CS 4375

ASSIGNMENT 1

Names of students in your group:

Shruti Bindingnavile

Number of free late days used:

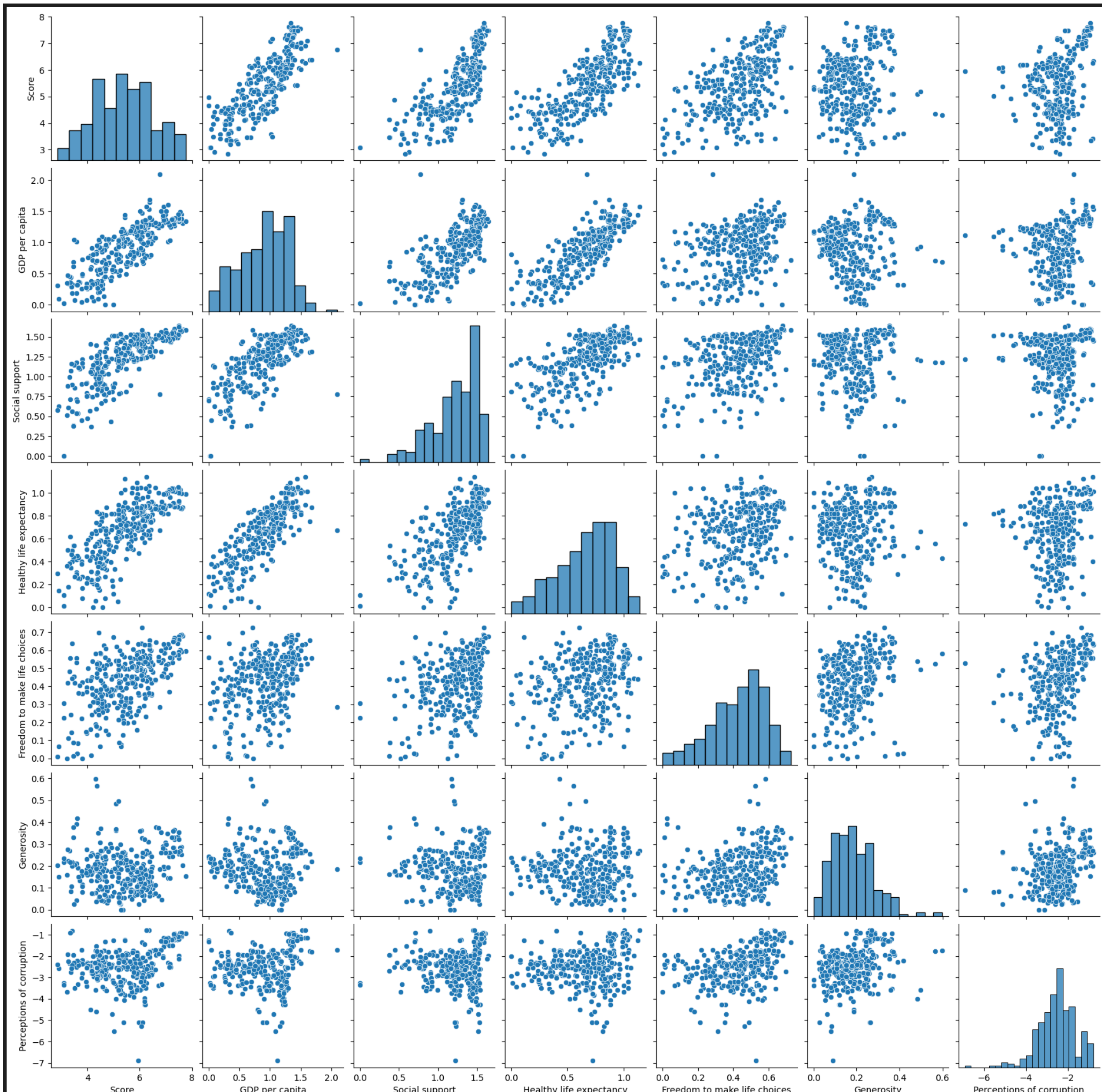
__0______________________

Note: You are allowed a **total** of 4 free late days for the **entire semester**. You can use at most 2 for each assignment. After that, there will be a penalty of 10% for each late day.

Please list clearly all the sources/references that you have used in this assignment.

Happiness Index 2018 - 2019

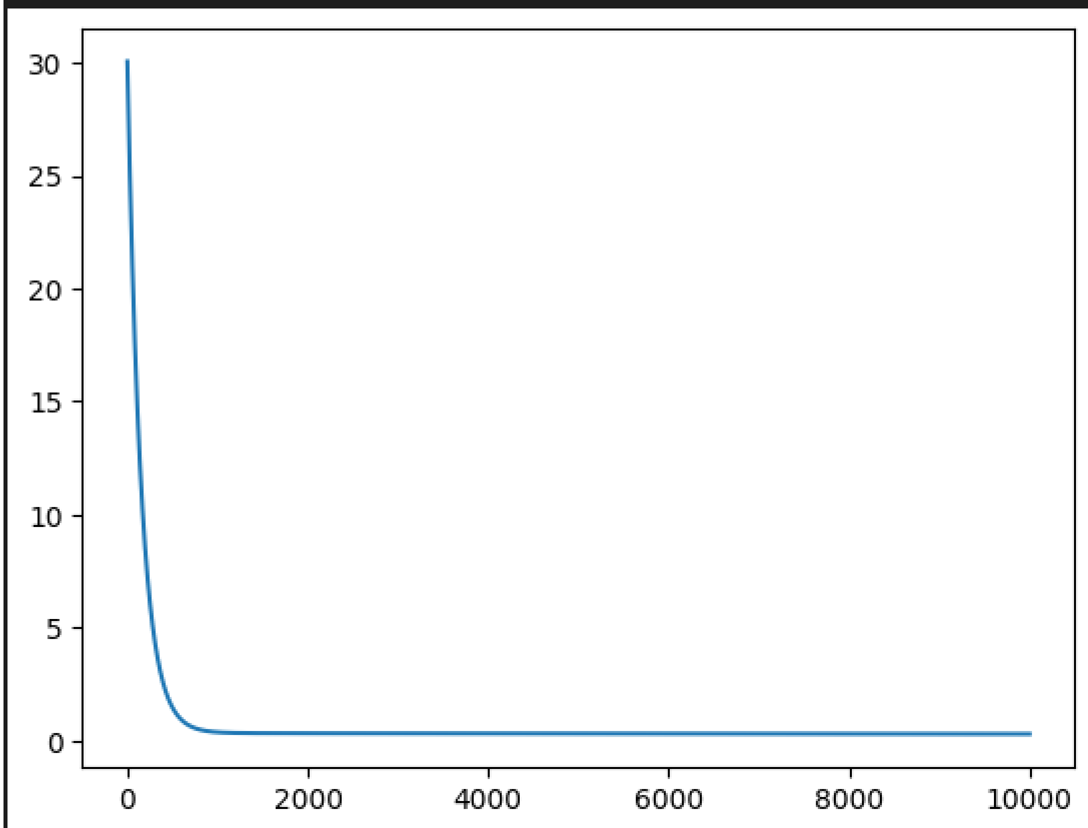
https://www.kaggle.com/datasets/sougatapramanick/happiness-index-2018-2019/code?resource=download


Score
GDP per capita
Social support
Healthy life expectancy
Freedom to make life choices
Generosity
Perceptions of corruption

Part 1

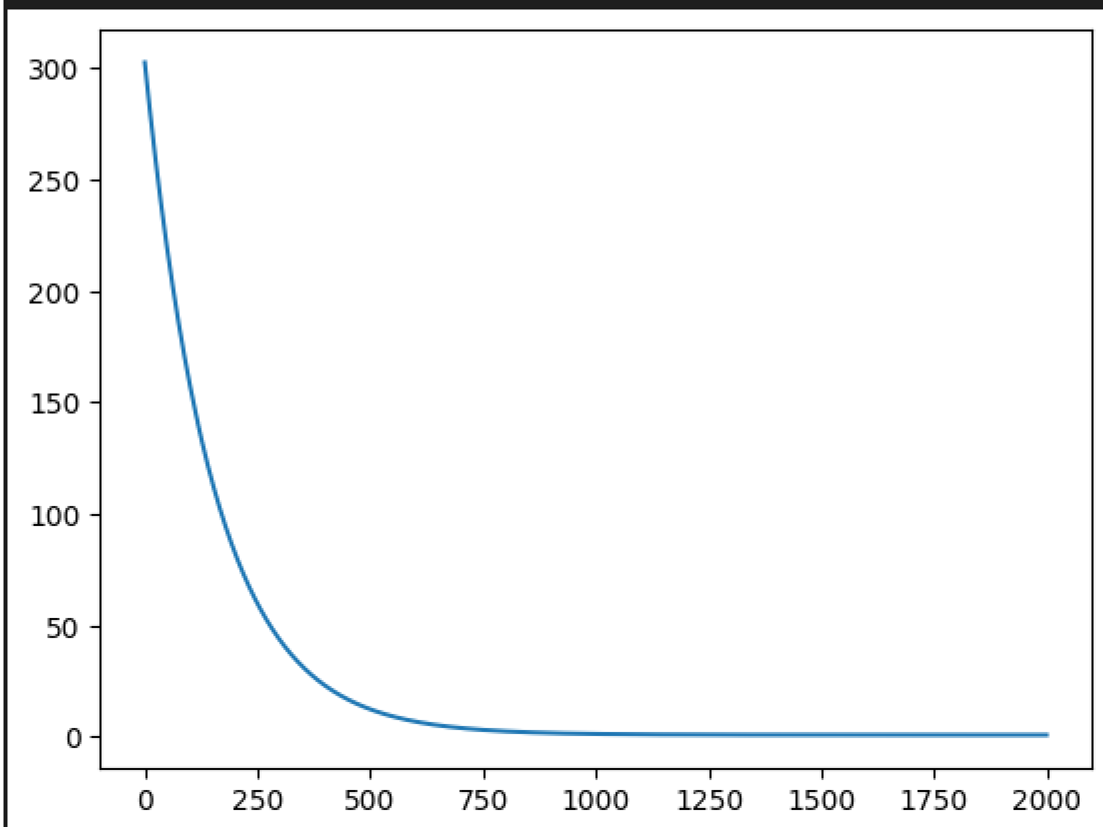
```
start=[0,0,0,0,0,0,0], learn_rate=0.0008,n_iter=10000, tolerance=1e-06
Train
r2 = 0.7560282578444792
mse = 0.295064924940832

Test
r2 = 0.759161879799913
mse = 0.3269598220836805
```



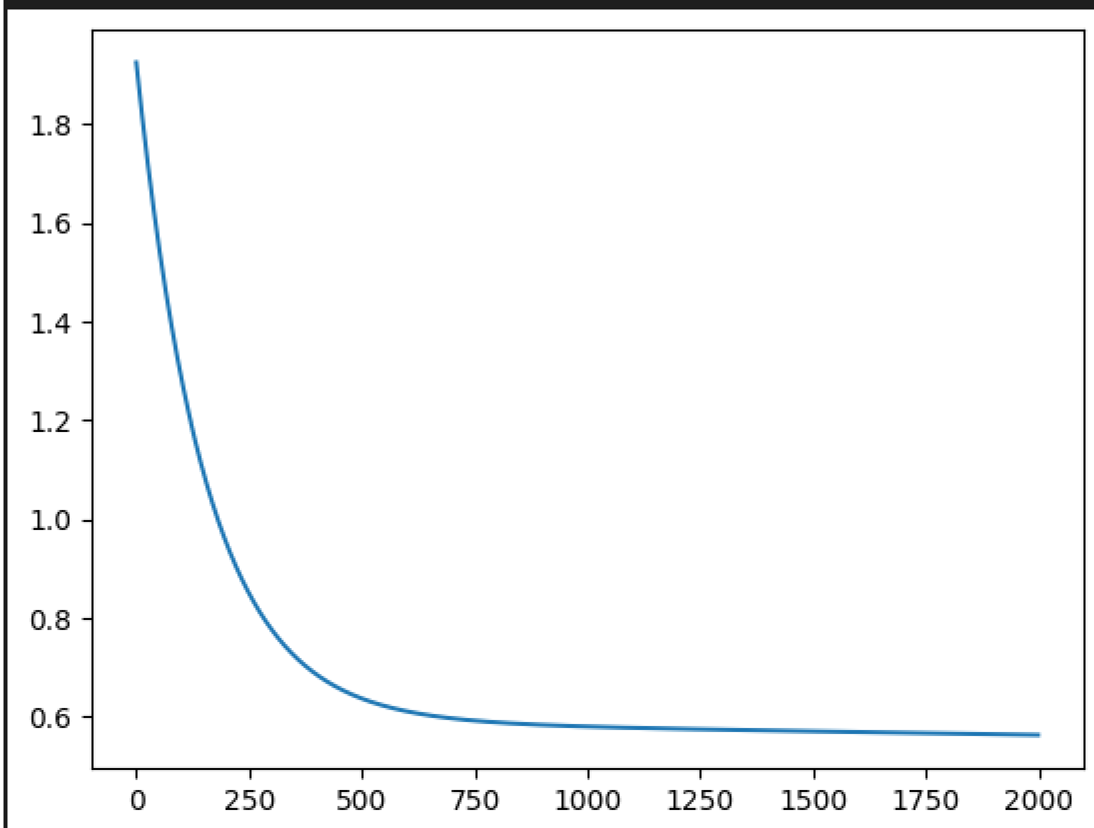
```
start=[5,5,5,5,5,5,5], learn_rate=0.0008,n_iter=2000, tolerance=1e-06
Train
r2 = 0.3023070561329305
mse = 0.8438055747565048

Test
r2 = 0.40231545549254244
mse = 0.8114115496831278
```



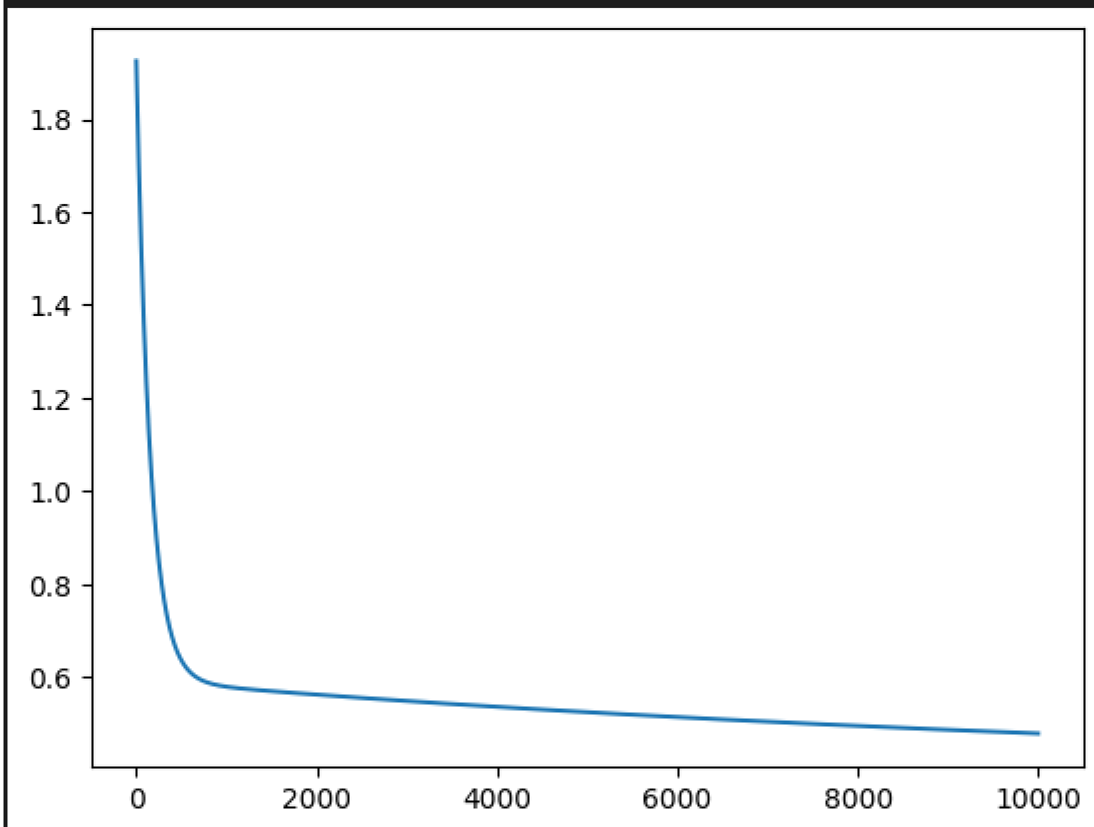
```
start=[0,1,0,2,3,4,0], learn_rate=0.0008,n_iter=2000, tolerance=1e-06
Train
r2 = 0.5347823777714291
mse = 0.5626446799585613

Test
r2 = 0.6161370427571641
mse = 0.5211291472477942
```



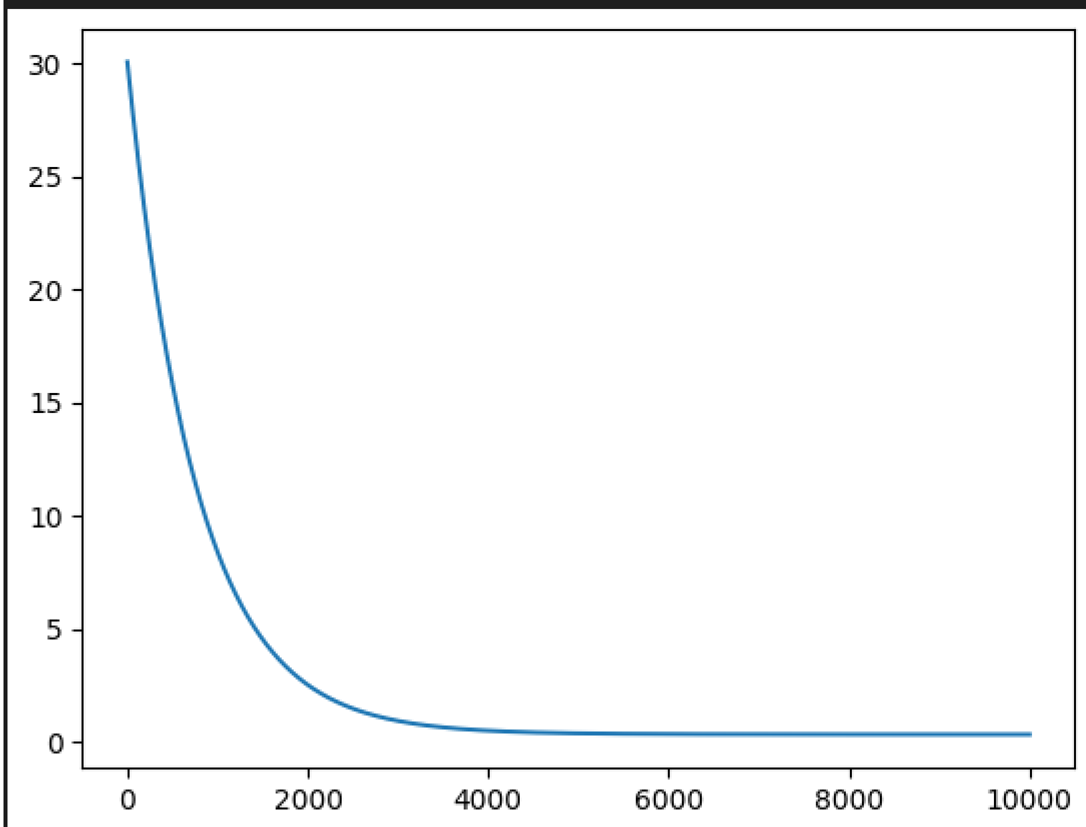
```
start=[0,1,0,2,3,4,0], learn_rate=0.0008,n_iter=10000, tolerance=1e-06
Train
r2 = 0.6038567952467403
mse = 0.4791045222845132

Test
r2 = 0.6778540677707743
mse = 0.4373426291450143
```



```
start=[0,0,0,0,0,0,0], learn_rate=0.00016,n_iter=10000, tolerance=1e-06
Train
r2 = 0.7707315627317421
mse = 0.27728241654616376

Test
r2 = 0.7580231923624826
mse = 0.32850569464588897
```




```
----------------------------------------------------------------------------
[1.56605901, 1.08720333, 1.55462775, 0.83534503, 0.72284645,
       0.34489078, 0.22899144]
It is hard to determine whether the best possible solution is found, as
every possible instance of the parameters was not tested. However, the
results of the regression had a good mean squared error and R^2.

One thing to note is the making the learning rate smaller improved the
model, however the difference is so minimal, that I went with the result
with the first option as it had a higher learning rate
```

Part 2

alpha=0.0008, max_iter=10000, tol=1e-6

Train
mse = 0.2665366916004825
r2 = 0.7882952363463965

Test
mse = 0.29168611779721315
r2 = 0.7351058997950832

alpha=0.00016, max_iter=10000, tol=1e-6
Train
mse = 0.2439097754706194
r2 = 0.8029500202730765

Test
mse = 0.38256746001776915
r2 = 0.6846412939889271

alpha=0.000001, max_iter=10000, tol=1e-6
Train
mse = 0.24428630241510046
r2 = 0.8026458314531184

Test
mse = 0.381992963942325
r2 = 0.6851148636410667

There is not much difference between the results of changing alpha (and also while not pictured, max_iter), the change in the test mse and r^2 was minimal.

[1.80200441, 0.91039399 1.26620038 0.87451504 1.23199449 0.45857401 0.52102031]

As stated above, it is hard to determine whether the best possible solution is found, as every possible instance of the parameters was not tested. However, the results of the regression had a good mean squared error and R^2. One thing to note is that while the second iteration training performed better, the testing error was better for the first option so I went with that. Some possible ways to improve results here include standardizing the data. I didn't do this in this case because I wanted to compare the results to my manual gradient descent.

The results of the two algorithms were comparable. Sklearn performed marginally better than my manual gradient descent algorithm but both produced similar weights and error rates.

```
sklearn:

[1.80200441, 0.91039399 1.26620038 0.87451504 1.23199449 0.45857401
0.52102031]

alpha=0.0008, max_iter=10000, tol=1e-6

Train
mse = 0.2665366916004825
r2 = 0.7882952363463965

Test
mse = 0.29168611779721315
r2 = 0.7351058997950832
manual:

[1.56605901, 1.08720333, 1.55462775, 0.83534503, 0.72284645,
       0.34489078, 0.22899144]

start=[0,0,0,0,0,0,0], learn_rate=0.0008,n_iter=10000, tolerance=1e-06
Train
mse = 0.295064924940832
r2 = 0.7560282578444792

Test
mse = 0.3269598220836805
r2 = 0.759161879799913
```