CMSC 436: Artificial Intelligence
Fall 2024, Instructor: Dr. Milos Manic,
http://www.people.vcu.edu/~mmanic **_Project 1_**

CMSC 436: Artificial Intelligence
Project No. 1
Due Wednesday, September 18, 2024, noon

Student certification:
Team member 1:
Print Name: __Alex Welk__________________________ *Date:* **9/16/2024**_________________
I have contributed by doing the following: Wrote the code for normalizing, plotting, and analyzing our dataset
Signed: __RAW_____ *(you can sign/scan or use e-signature)*

Team member 2:
Print Name: ______Ethan Scott____________________ *Date:* ___9/16/2024____
I have contributed by doing the following: _Helped answer questions and created truth table_______
Signed: ____Ethan Scott___________ *(you can sign/scan or use e-signature)*

Team member 3:
Print Name: ___________________________________ *Date:* __________________ *I have contributed by doing the following:* ____ *Signed:* ____ *(you can sign/scan or use e-signature)*

Pr.1

A) Understand and explore a data set (10 pts)

Three data sets (set A, B, and C) have been created following normally distributed classes. These data sets provide examples of car models where:

- The first column represents the cost in USD.
- The second column represents the weight in pounds.
- The third (last) column corresponds to the type (0 for small, 1 for big car).

Each data set contains 4,000 samples.

For each data set, do the following:

1. Normalize the data, then plot the data using software of your choice (for two vehicle types). (2 points) (resource to understand data normalization)

Normalized GroupA.txt

```
normalized cost: [[0.74852773]
 [0.85250268]
 [0.71946409]
 ...
 [0.17080695]
 [0.17823106]
 [0.24564387]]
normalized weight: [[0.72769325]
 [0.64208772]
 [0.67234462]
 ...
 [0.22261497]
 [0.17925974]
 [0.37604494]]
```

Normalized GroupB.txt

```
normalized cost: [[0.57917031]
 [0.67454132]
 [0.73591308]
 ...
 [0.19821815]
 [0.26122069]
 [0.3498881 ]]
normalized weight: [[0.81915599]
 [0.56229673]
 [0.64651489]
 ...
 [0.32573415]
 [0.37929542]
 [0.18818731]]
```

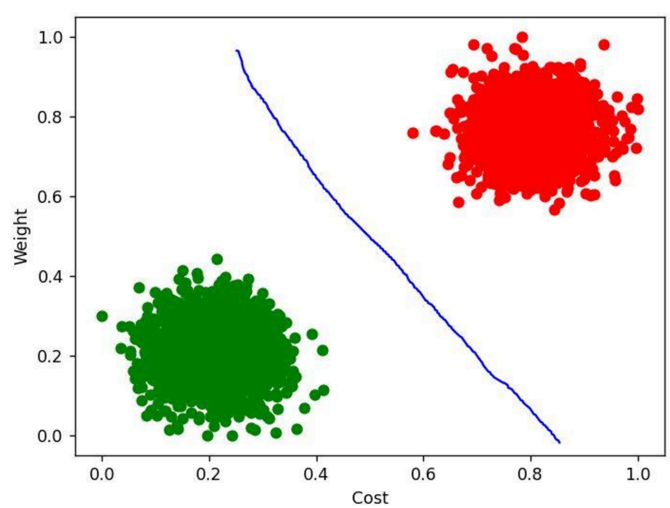
Normalized GroupC.txt

```
normalized cost: [[0.74663625]
 [0.77527293]
 [0.3764466 ]
 ...
 [0.5827599 ]
 [0.42593684]
 [0.33518965]]
normalized weight: [[0.73633296]
 [0.56627434]
 [0.44994154]
 ...
 [0.32442605]
 [0.38809045]
 [0.60343531]]
```

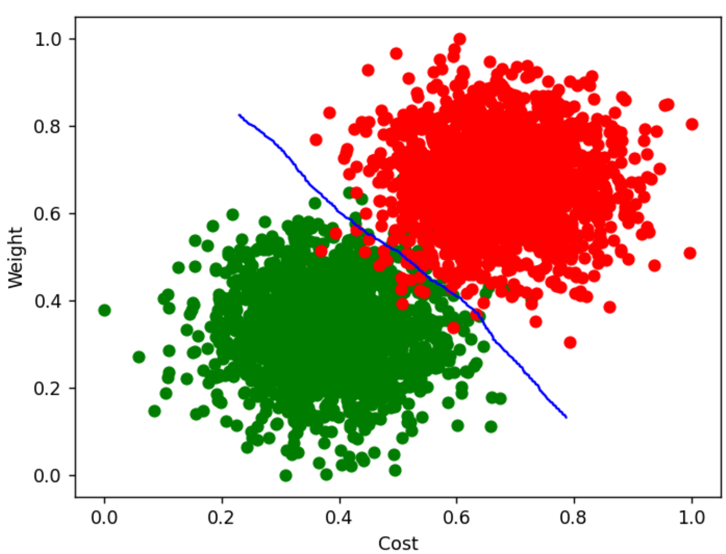
These are just small parts of the normalized data that are outputted when I print the list.

2. Estimate a separation line and draw it manually (by hand) on that plot. This line will be a linear separator, which separates "small" cars from "big" cars. At this time, no running algorithm is needed (we will do that in the next assignment). (1 point)

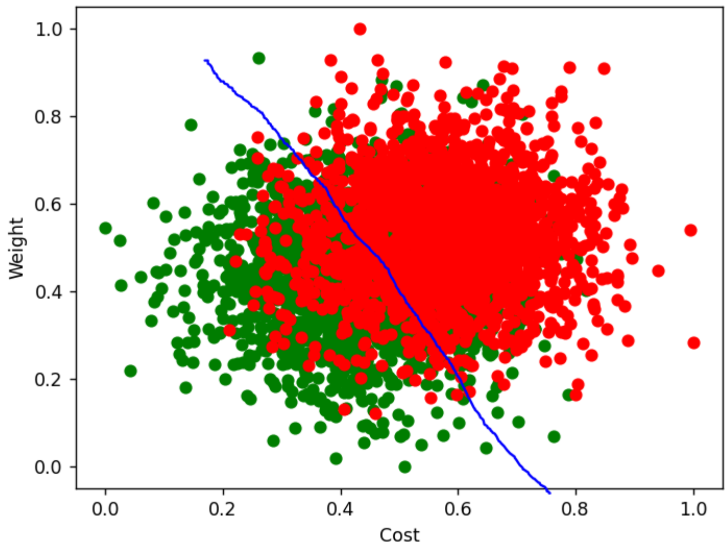
GroupA.txt Graph with Estimated Linear Separator



GroupB.txt Graph with Estimated Linear Separator



GroupC.txt Graph with Estimated Linear Separator



3. Determine the mathematical definition of this linear separator. Based on this linear separator (equation), determine the inequality that selects "big" cars. What are the weights and threshold? Comment. (2 points)
    a. Important: inequality defines neuron's functionality (think of the inequality we covered in Session 04&05). Inequality "makes" the decision (chooses a portion of xOy space here). Deciding on inequality means determining weights and threshold of a neuron.

- GroupA.txt Linear Separator: -8.776 * Cost + -7.942 * Weight = -8.280
    - Cars Inequality: -8.776 * Cost + -7.942 * Weight < -8.280
    - Trucks Inequality: -8.776 * Cost + -7.942 * Weight > -8.280
- GroupB.txt Linear Separator: -11.853 * Cost + -12.882 * Weight = -12.606
    - Cars Inequality: -11.853 * Cost + -12.882 * Weight < -12.606
    - Trucks Inequality: -11.853 * Cost + -12.882 * Weight > -12.606
- GroupC.txt Linear Separator: -6.930 * Cost + -4.337 * Weight = -5.592
    - Cars Inequality: -6.930 * Cost + -4.337 * Weight < -5.592
    - Trucks Inequality: -6.930 * Cost + -4.337 * Weight > -5.592

4. Provide a confusion matrix (false positives, false negatives, etc.) (1 point)

- Confusion Matrix for GroupA.txt

  [378   0]
  [0 422]
- Confusion Matrix for GroupB.txt

  [375   3]
  [7 415]
- Confusion Matrix for GroupC.txt

  [280 98]
  [107 315]

5. Calculate accuracy, error, true positive rate and true negative rate, false-positive rate, and false-negative rate. (Note: the true positive rate is different from the true positive). (2 points)

- GroupA.txt Metrics:
    - Accuracy: 1.0
    - Error Rate: 0.0
    - True Positive Rate: 1.0
    - True Negative Rate: 1.0
    - False Positive Rate: 0.0

    - False Negative Rate: 0.0
- GroupB.txt Metrics:
    - Accuracy: 0.9875
    - Error Rate: 0.012499999999999956
    - True Positive Rate: 0.9834123222748815
    - True Negative Rate: 0.9920634920634921
    - False Positive Rate: 0.007936507936507936
    - False Negative Rate: 0.016587677725118485
- GroupC.txt Metrics:
    - Accuracy: 0.74375
    - Error Rate: 0.25625
    - True Positive Rate: 0.7464454976303317
    - True Negative Rate: 0.7407407407407407
    - False Positive Rate: 0.25925925925925924
    - False Negative Rate: 0.25355450236966826

6. Compare results for each data set and explain the differences. How are these datasets different? Why was data normalization helpful? (2 points)

Each of the training sets differ by being slightly less defined with each grouping as you progress from A to C. A is perfectly accurate, B has some errors, and C has the most errors. Normalization helps because it gives every value a place on a scale with a known range rather than an arbitrary set of values of unknown range. This allows models to interpret and converge more quickly and accurately. Normalization also allows for better database organization, reduced redundancy, data consistency, and makes data easier to work with.

CMSC 436: Artificial Intelligence

Fall 2024, Instructor: Dr. Milos Manic,

http://www.people.vcu.edu/~mmanic ***Project 1***

Note: An example of true positive: the class is “it is a big car” and prediction is “big car”

B) **McCulloch-Pitts neurons (5 pts)**

1. Create a truth table for the artificial neuron below. What is the **functionality of** this neuron? (3 points)

A	B	C	net = -3A + 2B + C	inequalities	out
0	0	0	0	0 > T; T = -1	1
0	0	1	1	1 > T	1
0	1	0	2	2 > T	1
0	1	1	3	3 > T	1
1	0	0	-3	-3 < T	0
1	0	1	-2	-2 < T	0
1	1	0	-1	-1 <= T	0
1	1	1	0	0 > T	1

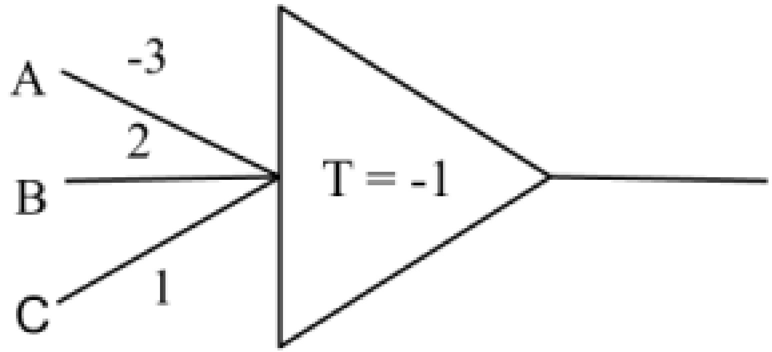
The functionality of this neuron is to determine when to fire given 3 inputs. This specific neuron puts a large negative weight on A which makes it so B and C need to both be present while A is present for the neuron to fire; without A present the neuron will fire regardless of whether or not B and/or C are present. This makes A a deciding force in the neuron.

2. Given the same set of weights and the determined functionality of a neuron, what would be the **range of possible values for threshold**? (2 points)

Given that the range of values for net is -3 to 3, the range of possible values for the threshold is -3<= T <= 3

Note: Consider unipolar hard threshold activation function (possible inputs/outputs are obviously 0 & 1). Always start with the unit definition (net, output). The functionality of the neuron is a Boolean function.

Hint: The truth table (similar to the one in class) should present inequalities that will evidence the functionality of a neuron (prove that it works as promised).



Note:

1. Compile all your deliverables into a single **pdf** file - this is YOUR REPORT.
2. Archive/zip report with other pertinent files (code, data, other).
3. Submit your file (Canvas). Please name the zip file as GroupName_Project1.zip.