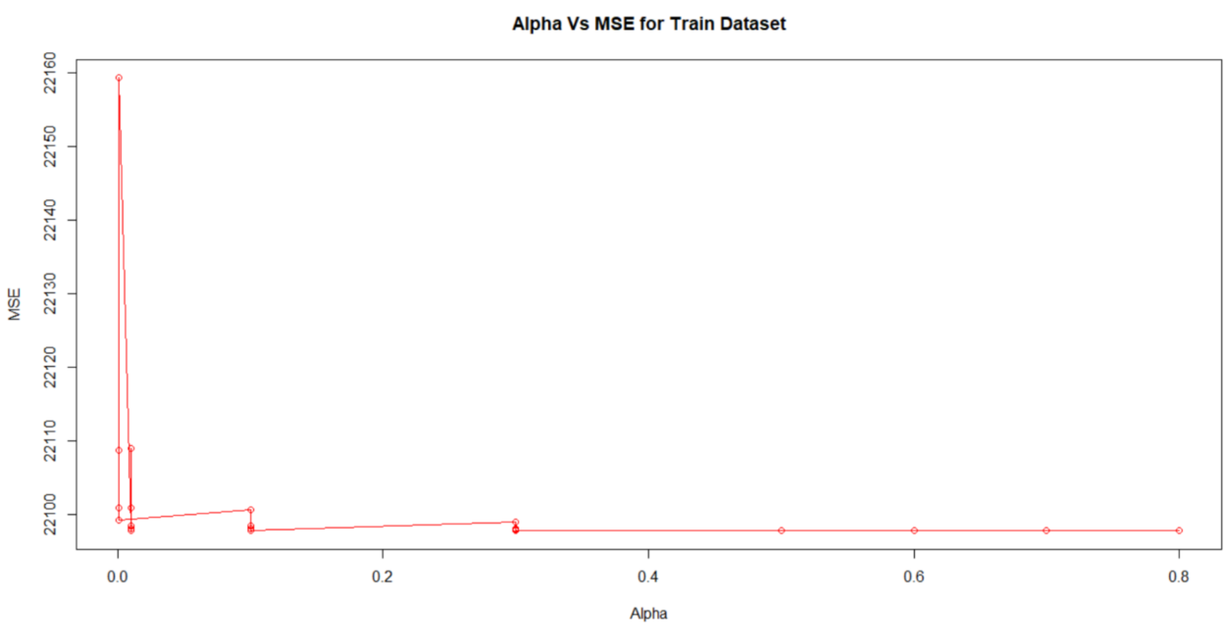
I have created the same for test dataset as well with the name Experimental_Dataset_test dataset

Filter

	iterations	MSE_new	converged	alpha	conv_threshold	max_iter	Dataset
1	56	0.6347100	Yes	0.1	1e-04	10000	Test
2	29	0.6361800	Yes	0.2	1e-04	10000	Test
3	37	0.6342600	Yes	0.3	1e-04	10000	Test
4	33	0.6342200	Yes	0.4	1e-04	10000	Test
5	16	0.6336100	Yes	0.5	1e-04	10000	Test
6	23	0.6359500	Yes	0.6	1e-04	10000	Test
7	20	0.6330900	Yes	0.7	1e-04	10000	Test
8	6	0.8577900	Yes	0.7	1e-01	10000	Test
9	10	0.6421800	Yes	0.7	1e-02	10000	Test
10	15	0.6343700	Yes	0.7	1e-03	10000	Test
11	29	0.6333231	Yes	0.7	1e-05	10000	Test
12	75	0.6329445	Yes	0.7	1e-06	10000	Test

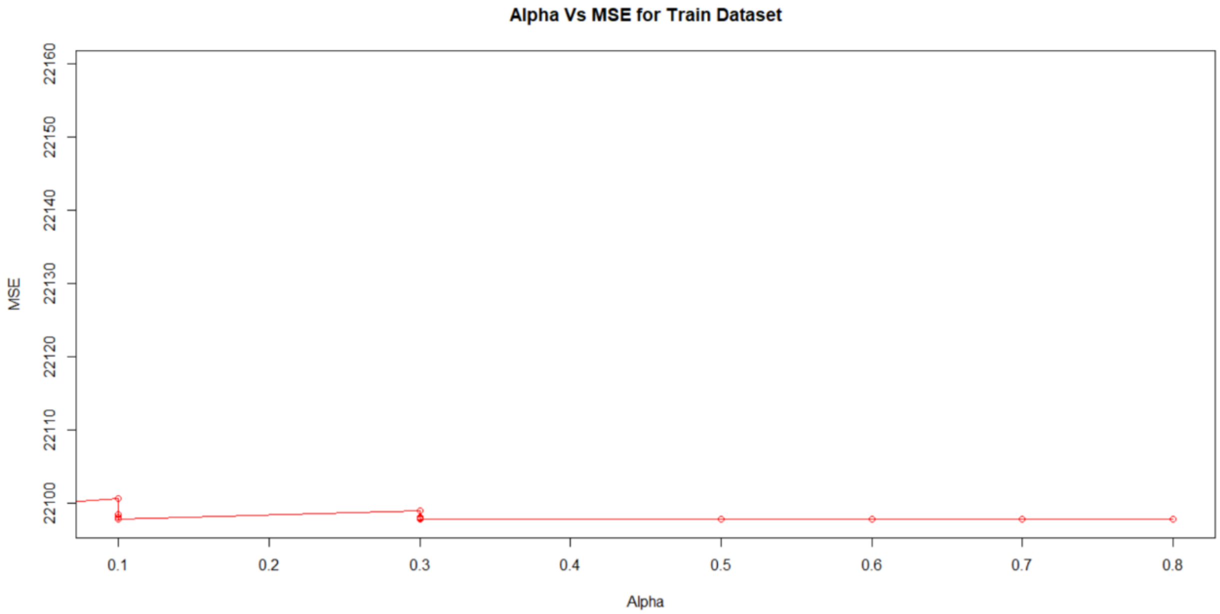
Showing 1 to 12 of 12 entries

I load that dataset in to R and plot graphs for Alpha and MSE for Train and Test Dataset



Closer look:

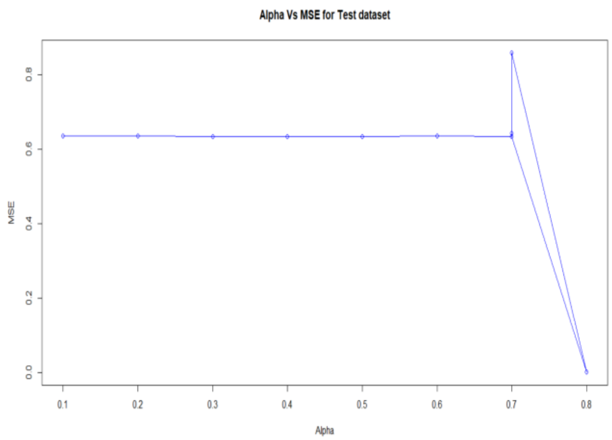
plot(Experimental_Dataset_train$alpha, Experimental_Dataset_train$MSE_new,type = "o", col = "red", xlab = "Alpha", ylab = "MSE", main = "Alpha Vs MSE for Train Dataset",xlim = c(0.1,0.8))

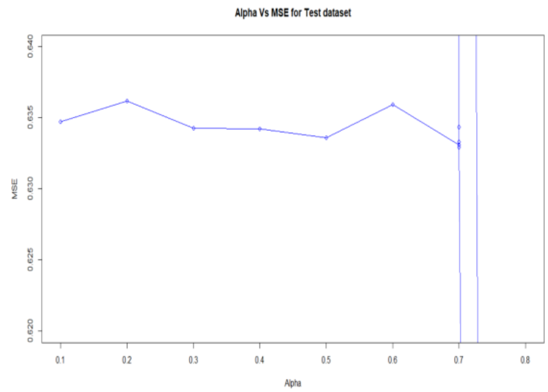


The optimal Error is at alpha = 0.7 where, Error = 22097.74 with the least iteration of 259 at Threshold value = 0.0001

10000	22099.17	No	0.001	1.00E-05	10000	Train
72	22100.67	Yes	0.1	1.00E-01	10000	Train
141	22098.37	Yes	0.1	1.00E-02	10000	Train
244	22098.01	Yes	0.1	1.00E-03	10000	Train
1001	22097.78	Yes	0.1	1.00E-04	10000	Train
1995	22097.79	Yes	0.1	1.00E-05	10000	Train
34	22098.99	Yes	0.3	1.00E-01	10000	Train
60	22098.1	Yes	0.3	1.00E-02	10000	Train
174	22097.87	Yes	0.3	1.00E-03	10000	Train
474	22097.75	yes	0.3	1.00E-04	10000	Train
339	22097.74	Yes	0.5	1.00E-04	10000	Train
514	22097.74	Yes	0.5	1.00E-05	10000	Train
704	22097.74	Yes	0.5	1.00E-06	10000	Train
280	22097.74	Yes	0.6	1.00E-04	10000	Train
259	22097.74	Yes	0.7	1.00E-04	10000	Train
277	22097.74	Yes	0.8	1.00E-04	10000	Train

Alpha Vs MSE for Test Dataset:





Filter

	iterations	MSE_new	converged	alpha	conv_threshold	max_iter	Dataset
1	56	0.6347100	Yes	0.1	1e-04	10000	Test
2	29	0.6361800	Yes	0.2	1e-04	10000	Test
3	37	0.6342600	Yes	0.3	1e-04	10000	Test
4	33	0.6342200	Yes	0.4	1e-04	10000	Test
5	16	0.6336100	Yes	0.5	1e-04	10000	Test
6	23	0.6359500	Yes	0.6	1e-04	10000	Test
7	20	0.6330900	Yes	0.7	1e-04	10000	Test
8	6	0.8577900	Yes	0.7	1e-01	10000	Test
9	10	0.6421800	Yes	0.7	1e-02	10000	Test
10	15	0.6343700	Yes	0.7	1e-03	10000	Test
11	29	0.6333231	Yes	0.7	1e-05	10000	Test
12	75	0.6329445	Yes	0.7	1e-06	10000	Test

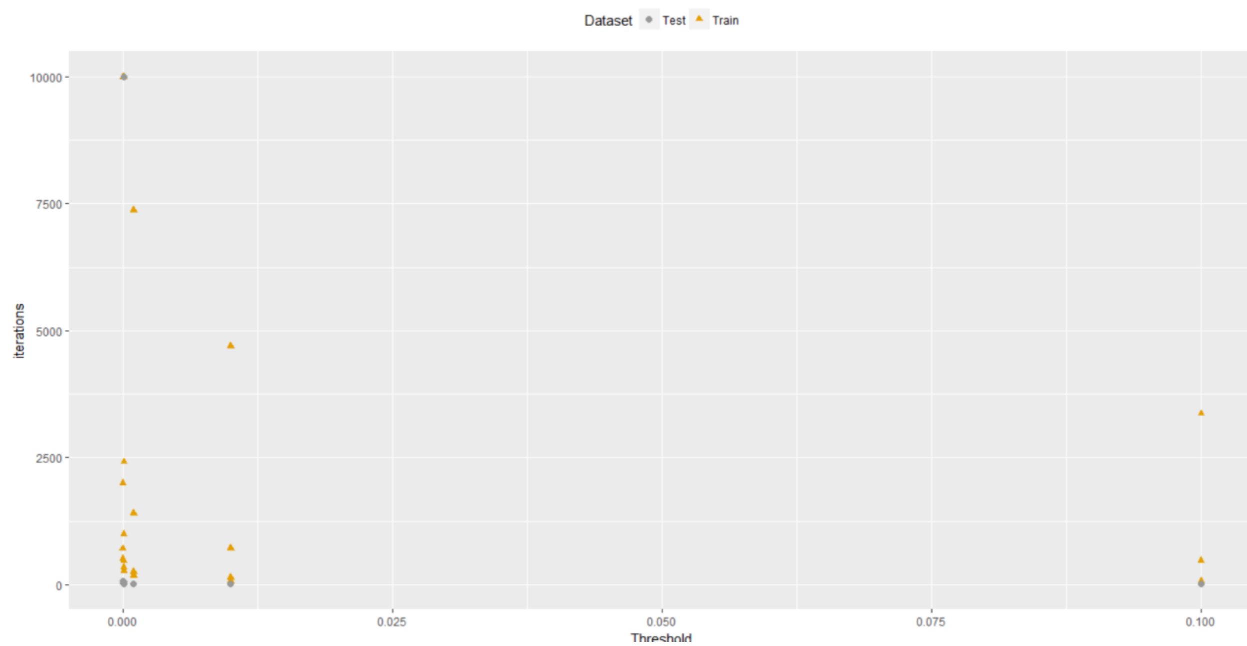
Showing 1 to 12 of 12 entries

The optimal Error is at alpha = 0.7 where, Error = 0.6330900 with the least iteration of 20 at Threshold value = 0.0001

So, our alpha will be alpha = 0.7 and value = 0.0001

2. Experiment with various thresholds for convergence. Plot error results for train and test sets as a function of threshold and describe how varying the threshold affects error. Pick your best threshold and plot train and test error (in one figure) as a function of number of gradient descent iterations.

10000	22099.17	No	0.001	1.00E-05	10000	Train
72	22100.67	Yes	0.1	1.00E-01	10000	Train
141	22098.37	Yes	0.1	1.00E-02	10000	Train
244	22098.01	Yes	0.1	1.00E-03	10000	Train
1001	22097.78	Yes	0.1	1.00E-04	10000	Train
1995	22097.79	Yes	0.1	1.00E-05	10000	Train
34	22098.99	Yes	0.3	1.00E-01	10000	Train
60	22098.1	Yes	0.3	1.00E-02	10000	Train
174	22097.87	Yes	0.3	1.00E-03	10000	Train
474	22097.75	yes	0.3	1.00E-04	10000	Train
339	22097.74	Yes	0.5	1.00E-04	10000	Train
514	22097.74	Yes	0.5	1.00E-05	10000	Train
704	22097.74	Yes	0.5	1.00E-06	10000	Train
280	22097.74	Yes	0.6	1.00E-04	10000	Train
259	22097.74	Yes	0.7	1.00E-04	10000	Train
277	22097.74	Yes	0.8	1.00E-04	10000	Train



For any Alpha basically Error seems to be least starts from Threshold = 1e-04 and number of iterations = 259 as demonstrated above.

3. Pick three features randomly and retrain your model only on these 3 features. Compare train and test error results for the case of using all features to using three random features. Report which three features did you select randomly.

```
alpha <- .7
Iterations <- 500
results <- gradDescent(X, y, theta, alpha, Iterations)
theta <- results[[1]]
cost1 <- results[[2]]
print(theta)
print(cost1)
```

Experimentation with all the variables:

```
> print(theta)
                   [,1]
x0         -0.009509739
season      0.119677481
mnth        0.001963071
hr          0.279129260
holiday    -0.016848005
weekday     0.020249116
workingday -0.006446663
weathersit  0.019572195
temp       -0.007589218
atemp       0.321201891
hum        -0.239504466
windspeed   0.031530754
> print(cost_hist)
  [1] 0.3691073 0.3446897 0.3332054 0.3265876 0.3226457 0.3202810 0.3188581 0.3179998 0.3174806 0.3171654
 [11] 0.3169731 0.3168550 0.3167817 0.3167357 0.3167062 0.3166867 0.3166735 0.3166642 0.3166572 0.3166517
 [21] 0.3166472 0.3166433 0.3166398 0.3166366 0.3166336 0.3166308 0.3166280 0.3166254 0.3166228 0.3166203
 [31] 0.3166178 0.3166154 0.3166130 0.3166106 0.3166083 0.3166060 0.3166038 0.3166016 0.3165994 0.3165972
 [41] 0.3165951 0.3165930 0.3165909 0.3165889 0.3165869 0.3165849 0.3165829 0.3165810 0.3165791 0.3165772
 [51] 0.3165754 0.3165735 0.3165717 0.3165699 0.3165682 0.3165664 0.3165647 0.3165630 0.3165613 0.3165597
```

This does not converge.

Exp1:

My initial variables are season, hr, atemp

```
> print(theta)
                [,1]
x0     -0.008964354
season  0.077563768
hr      0.343994935
atemp   0.326383067
> print(cost_hist)
 [1] 0.3473538 0.3434317 0.3429641 0.3428872 0.3428700 0.3428654 0.3428640 0.3428636 0.3428635 0.3428635
[11] 0.3428635 0.3428635 0.3428635 0.3428635 0.3428635 0.3428635 0.3428635 0.3428635 0.3428635 0.3428635
[21] 0.3428635 0.3428635 0.3428635 0.3428635 0.3428635 0.3428635 0.3428635 0.3428635 0.3428635 0.3428635
[31] 0.3428635 0.3428635 0.3428635 0.3428635 0.3428635 0.3428635 0.3428635 0.3428635 0.3428635 0.3428635
```

converges at 0.3428635

Exp2:

My initial variables are season, windspeed, temp

```
> print(theta)
                  [,1]
x0         -0.005524457
season      0.084668051
windspeed   0.119652175
temp        0.376385489
> print(cost_hist)
 [1] 0.3982346 0.3939022 0.3931955 0.3930376 0.3929956 0.3929836 0.3929801 0.3929790 0.3929787 0.3929786
[11] 0.3929786 0.3929786 0.3929786 0.3929786 0.3929786 0.3929786 0.3929786 0.3929786 0.3929786 0.3929786
```

converges at 0.3929786

Exp3:

My initial variables are hr, windspeed, atemp

```
> print(theta)
                  [,1]
x0         -0.009124658
hr          0.330397461
windspeed   0.081202193
atemp       0.357749894
> print(cost_hist)
 [1] 0.3472154 0.3425226 0.3423485 0.3423410 0.3423406 0.3423405 0.3423405 0.3423405 0.3423405 0.3423405
[11] 0.3423405 0.3423405 0.3423405 0.3423405 0.3423405 0.3423405 0.3423405 0.3423405 0.3423405 0.3423405
[21] 0.3423405 0.3423405 0.3423405 0.3423405 0.3423405 0.3423405 0.3423405 0.3423405 0.3423405 0.3423405
[31] 0.3423405 0.3423405 0.3423405 0.3423405 0.3423405 0.3423405 0.3423405 0.3423405 0.3423405 0.3423405
```

converges at 0.3423405

Exp4:

My initial variables are hum, windspeed, atemp

```
> print(theta)
                [,1]
x0        -0.01009416
hum       -0.28148950
windspeed  0.04040799
atemp      0.39126414
> print(cost_hist)
 [1] 0.3677227 0.3602229 0.3595248 0.3594357 0.3594194 0.3594157 0.3594147 0.3594144 0.3594143 0.3594143
[11] 0.3594143 0.3594143 0.3594143 0.3594143 0.3594143 0.3594143 0.3594143 0.3594143 0.3594143 0.3594143
[21] 0.3594143 0.3594143 0.3594143 0.3594143 0.3594143 0.3594143 0.3594143 0.3594143 0.3594143 0.3594143
[31] 0.3594143 0.3594143 0.3594143 0.3594143 0.3594143 0.3594143 0.3594143 0.3594143 0.3594143 0.3594143
[41] 0.3594143 0.3594143 0.3594143 0.3594143 0.3594143 0.3594143 0.3594143 0.3594143 0.3594143 0.3594143
```

converges at 0.3594143

Exp5:

My initial variables are hum, hr, atemp

```
> print(theta)
            [,1]
x0    -0.01086535
hum   -0.21990226
hr     0.28381394
atemp  0.35117080
> print(cost_hist)
 [1] 0.3258390 0.3236135 0.3234639 0.3234469 0.3234447 0.3234443 0.3234442 0.3234442 0.3234442 0.3234442
[11] 0.3234442 0.3234442 0.3234442 0.3234442 0.3234442 0.3234442 0.3234442 0.3234442 0.3234442 0.3234442
[21] 0.3234442 0.3234442 0.3234442 0.3234442 0.3234442 0.3234442 0.3234442 0.3234442 0.3234442 0.3234442
[31] 0.3234442 0.3234442 0.3234442 0.3234442 0.3234442 0.3234442 0.3234442 0.3234442 0.3234442 0.3234442
[41] 0.3234442 0.3234442 0.3234442 0.3234442 0.3234442 0.3234442 0.3234442 0.3234442 0.3234442 0.3234442
```

converges at 0.3234442

For Testing Dataset: I would use hum, hr, atemp as my three variables because it has the least error and converges at very less iterations for the best alpha and Iteration values which are determined earlier.

Same applies for training dataset as well. So, I would choose hum, hr, atemp as my feature variables.

4. Now pick three features that you think are best suited to predict the output, and retrain your model using these three features. Compare to the case of using all features and to random features case. Did your choice of features provide better results than picking random features? Why? Did your choice of features provide better results than using all features? Why?

My choice of features will be hum, hr, atemp as I got optimal cost when I used the gradient descent. If I use all features the cost taking longer to reach the optimal point