

Hayden Feddock

ECE 1395 – Dr. Dallal

1/20/2023

All of my code and responses are in the single ps1.py file. That file can be ran as a normal python3 file would be, however you may need to install the following libraries: numpy, matplotlib, and potentially timeit (however this may be included in the default libraries so you may already have this one). I found the wheel files of the latest releases of these online on the pypi.org website, but I just installed them onto my system using pip which made it really easy.

Section 1: Conceptual Question

Propose a new regression problem that you can solve with ML and describe how you would go about solving it

Problem: Predict the traffic in South Oakland based on the time.

I would use the features (x) for the time of day.

I would use the labels (y) for the avg vehicle's mph under the speed limit, so that larger differences in mph will reflect higher traffic.

I could collect data on the speed of each car using gps (on individuals phones), or I could set up speedometers to check the vehicles speed on each street.

Tracking the speed of each car using gps could be an invasion of privacy and would be difficult to write the code for, and setting up lots of speedometers could get expensive and be affected by other moving objects.

Section 2: Conceptual Question 2

Propose a new classification problem that you can solve with ML and describe how you would go about solving it

Problem: Predict whether I'll eat breakfast based on when I wake up.

I would use the features (x) for the time that I wake up.

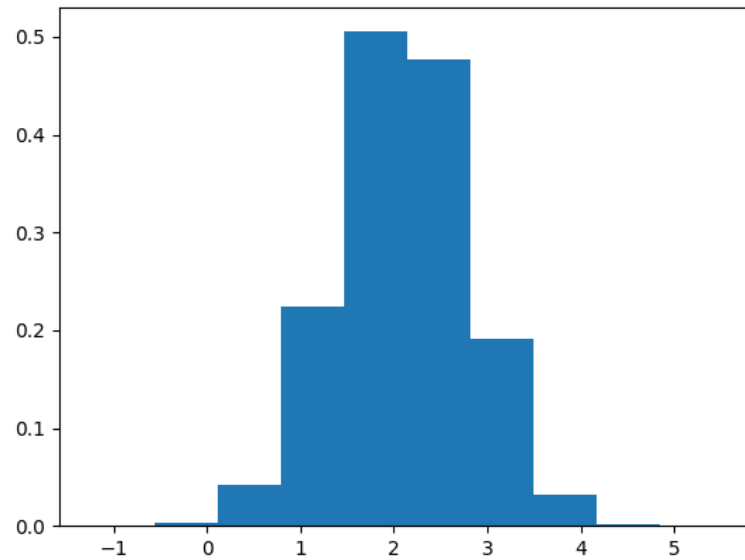
I would use the labels (y) for whether or not I eat breakfast.

I could collect data at the time that I wake up and then whether or not I ate breakfast that morning.

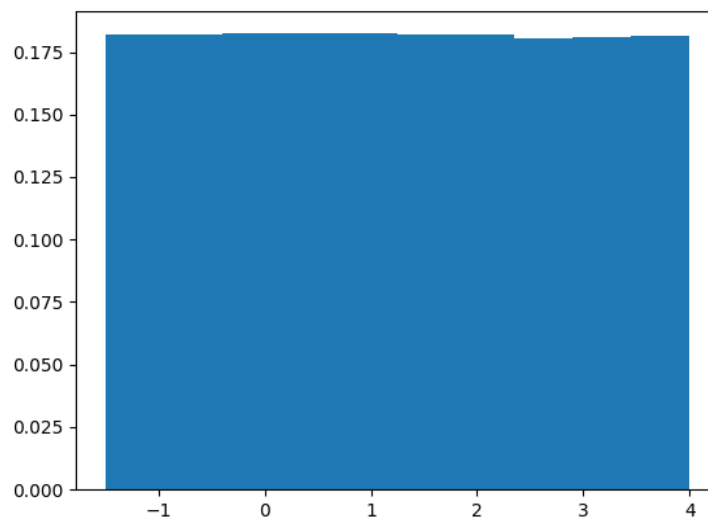
There could be other factors that affect whether or not I eat breakfast such as rushing for an early class, or I could forget to record whether or not I ate.

Section 3: Basic Operations

```
# a. Generate a 1,000,000 by 1 vector x of random numbers from a Gaussian  
(normal) distribution with a mean of 2.1 and a standard deviation of 0.7
```



```
# b. Generate a 1,000,000 by 1 vector z of random numbers from a uniform  
distribution between [-1.5 4]
```



```
# c. Plot the normalized histogram of vectors x and z. Store them as png images.
```

```

# The histogram for vector x does look like a Gaussian distribution because it
has a bell-like curve which is a characteristic of a Gaussian distribution.

# The histogram for vector z does look like a uniform distribution because it
almost forms a flat line which means that there is a near equal (uniform) value
across each bin.

# d. Add 1 to every value in x using a loop and calculate process time
Array increment with loop process time: 3.7585808000003453
# The processing time of incrementing every value of the vector using a loop took
3.758... seconds and appears to usually come in around 4 seconds.

# e. Add 1 to every value in x without using a loop and calculate process time
Array increment without loop process time: 0.004237000001012348
# The processing time of incrementing every value of the vector without using a
loop took 0.00424... seconds and is much faster and therefore more time efficient
than the loop method.

# f. Define vector y with values between 0 and -1 from vector z
Dimensions of vector y: (182250, 1)
# The number of elements between 0 and -1 that I retrieved from vector z and
copied to y was 182,250. Upon running this code two more times that number of
retrieved elements is not exactly the same however it is similar.
# Since vector z uses a uniform distribution model and ideally the number of
elements in each bin (range of values) is the same after each run, so it would
make sense that the values after each run are not identical, but similar.

### Section 4: Linear Algebra ###

# a. Define matrix A without using loops.
Matrix A is:
[[ 2  1  3]
 [ 2  6  8]
 [ 6  8 18]]
# Find the min of each column in A
Column 0 has min: 2
Column 1 has min: 1
Column 2 has min: 3
# Find the max of each row in A
Row 0 has max: 3
Row 1 has max: 8
Row 2 has max: 18
# Find the min value overall of A
The overall min is: 1

```

```

# Find the sum of each row in A
Row 0 has sum: 6
Row 1 has sum: 16
Row 2 has sum: 32
# Find the sum of all elements in A
The overall sum is: 54
# Compute matrix B which is the matrix A with an element-wise square
Matrix B is:
[[ 4  1  9]
 [ 4 36 64]
 [36 64 324]]

# b. Solve the system of linear equations
The value of x is: [4.81096644e-16]
The value of y is: [-0.2]
The value of z is: [0.4]

# c. Analytically compute the L1 and L2 norms for vectors x1 and x2
# L1 norm for vector x1
    # Step 1: Take the absolute value of the first element
    #  $|0.5| = 0.5$ 
    # Step 2: Take the absolute value of the second element and add it to the
first element
    #  $0.5 + |0.0| = 0.5$ 
    # Step 3: Take the absolute value of the third element and add it to the
total
    #  $0.5 + |-1.5| = 2$ 
# Check L1 norm
L1 norm for vector x1: 2.0
# L2 norm for vector x1
    # Step 1: Square the value of the first element
    #  $(0.5)^2 = 0.25$ 
    # Step 2: Square the value of the second element and add it to the first
element
    #  $0.25 + (0.0)^2 = 0.25$ 
    # Step 3: Square the value of the third element and add it to the total
    #  $0.25 + (-1.5)^2 = 2.5$ 
    # Step 4: Square root the total
    #  $(2.5)^{(1/2)} = 1.581...$ 
# Check L2 norm
L2 norm for vector x1: 1.5811388300841898

# L1 norm for vector x2
    # Step 1: Take the absolute value of the first element
    #  $|1| = 1$ 

```

```

    # Step 2: Take the absolute value of the second element and add it to the
first element
    #  $1 + |-1| = 2$ 
    # Step 3: Take the absolute value of the third element and add it to the
total
    #  $2 + |0| = 2$ 
# Check L1 norm
L1 norm for vector x2: 2.0
# L2 norm for vector x2
    # Step 1: Square the value of the first element
    #  $(1)^2 = 1$ 
    # Step 2: Square the value of the second element and add it to the first
element
    #  $1 + (-1)^2 = 2$ 
    # Step 3: Square the value of the third element and add it to the total
    #  $2 + (0)^2 = 2$ 
    # Step 4: Square root the total
    #  $(2)^{(1/2)} = 1.414...$ 
# Check L2 norm
L2 norm for vector x2: 1.4142135623730951

### Section 5: Write a Function ###

# Test function with matrix 1
    # Input:
    # [[5, 2, 3],
    #  [7, 4, 2],
    #  [1, 0, 8]]
Matrix m1 is:
[[5 2 3]
 [7 4 2]
 [1 0 8]]
    # Output:
    # [[38],
    #  [69],
    #  [65]]
Column of sums of squared rows b1 is:
[[38.]
 [69.]
 [65.]]

# Test function with matrix 2
    # Input:
    # [[1, 8, 2, 4],
    #  [3, 9, 0, 1],

```

```
        # [4, 9, 7, 0]]
Matrix m2 is:
[[1 8 2 4]
 [3 9 0 1]
 [4 9 7 0]]
    # Output:
        # [[85],
        #  [91],
        #  [146]]
Column of sums of squared rows b2 is:
[[ 85.]
 [ 91.]
 [146.]]
```