Ryan Neumann

Dr. Rivas

Midterm

13 October 2016

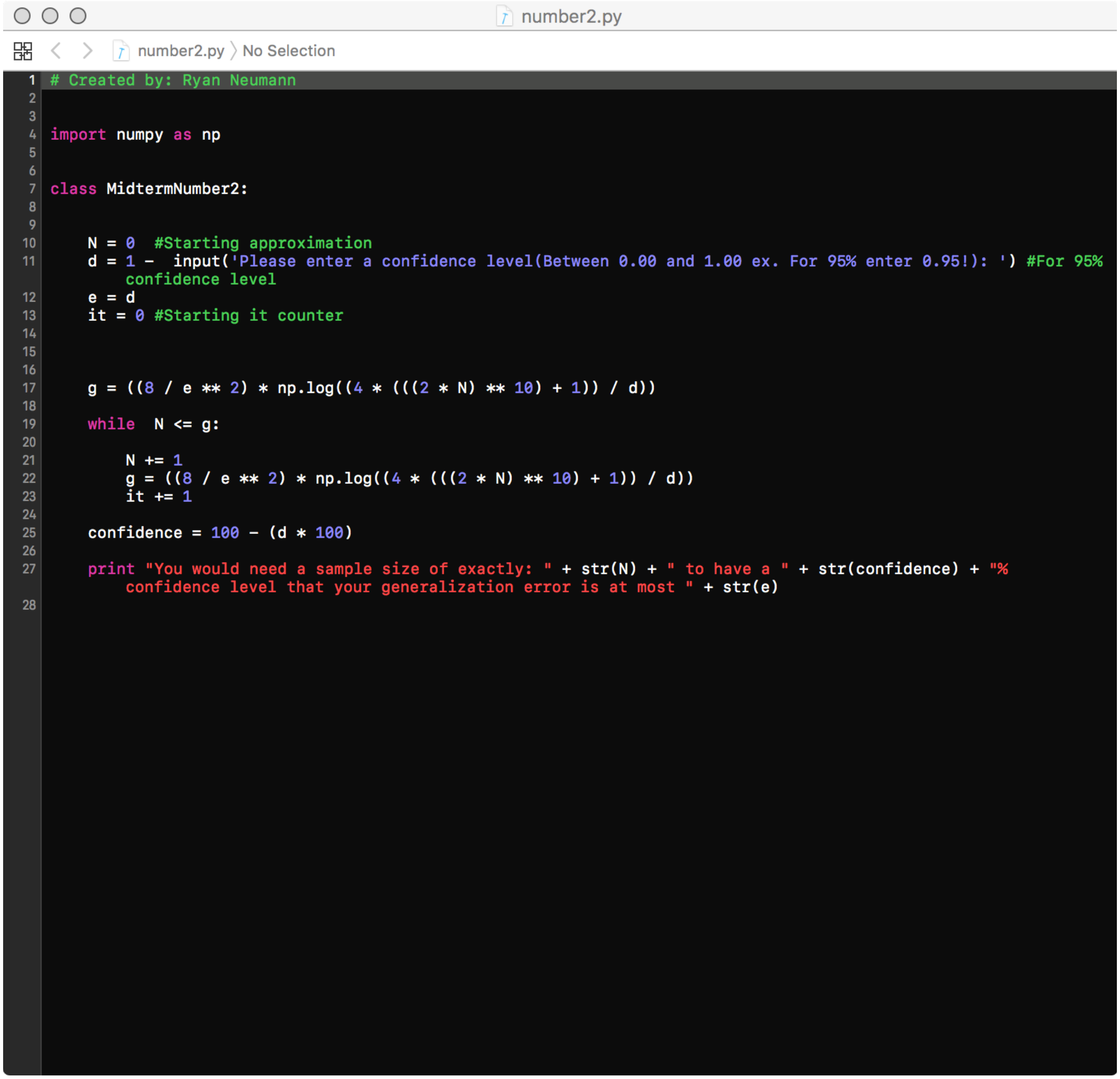
1)

a. I stopped the pocket algorithm after 5000 iterations. I ended up with a final classification error of 0.0120696749417, however, if I didn't stop the algorithm from running, it would go on forever. Starting at w = 0, the line incrementally works its way to a solution.

b. The program came to the conclusion that 0.0152242490742 was the best w. The pocket algorithm took significantly longer to find $E_{out,}$ then linear regression did. With this solution, linear regression was able to mathematically come to a conclusion on the first iteration. However, even though this algorithm is faster, it also usually has a higher rate of error. This method had the lowest amount of computational power.

c. The pocket algorithm, starting from the solution provided by linear regression, will get the closest to the actual value out of all three algorithms. Instead of starting from 0, we start from the already calculated linear regression, which already gives it an accurate starting point. It still ran through the 5000 iterations. However as stated before, if it didn't stop there it would go on forever.

2)



```
# Created by: Ryan Neumann


import numpy as np


class MidtermNumber2:


    N = 0  #Starting approximation
    d = 1 -  input('Please enter a confidence level(Between 0.00 and 1.00 ex. For 95% enter 0.95!): ') #For 95% confidence level
    e = d
    it = 0 #Starting it counter



    g = ((8 / e ** 2) * np.log((4 * (((2 * N) ** 10) + 1)) / d))

    while  N <= g:

        N += 1
        g = ((8 / e ** 2) * np.log((4 * (((2 * N) ** 10) + 1)) / d))
        it += 1

    confidence = 100 - (d * 100)

    print "You would need a sample size of exactly: " + str(N) + " to have a " + str(confidence) + "% confidence level that your generalization error is at most " + str(e)
```

3)