Data Science: Artificial Intelligence (BU.920.624)

Group Assignment #1

Due on November 6th via Canvas

Group Members: Mohammed Bo Khamseen, Bonnie Koo, Pranab Sharma, Pooja Singh

---

1. **AI Model Development Using No-Code Tools for Image Classification**

- **Step 1 (Problem):**
  The problem to be solved is detecting if the vehicle captured is allowed to be in the captured road. For example, this can be used in park roads where only bicycles are allowed, and no buses are allowed or other roads where no cars are allowed. This can be done by classifying the type of vehicle (Bicycle, Car, Bus, Motorcycle). It's relevant in the business context as we can issue fines on their vehicles and collect revenue for the park

- **Step 2 (Dataset):**
  The dataset im using is the MIO-TCD dataset. The dataset is acquired at different times of the day and different periods of the year by thousands of traffic cameras deployed all over Canada and the United States.
  The original dataset Contains 648,959 images divided into 11 categories:

  1. Articulated truck
  2. Bicycle
  3. Bus
  4. Car
  5. Motorcycle
  6. Non-motorized vehicle
  7. Pedestrian
  8. Pickup truck
  9. Single unit truck
  10. Work van
  11. Background

  For this exercise, I took a subset of the dataset focusing on 4 categories

  1. Bicycle (50 Pictures)
  2. Bus (50 Pictures)
  3. Car (50 Pictures
  4. Motorcycle (50 Pictures)

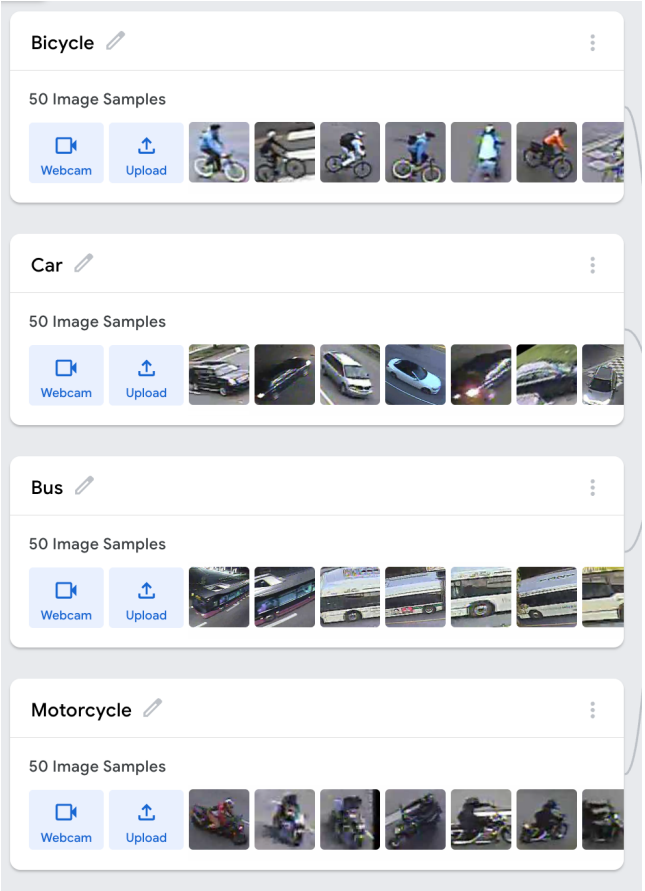
  I've also copied a testing set of 15 picture from each class

- **Step 3 (Model Training)**

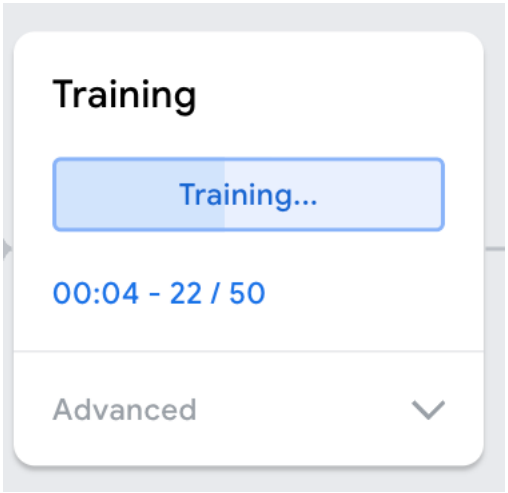
  I'm using Teachable Machines as the selected tool.

  Model Link

  I've created the four classes and uploaded the training images

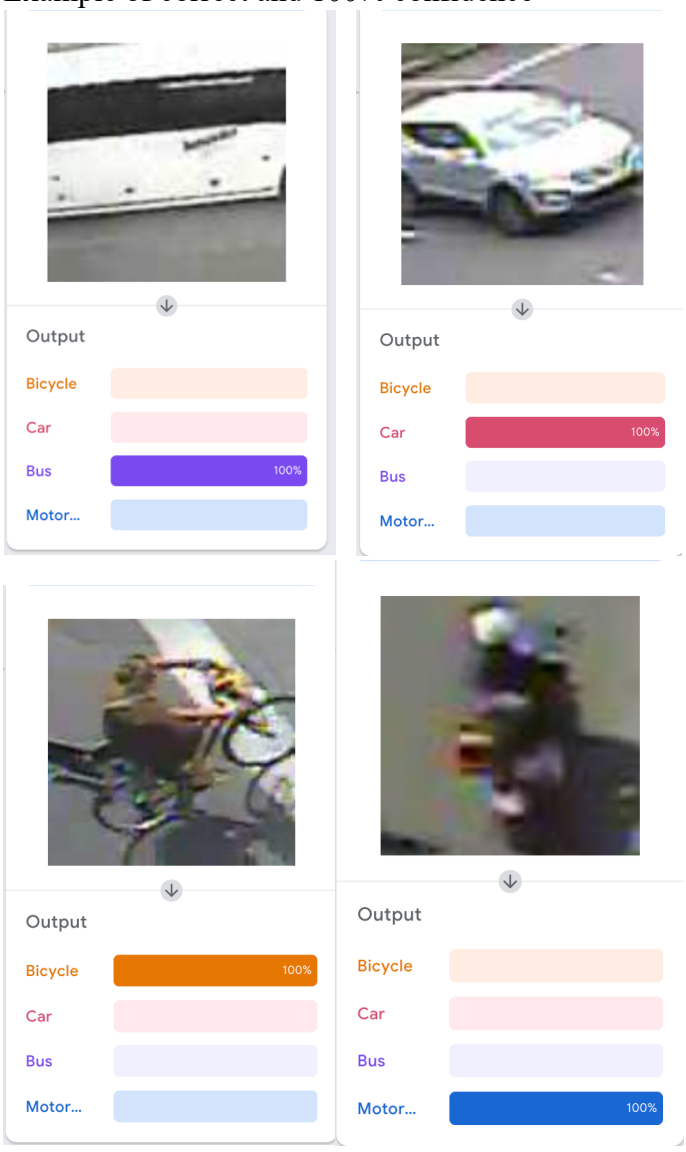


I've trained the model

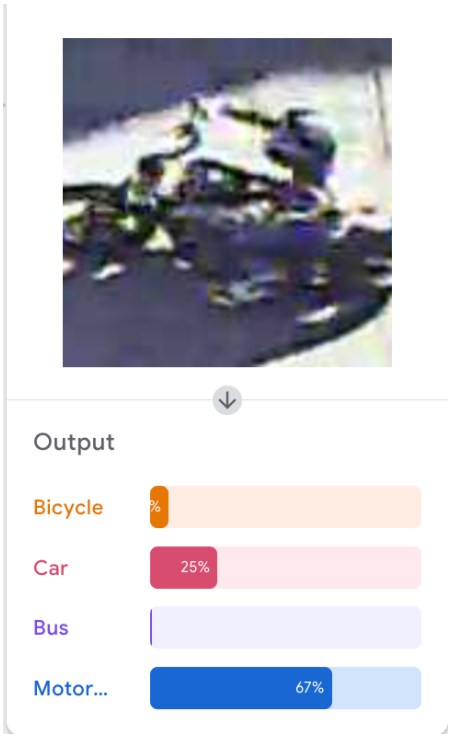


The model performed very accurately, it answered most pictures correctly with 100% confidence. Some pictures were wrong or not with full confidence.

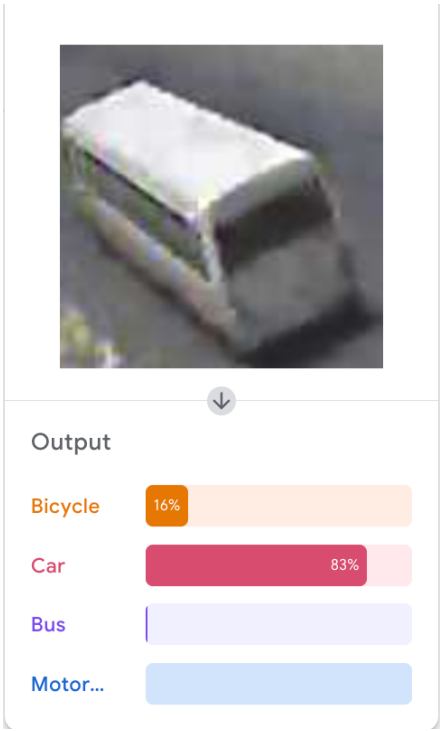
Example of correct and 100% confidence



Example of correct but not full confidence. 67% motorcycle, 25% car



Example of incorrect answer. Picture identified as car instead of bus



- **Step 4** (Result Reflection):

The model performed generally accurate. There were no challenges in training the model. One challenge in obtaining accurate classification in this model is that the pictures are in different angles. It's hard to understand a bus from behind vs sideways. This challenge might be mitigated with enough pictures.

The model could be improved by putting more pictures from the dataset. It is helpful to train the model on the same angle it would be used (i.e., if we plan to use this model on an overhead camera looking at cars from behind, we should train it with pictures from that position)

This model can be used in Parks where they have specific roads that cars/buses should not enter, and only bicycles are allowed. the model will alert the park ranger if some car entered the road and the ranger can go and issue a citation to them. If the model is more advanced, we can even issue the citation automatically by detecting the license plate.

2. **Understanding Linearly Classifiable vs. Non-Linearly Classifiable Datasets.**

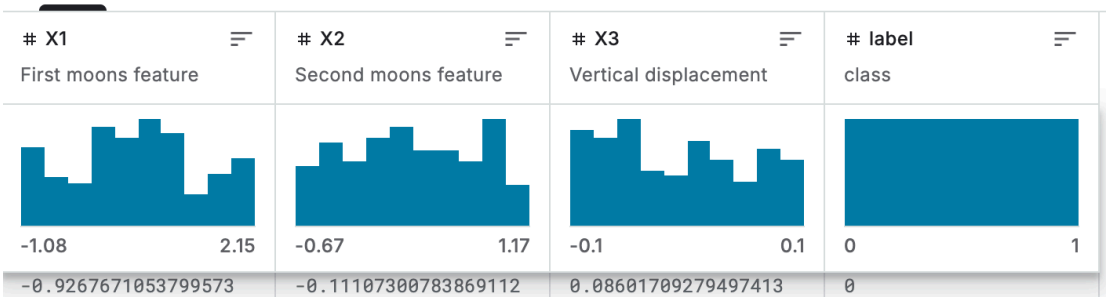
   What is a linearly classifiable dataset? What is a non-linearly-classifiable dataset? Provide an example dataset that is not linearly classifiable. You may plot your dataset without listing each data point.

   Linearly classifiable data is a dataset, in which data points can be classified using a single straight line in 2D or a hyperplane in higher dimensions. Non-linearly classifiable dataset will be a dataset in which a single straight line or hyperplane cannot completely separate the data points.

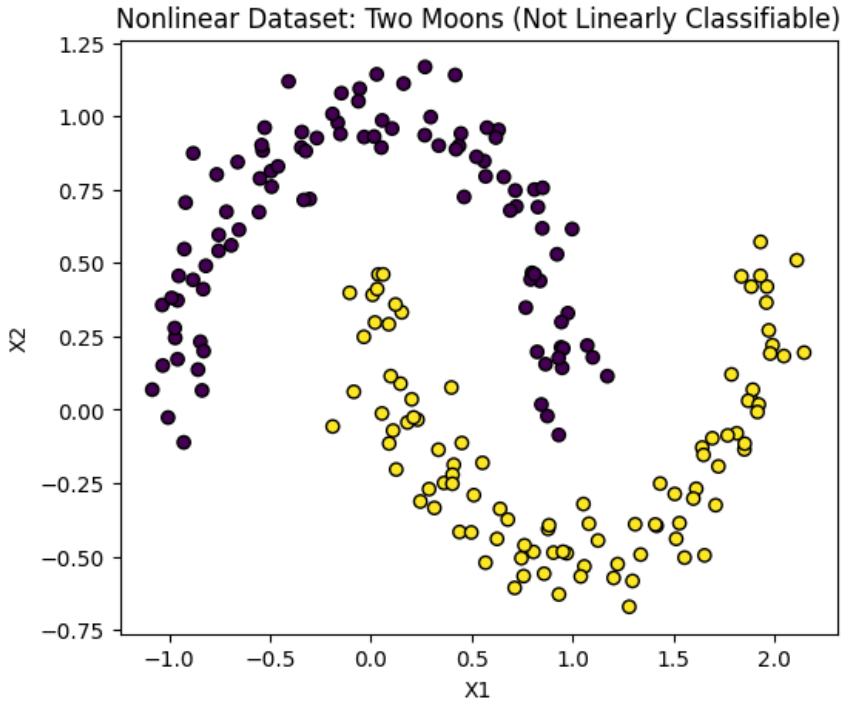
   MoonDataset from Kaggle (https://www.kaggle.com/datasets/emadmakhlouf/linearly-inseperable-dataset)

moonDataset.csv (12.39 kB)

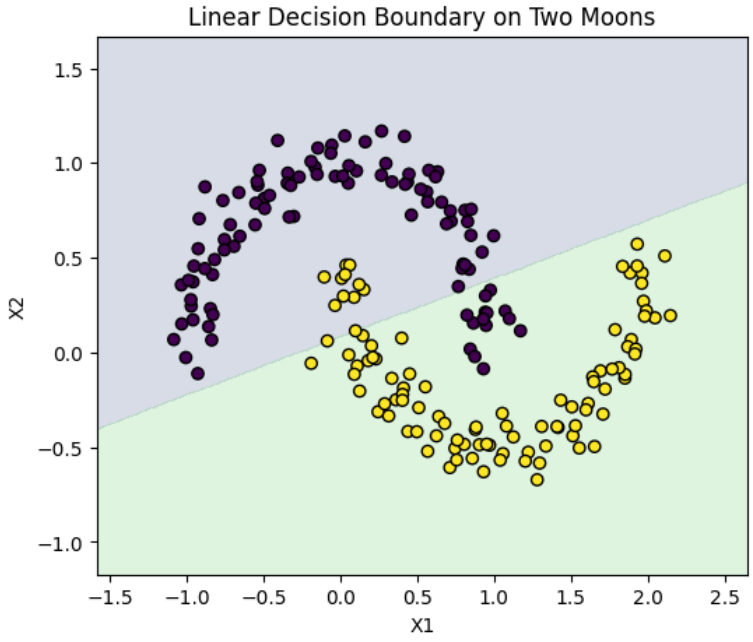
Detail Compact Column



# X1 First moons feature	# X2 Second moons feature	# X3 Vertical displacement	# label class
-1.08 … 2.15	-0.67 … 1.17	-0.1 … 0.1	0 … 1
-0.9267671053799573	-0.11107300783869112	0.08601709279497413	0
-0.9175829244842931	0.706006215332596	0.05804064197611089	0
0.43798448453784156	0.899092529285375	0.07254294595290847	0
0.0896938727056683	0.29144613684431875	0.07044433219140955	1
0.11067169363473918	-0.07080569263564954	-0.0903758825121655	1
-0.05472229722846312 6	1.094161850351758	-0.02121250219828693 7	0
0.5628647748239441	0.8474757124177922	0.002737869343484744 5	0
0.5691548048568432	-0.5210796391777477	-0.06896611768740618	1
-0.06095574669174167	1.051386074011063	0.005245978286416666 5	0
0.1539947810963499	0.3320025272995635	-0.08020939168326358	1
0.9772591620027481	0.3291792892723379	-0.07409126382698346	0
-0.9694355496435786	0.24478900633625386	-0.03195908393587468	0
0.866743016352622	0.1554577462227999	0.03811589521567804	0
0.4517879362424243	-0.11305878469245469	-0.020588900014827022	1
1.6936279710131528	-0.09694563323649873	0.013696621095748734	1

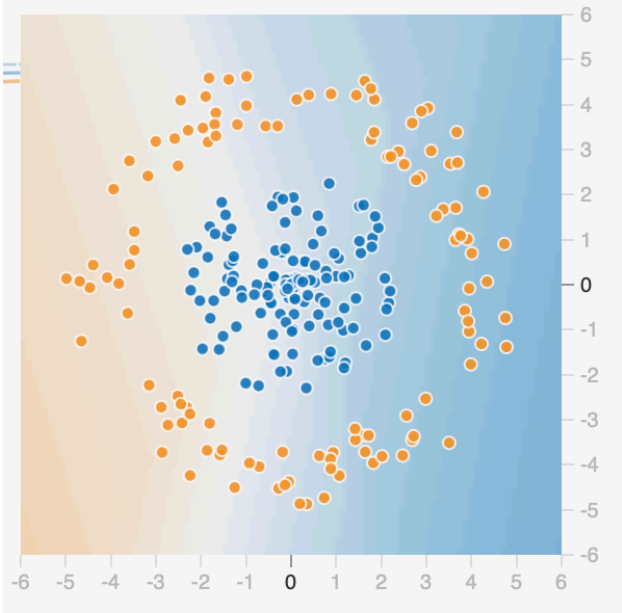


The graph below shows that the MoonDataset cannot be classified by a linear boundary.



3. **Building and Describing a Neural Network for Non-Linear Classification.**

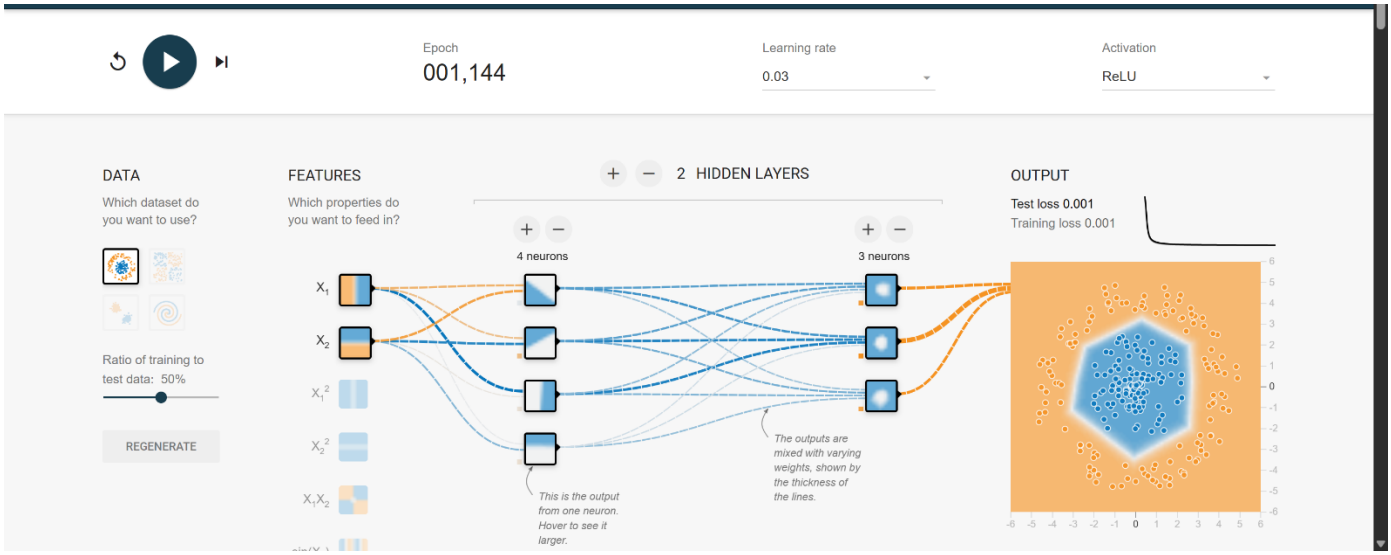
Consider the following training dataset from TensorFlow Playground (https://bit.ly/annga1):



The orange points are labeled with class label 0, and the blue points are labeled with class label 1. Using the TensorFlow Playground tool (see the link above), train a feedforward neuron network model. Write a mathematical expression using linear layers and ReLU activations that represents the structure of your model.

Analysis:

Tensor flow view after feedforward neuron network model training.



During feedforward neuron network model training we:

1. Changed neurons per layer to 4 and 3,
2. Activation function from Than to ReLU
3. Test loss changed from 0.516 to 0.001 post training

Mathematical equation –

$$\hat{y} = \sigma(W_3\ \mathrm{ReLU}(W_2\ \mathrm{ReLU}(W_1\mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2) + b_3$$

- $W_1 \in \mathrm{R}^{4\times2}$ → 4 neurons in layer 1
- $W_2 \in \mathrm{R}^{3\times4}$ → 3 neurons in layer 2
- $W_3 \in \mathrm{R}^{1\times3}$ → 1 output neuron

4. **Convolutional Filter Application.**

   A two-dimensional, 3 × 3 convolutional filter, that is,

1	0	1
0	1	0
0	0	1

   is applied to the following two-dimensional 5 × 5 input feature map

0	1	1	1	0
0	1	0	0.5	1
0	0	1	0	1
0	0	0.5	1	1
0	0	0	0	0

   What is the shape of the output feature map? Represent the output feature map.

Shape of output feature map: (5-3+1)*(5-3+1)=3*3

(1,1) of output feature map= (3*3 filter)*(Input 1) :

101 011
010 * 010
001 001

1*0 + 0*1 + 1*1 + 0*0 + 1*1 + 0*0 + 0*0 + 0*0 + 1*1 = 0+0+1+0+1+0+0+0+1 = 3

(1,2) of output feature map:
101 111
010 * 1 0 0.5
001 010

1*1 + 0*1 + 1*1 + 0*1 + 1*0 + 0*0.5 + 0*0 + 0*1 + 1*0 = 1+0+1+0+0+0+0+0+0 = 2

(1,3) of output feature map:

```
101       110
010   *   0 0.5 1
001       0 1
```

1*1 + 0*1 + 1*0 + 0*0.5 + 1*1 + 0*1 + 0*0 + 0*0 + 1*1 = 1+0+0+0+1+0+0+0+1 = 3

(2,1) of output feature map:

```
101       010
010   *   001
001       0 0.5 1
```

1*0 + 0*1 + 1*0 + 0*0 + 1*0 + 0*1 + 0*0 + 0*0.5 + 1*1 = 0+0+0+0+0+0+0+0+1 = 1

(2,2) of output feature map:

```
101       1 0 0.5
010   *   010
001       0.5 11
```

1*1 + 0*0 + 1*0.5 + 0*0 + 1*1 + 0*0 + 0*0.5 + 0*1 + 1*1 = 1+0+0.5+0+1+0+0+0+1 = 3.5

(2,3) of output feature map:

```
101       0 0.5 1
010   *   101
001       111
```

1*0 + 0*0.5 + 1*1 + 0*1 + 1*0 + 0*1 + 0*1 + 0*1 + 1*1 = 0+0+1+0+0+0+0+0+1 = 2

(3,1) of output feature map:

```
101       001
010   *   0 0.5 1
001       000
```

1*0 + 0*0 + 1*1 + 0*0 + 1*0.5 + 0*1 + 0*0 + 0*0 + 1*0 = 0+0+1+0+0.5+0+0+0+0 = 1.5

(3,2) of output feature map:

```
101       010
010   *   0.5 1 1
001       000
```

1*0 + 0*1 + 1*0 + 0*0.5 + 1*1 + 0*1 + 0*0 + 0*0 + 1*0 = 0+0+0+0+1+0+0+0+0 = 1

(3,3) of output feature map:

```
101       101
010   *   111
001       000
```

1*1 + 0*0 + 1*1 + 0*1 + 1*1 + 0*1 + 0*0 + 0*0 + 1*0 = 1+0+1+0+1+0+0+0+0 = 3

Final output feature map:

3	2	3
1	3.5	2
1.5	1	3