HOMEWORK 1

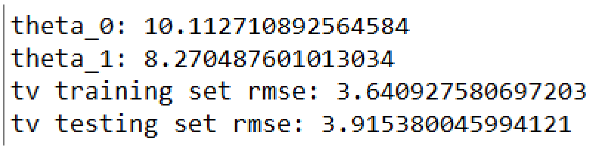
1.

When using TV features

θ_0=10.112710892564584

θ_1=8.270487601013034

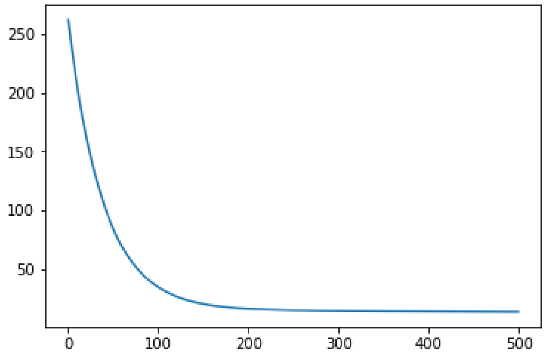
The picture below shows the value which I calculated from my machine



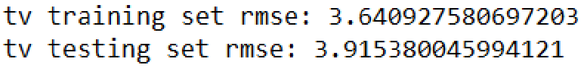
```
theta_0: 10.112710892564584
theta_1: 8.270487601013034
tv training set rmse: 3.640927580697203
tv testing set rmse: 3.915380045994121
```

2.

By using plot function from ‘matplotlib.pyplot’ module, I could obtain this graph representing the change of cost function in TV features, which x-axis stands for the iteration times and y-axis stands for the change of J_θ.



3.



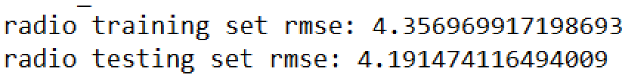
```
tv training set rmse: 3.640927580697203
tv testing set rmse: 3.915380045994121
```

RMSE for training set(TV feature): 3.640927580697203

4.

RMSE for test set(TV feature): 3.915380045994121

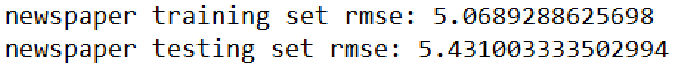
5.



```
radio training set rmse: 4.356969917198693
radio testing set rmse: 4.191474116494009
```

RMSE for test set(Radio feature): 4.191474116494009

6.



```
newspaper training set rmse: 5.0689288625698
newspaper testing set rmse: 5.431003333502994
```

RMSE for test set(newspaper feature): 5.431003333502994

7.TV > Radio > Newspaper.
The performance of models depend on the value of RMSE.The model which has lower rmse value has the better performance,so the rank is like that.

	TV	RATIO	NEWSPAPER
Training_set	3.640927580697203	4.356969917198693	5.0689288625698
Test_set	3.915380045994121	4.191474116494009	5.431003333502994

Coding part

```
import numpy as np
import matplotlib.pyplot as plt
import math

with open('Advertising.csv','r') as file:
    tv = []
    radio = []
    newspaper = []
    sales =[]
    splited = []
    for row in file:
        row = row.strip('\n')
        splited = row.split(',')#[tv,ratio,]
        if splited[1] != 'TV':
            tv.append(float(splited[1]))
        if splited[2] != 'Radio':
            radio.append(float(splited[2]))
        if splited[3] != 'Newspaper':
            newspaper.append(float(splited[3]))
        if splited[4] != 'Sales':
            sales.append(float(splited[4]))


def zero_one_format(tempp):
    changed = []
    for i in tempp:
        num = (i-min(tempp))/(max(tempp)-min(tempp))
        changed.append(round(num,2))
    return changed

newspaper = zero_one_format(newspaper)
newspaper_training = newspaper[:190]
newspaper_testing = newspaper[190:]
```

```
print(sales)

theta_0 = -1
theta_1 = -0.5
learning_rate = 0.01
loss = []
for i in range(500):
    error = []
    error_1 = []
    for j in range(190):
        predicted = theta_0 +theta_1 * newspaper_training[j]
        error.append(sales[j] - predicted)
        error_1.append((sales[j] - predicted)*newspaper_training[j])
    theta_0 = theta_0 + learning_rate * (np.mean(error))
    theta_1 = theta_1 + learning_rate * (np.mean(error_1))
    loss.append([np.mean(error)])



###print(theta_1)
plt.plot(loss)
plt.show()
print(f'The third time:\ntheta_0: {theta_0}')
print(f'theta_1: {theta_1}')

###rmse
###for test set
error_rmse = []
for i in range(len(newspaper_training)):
    predicted = theta_0 + theta_1 * newspaper_training[i]
    temp = (sales[i] - predicted)**2
    error_rmse.append(temp)

print(f'newspaper training set rmse: {math.sqrt(np.mean(error_rmse))}')
```