Rayan Hassan
4511021

ECE 1395 – Homework 1 report

1) Regression problem: Predict body fat percentage
   a) We could use features like body mass index (BMI) or waist to hip ratio
   b) Y would be the body fat percentage
   c) A sample of people, collect information about their BMIs and/or waist to hip ration, and note their corresponding body fat percentage
   d) It might be challenging because we might need more features to predict body fat percentages. Everyone is different, so people that have different BMIs and/or waist to hip ratios might have the same body fat percentage and vice versa.

2) Classification problem: Predict if a fruit is an apple or pear.
   a) We could use features like size, colour, shape, weight, etc.
   b) Labels would be "apple" (or '0') and "pear" ('1')
   c) Collected data could be [red, green, peach] for colour, [height, diameter, etc.] for size, [spherical, pear-shaped] for shape, etc.
   d) Challenges: apples and pears can have the same colours, sizes, weights, etc. Also, it is hard to tell the algorithm how to detect their shapes.

3) a) The following screenshot shows vector 'x' of random values from a Gaussian distribution with mean 2.1 and standard deviation 0.7.

```
In [7]: runfile('C:/Users/RAYAN/OneDriv
OneDrive/Desktop')
Vector x is:

[[1.15071731]
 [1.65719591]
 [2.49184362]
 ...
 [2.86616449]
 [2.30161105]
 [1.32264072]]
```

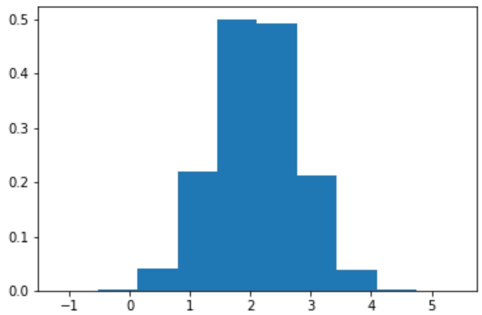
   b) The following screenshot shows vector 'x' of random values from a uniform distribution from [-1.5, 4]

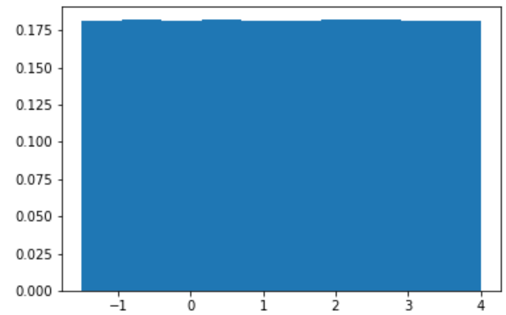
```
Vector z is:

[[ 1.50296866]
 [-0.88320024]
 [ 0.90875934]
 ...
 [ 2.37259103]
 [ 3.04691194]
 [ 2.16367628]]
```

Rayan Hassan
4511021

c) Here are the normalized histograms of vectors x on the left (ps1-3-c-1.png) and z on the right (ps1-3-c-2.png)
As expected, the histogram of x looks like a Gaussian (normal) distribution and the one for z like a uniform distribution.





d) The execution time for adding 1 to every value in x using a loop is ≈ 1.209 seconds as shown.

```
Execution time (using a loop):
1.2092383999997764

Vector x after adding 1 to every value:

[[2.15071731]
 [2.65719591]
 [3.49184362]
 ...
 [3.86616449]
 [3.30161105]
 [2.32264072]]
```

e) The execution time for adding 1 to every value in x without using a loop is ≈ 0.0028 seconds.

```
Execution time (without a loop):

0.0027958999999100342

Vector x after adding 1 to every value:

[[2.15071731]
 [2.65719591]
 [3.49184362]
 ...
 [3.86616449]
 [3.30161105]
 [2.32264072]]
```

Without using a loop, the code executes faster (lower execution time), so it is more efficient that way.

Rayan Hassan
4511021

f) The number of elements retrieved in vector y is 181977 as shown below

```
Number of retreived elements in y:  181977
```

After rerunning the code two times, we get the values 181635 and 181299

```
 [3.3286229 ]]

Number of retreived elements in y:  181635
```

```
Number of retreived elements in y:  181299
```

The numbers are different but they are close, since we are generating random number from

[-1.5, 4] in vector z, so distance from -1 to 0 (which is the range of the elements in vector y] is much smaller that the range for the entire vector z, and much larger than that of positive numbers in z.

4) a) The following shows all the required outputs. Matrix B has elements that are the square of the corresponding elements in A.

```
The matrix A is:

[[ 2  1  3]
 [ 2  6  8]
 [ 6  8 18]]

The following list shows minimum value in each column:
[2 1 3]

The following list shows maximum value in each row:
[ 3  8 18]

The smallest value in A is:
1

The following list shows sum of each row in A:
[ 6 16 32]

The sum of all elements in A is:
54

The matrix B is:
[[  4   1   9]
 [  4  36  64]
 [ 36  64 324]]
```

Rayan Hassan
4511021

b) The following shows the solution to the system of equalities. As expected, the solutions are {-0.2, 0, 0.4}

```
Solve system 2x+y+3z=1, 2x+6y+8z=2, 3x+5y+15z=5
First matrix:
[[ 2  1  3]
 [ 2  6  8]
 [ 3  5 15]]

Output matrix:
[1 2 5]


The solution to the system is:
 [ 1.20274161e-16 -2.00000000e-01  4.00000000e-01]
```

c) **By hand**

For x1: L1 norm is: $0.5 + |-1.5| = 2$

L2 norm is: $\sqrt{0.5^2 + 1.5^2} = 1.5811$

For x2: L1 norm is: $1 + |-1| = 2$

L2 norm is: $\sqrt{1^2 + 1^2} = \sqrt{2} = 1.414$

Using Python

```
Vector x1 is:  [ 0.5  0.  -1.5]

Norm L1 of x1 is:  2.0

Norm L2 of x1 is:  1.5811388300841898

Vector x2 is:  [ 1 -1  0]

Norm L1 of x2 is:  2.0

Norm L2 of x2 is:  1.4142135623730951
```

5) These are the tested input matrices for the function sum_sq_col and their corresponding output

```
Input matrix A1 is:
[[2 3 4]
 [2 2 2]
 [4 1 3]
 [6 5 3]]

Output matrix B1 is:
[[29]
 [12]
 [26]
 [70]]

Input matrix A2 is:
[[ 1  6  3]
 [ 3 10  2]]

Output matrix B2 is:
[[ 46]
 [113]]
```