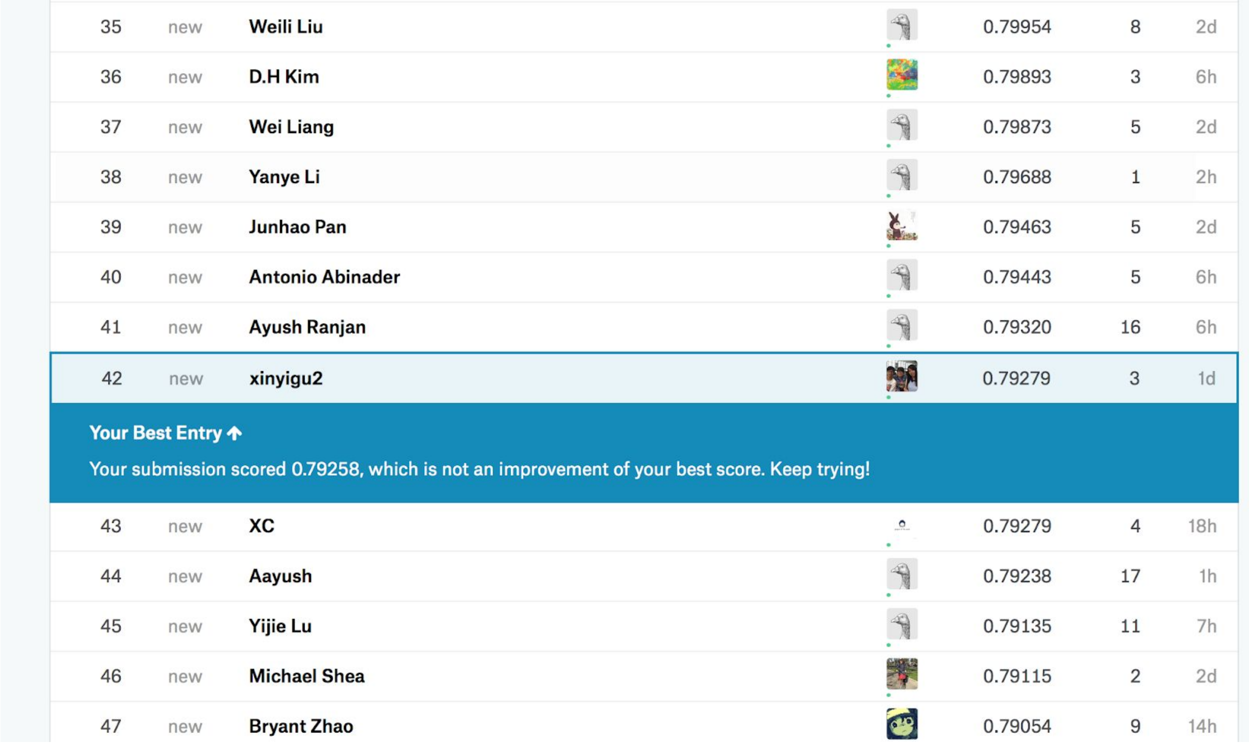
Page 1: screenshot of your leaderboard accuracy **and** mention your **best test dataset accuracy obtained on kaggle**.



35	new	**Weili Liu**	0.79954	8	2d
36	new	**D.H Kim**	0.79893	3	6h
37	new	**Wei Liang**	0.79873	5	2d
38	new	**Yanye Li**	0.79688	1	2h
39	new	**Junhao Pan**	0.79463	5	2d
40	new	**Antonio Abinader**	0.79443	5	6h
41	new	**Ayush Ranjan**	0.79320	16	6h
42	new	**xinyigu2**	0.79279	3	1d

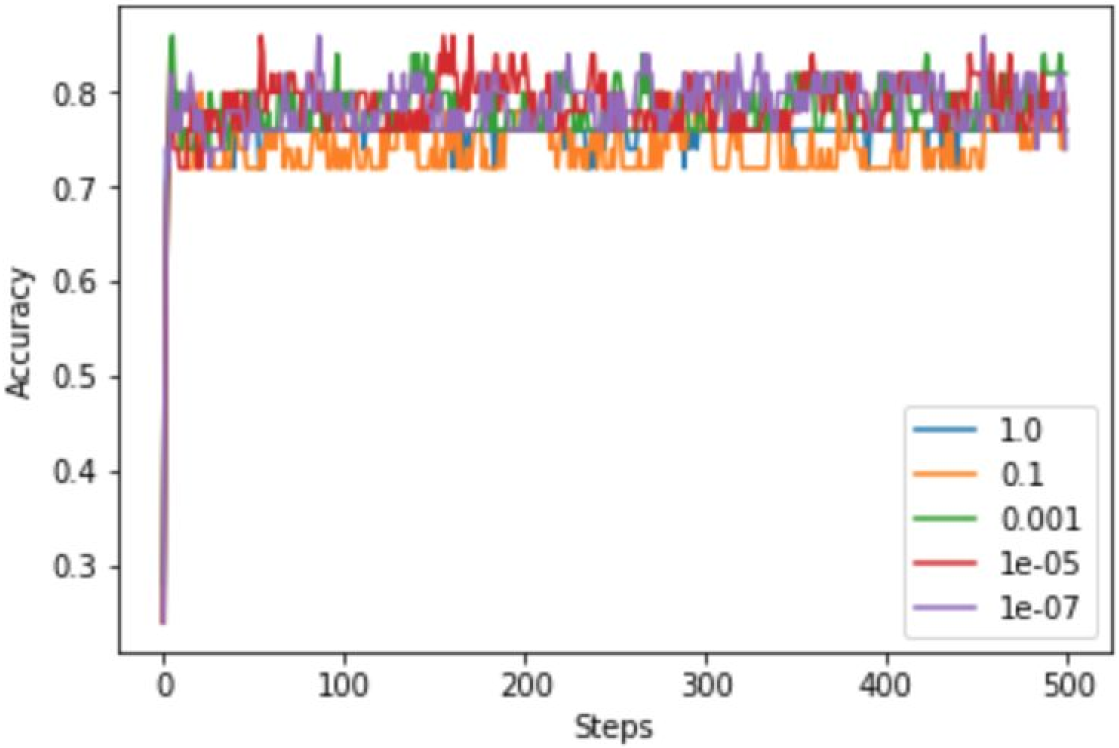
Your Best Entry ↑

Your submission scored 0.79258, which is not an improvement of your best score. Keep trying!

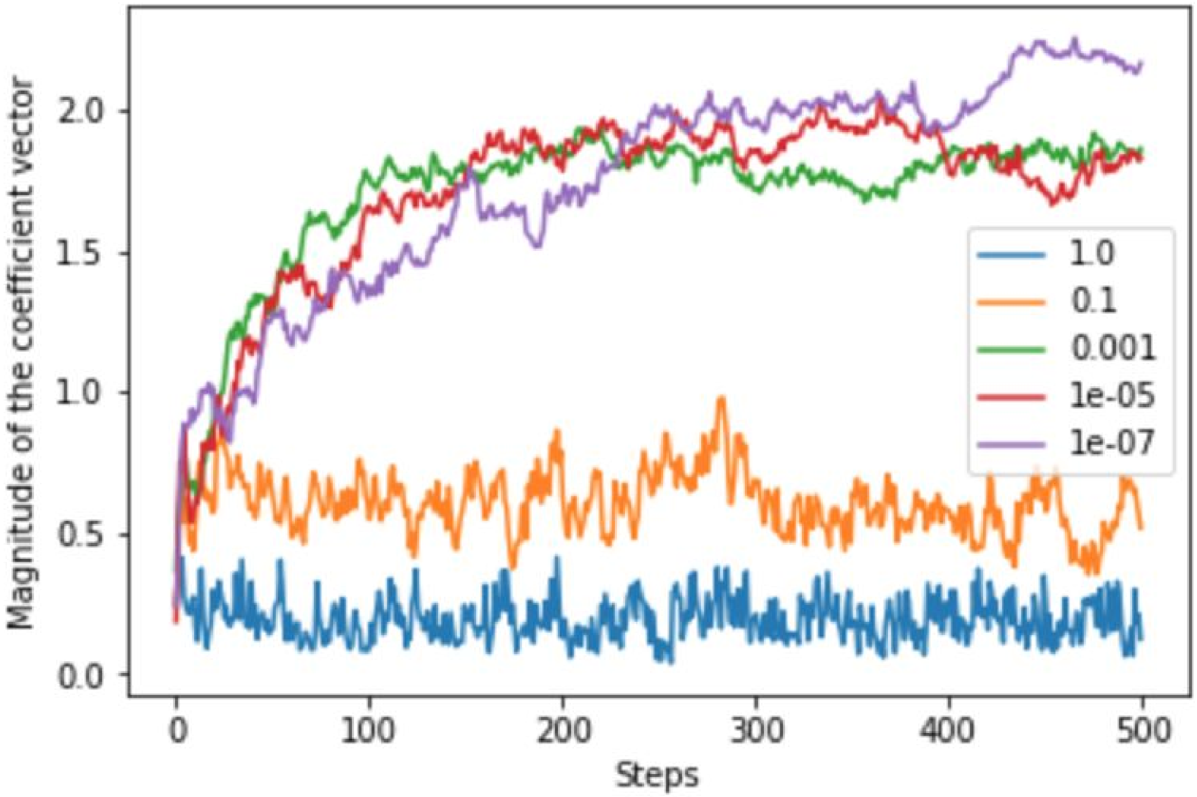
43	new	**XC**	0.79279	4	18h
44	new	**Aayush**	0.79238	17	1h
45	new	**Yijie Lu**	0.79135	11	7h
46	new	**Michael Shea**	0.79115	2	2d
47	new	**Bryant Zhao**	0.79054	9	14h

My best test dataset accuracy obtained on kaggle is 0.79279

- **Page 2:** A plot of the accuracy every 30 steps, for each value of the regularization constant. You should plot the curves for all regularization constants in the same plot using different colors with a label showing the corresponding values



- **Page 3:** A plot of the magnitude of the coefficient vector every 30 steps, for each value of the regularization constant. You should plot the curves for all regularization constants in the same plot using different colors with a label showing the corresponding values.

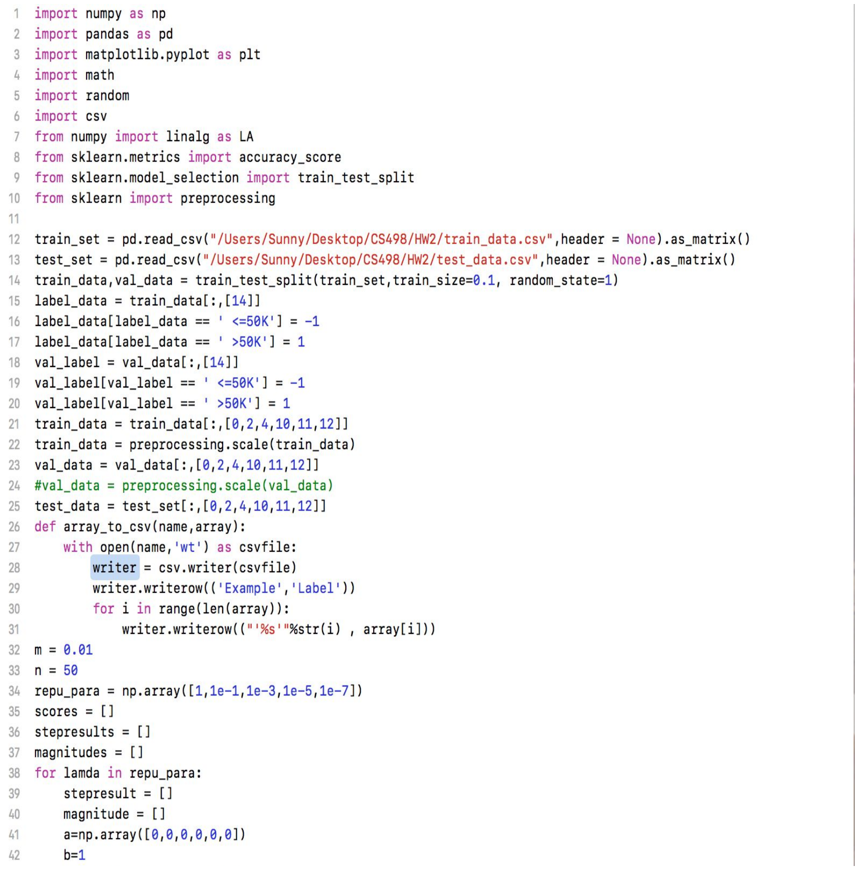


- **Page 4:** Your estimate of the best value of the regularization constant, together with a brief description of why you believe that is a good value. What was your choice for the learning rate and why did you choose it ?

My estimate of the best value of the regularization constant is 0.1
It is concluded due to the scores calculated by the validation data set multiple times
Though the accuracy score would change because of randomness, 0.1 can lead to a fairly stable accuracy

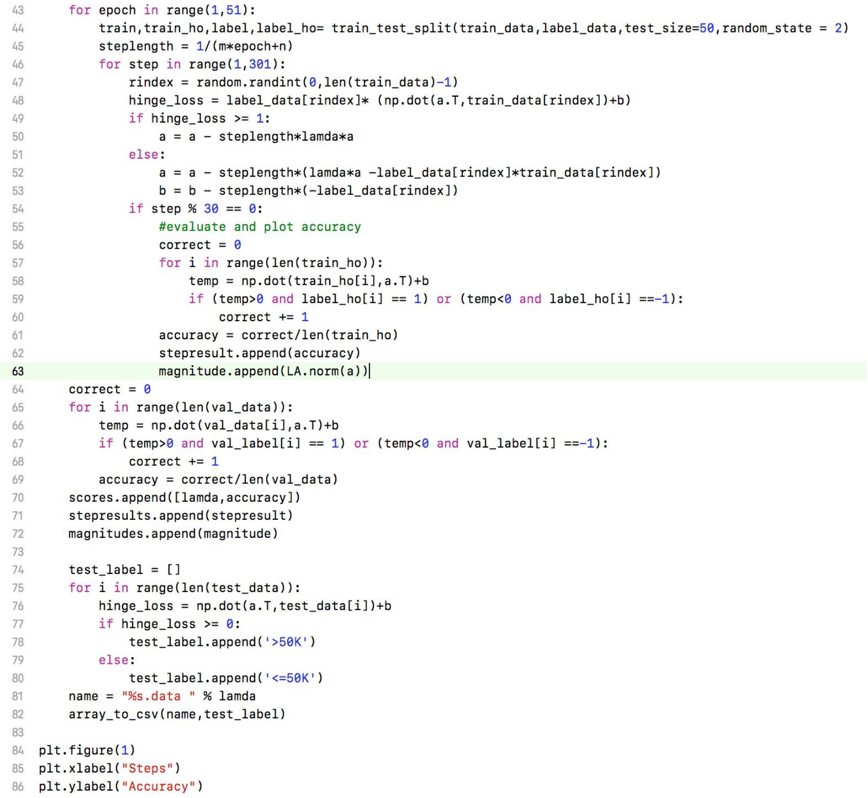
My choice for the learning rate is m = 0.01 and n = 50
I chose it because it is given in the textbook and I tried running classifier with different learning rate in a range and this learning rate gave me a fairly average good accuracy

- **Page 5:** 1 page screenshot of your code.



```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import math
import random
import csv
from numpy import linalg as LA
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn import preprocessing

train_set = pd.read_csv("/Users/Sunny/Desktop/CS498/HW2/train_data.csv",header = None).as_matrix()
test_set = pd.read_csv("/Users/Sunny/Desktop/CS498/HW2/test_data.csv",header = None).as_matrix()
train_data,val_data = train_test_split(train_set,train_size=0.1, random_state=1)
label_data = train_data[:,[14]]
label_data[label_data == ' <=50K'] = -1
label_data[label_data == ' >50K'] = 1
val_label = val_data[:,[14]]
val_label[val_label == ' <=50K'] = -1
val_label[val_label == ' >50K'] = 1
train_data = train_data[:,[0,2,4,10,11,12]]
train_data = preprocessing.scale(train_data)
val_data = val_data[:,[0,2,4,10,11,12]]
#val_data = preprocessing.scale(val_data)
test_data = test_set[:,[0,2,4,10,11,12]]
def array_to_csv(name,array):
    with open(name,'wt') as csvfile:
        writer = csv.writer(csvfile)
        writer.writerow(('Example','Label'))
        for i in range(len(array)):
            writer.writerow(("'%s'"%str(i) , array[i]))
m = 0.01
n = 50
repu_para = np.array([1,1e-1,1e-3,1e-5,1e-7])
scores = []
stepresults = []
magnitudes = []
for lamda in repu_para:
    stepresult = []
    magnitude = []
    a=np.array([0,0,0,0,0,0])
    b=1
```



```python
    for epoch in range(1,51):
        train,train_ho,label,label_ho= train_test_split(train_data,label_data,test_size=50,random_state = 2)
        steplength = 1/(m*epoch+n)
        for step in range(1,301):
            rindex = random.randint(0,len(train_data)-1)
            hinge_loss = label_data[rindex]* (np.dot(a.T,train_data[rindex])+b)
            if hinge_loss >= 1:
                a = a - steplength*lamda*a
            else:
                a = a - steplength*(lamda*a -label_data[rindex]*train_data[rindex])
                b = b - steplength*(-label_data[rindex])
            if step % 30 == 0:
                #evaluate and plot accuracy
                correct = 0
                for i in range(len(train_ho)):
                    temp = np.dot(train_ho[i],a.T)+b
                    if (temp>0 and label_ho[i] == 1) or (temp<0 and label_ho[i] ==-1):
                        correct += 1
                accuracy = correct/len(train_ho)
                stepresult.append(accuracy)
                magnitude.append(LA.norm(a))
    correct = 0
    for i in range(len(val_data)):
        temp = np.dot(val_data[i],a.T)+b
        if (temp>0 and val_label[i] == 1) or (temp<0 and val_label[i] ==-1):
            correct += 1
        accuracy = correct/len(val_data)
    scores.append([lamda,accuracy])
    stepresults.append(stepresult)
    magnitudes.append(magnitude)

    test_label = []
    for i in range(len(test_data)):
        hinge_loss = np.dot(a.T,test_data[i])+b
        if hinge_loss >= 0:
            test_label.append('>50K')
        else:
            test_label.append('<=50K')
    name = "%s.data " % lamda
    array_to_csv(name,test_label)

plt.figure(1)
plt.xlabel("Steps")
plt.ylabel("Accuracy")
```