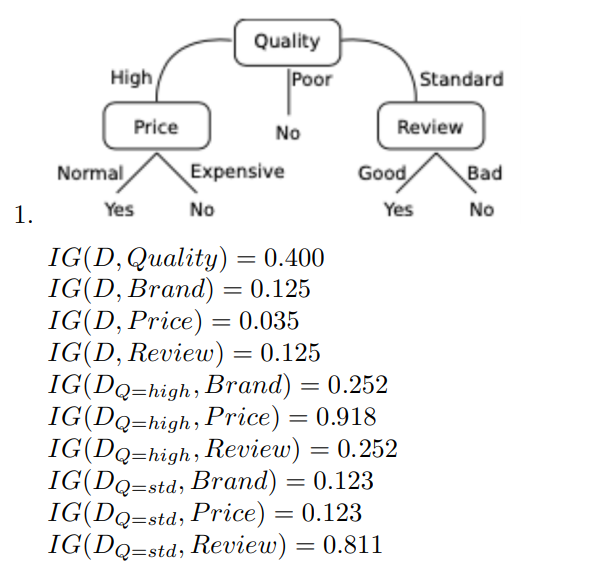
1. a) This is part of the exercises



1 (b)

**According to Piazza:**

* Use distance (Label same = 0, different = 1)
* Use euclidean z with “ordered” categories (like low, medium, high)

E.g. Low = 0, neutral = 0.5, high = 1

E.g. Poor = 0, standard = 0.5, High = 1

Then distance function can be:

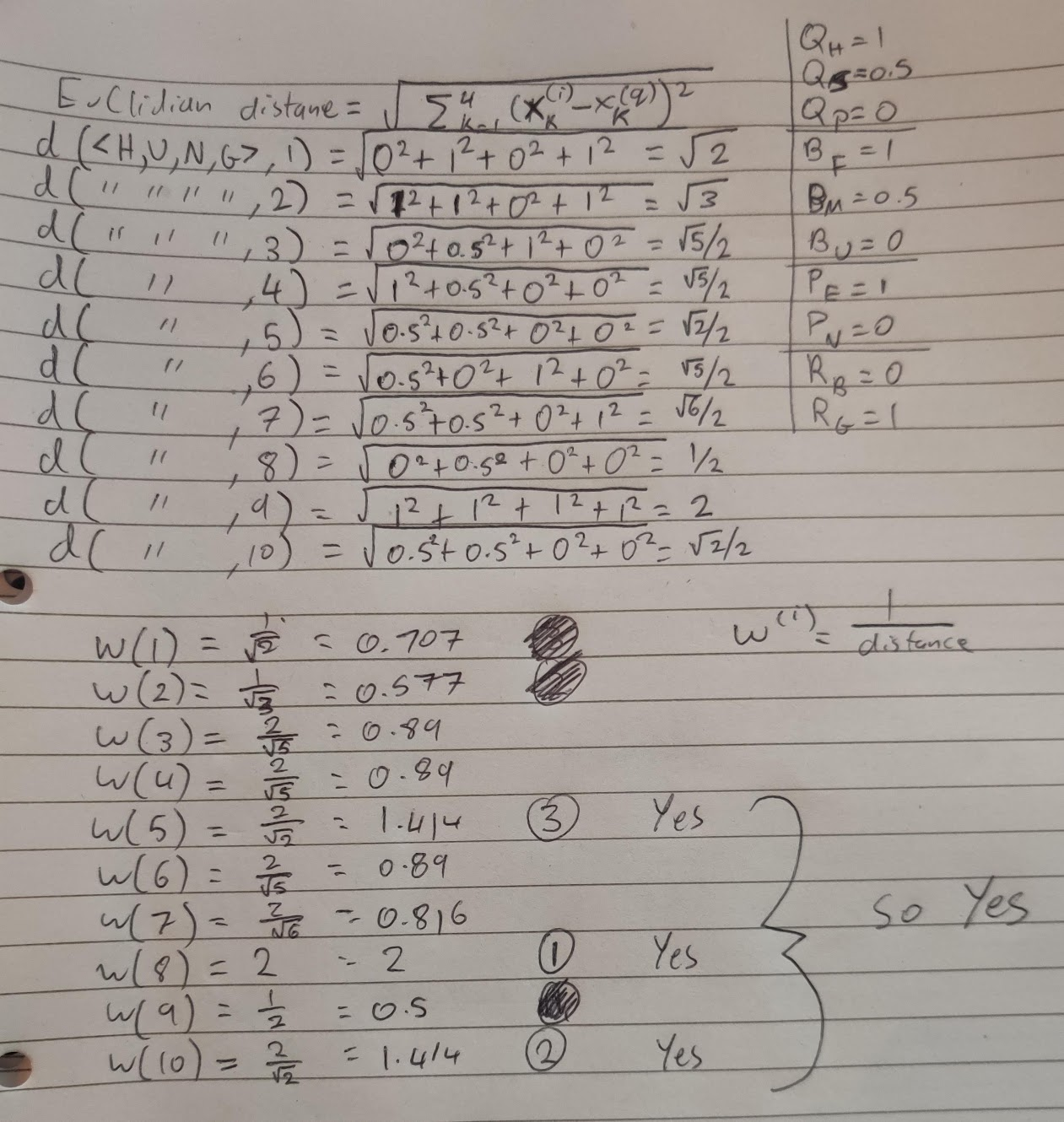
* Euclidean dist (sqrt x^2 + y^2 + z^2…)
* Sum of the differences (Hamming)

Weight function can be:

* e^-d
* 1 / d

Final answer would be YES with either calculation.

Another attempt at 1b:



c) Acc is 0.892

Recalls: 1, 0, 0.97 (Avg 0.657)

Precision: 0.5, 0, 0.98 (Avg 0.493)z

**Piazza has spoken: **

F1: 0.667, 0 , 0.975 (Avg 0.547)

F1 is most suitable, as it takes into account both precision and recall for each class.

Accuracy is biased towards the majority class, therefore not suitable for imbalanced data.

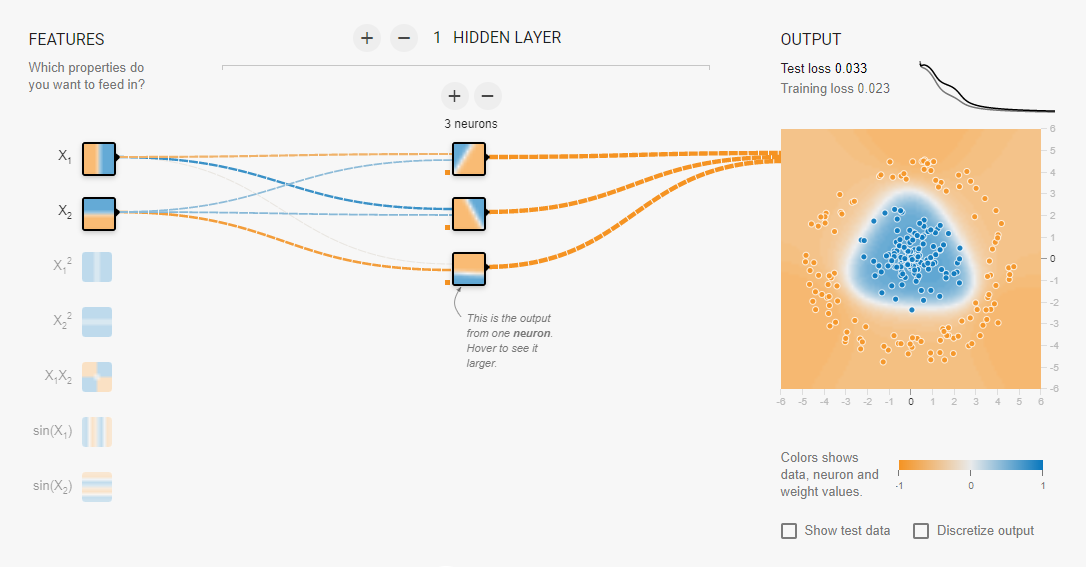
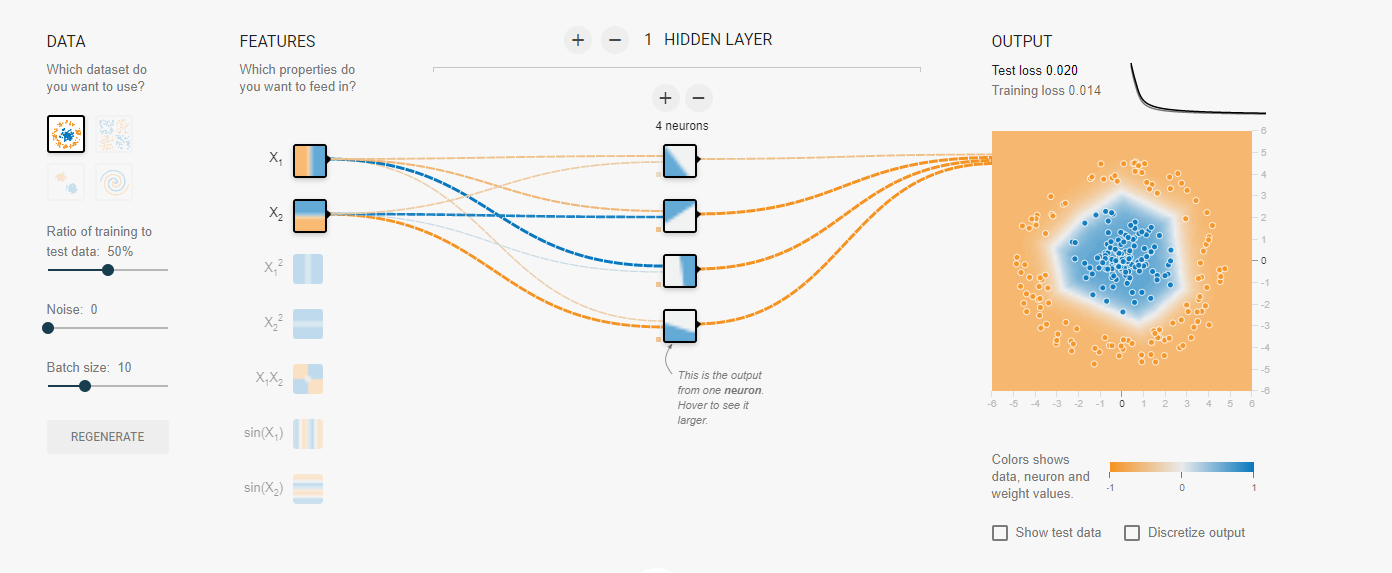
To mitigate, can oversample minority classes to balance out the distribution.

2)=

**Note: According to Piazza, you can’t change the inputs, ignore X^2 and Y^2**

**Standard solution:**  
Any reasonable answer should work, maybe just not too big.  
1 Hidden layer with 3+ neurons (Like making a triangle around the middle)

Decision boundary will be some shape, depending on your neurons (more neurons = closer to circle)

Sigmoid Activation after hidden layer?   


￼

b) Do we have to calculate it for both combined or just single sample?

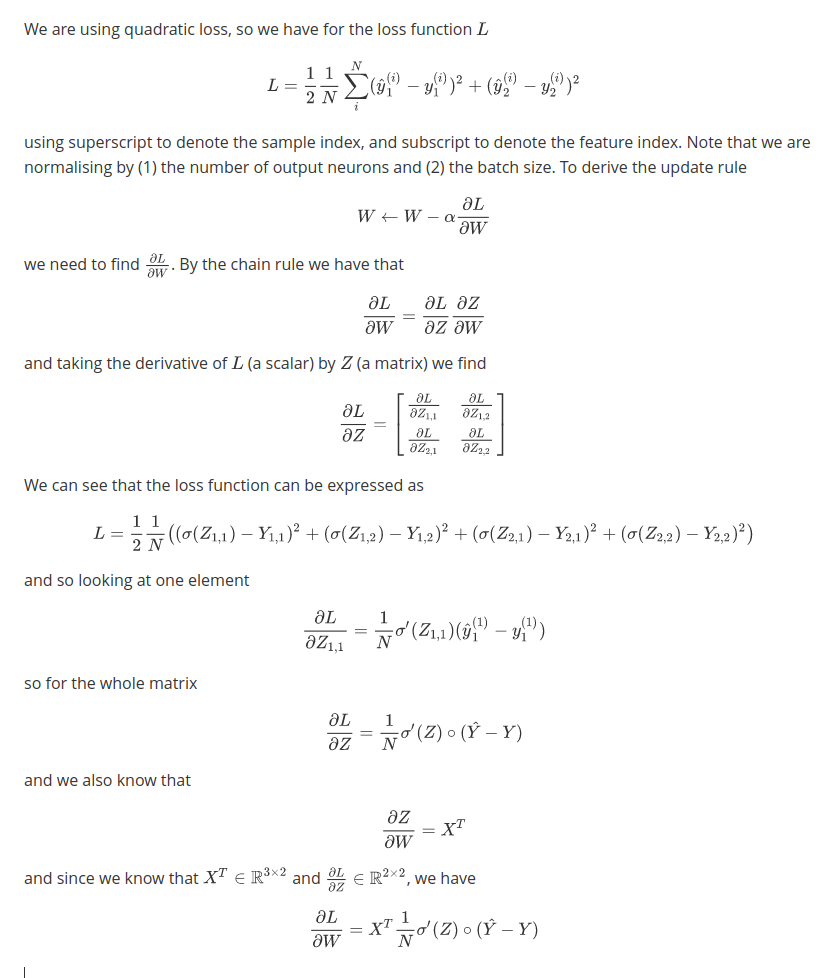
Derivative of loss function = 2 \* (Ŷ\_1 - Y\_1) + 2 \* (Ŷ\_2 - Y\_2)

Derivative of loss function = [(Ŷ\_1 - Y\_1), (Ŷ\_2 - Y\_2)] (should be a vector I think)

Derivative of Sigmoid = Previous derivative \* [(Ŷ\_1) \* (1 - Ŷ\_1), (Ŷ\_2) \* (1 - Ŷ\_2)]

Derivative of Weights = Previous derivative \* [x\_1\_1, x\_1\_2, x\_1\_3]

Another solution:



c)

200 samples.

Shuffle the dataset

Use k-fold cross validation, take k = 10 or 5?

For each split, take the remaining k-1 and perform the split again.

Use the inner cross-validation to split into train and validation.

Perform training on train set, then hyperparameter tune on validation.

Pick best parameters on each inner split.

Compare outer splits on test set and pick best one?

Then before production, train on the whole data

SS

3) Refer to GA tutorial differences:

1. Genotype: Binary string of length 12, 4-bits per angular position (ceil(log\_2(11)) = 4) # this will give some invalid encodings that need to be handled by returning the lowest fitness score

Phenotype: A list of 3 angular positions, can be “None” in the case of an invalid encoding

Function: Split genotype (the binary string) on length ‘4’, then convert from binary to decimal, then use as index into “angular\_positions” list to obtain the angular position. If out of bounds index then return “None” (as invalid encoding)

1. Are we meant to state the specific operators?

Selection: Picking N fittest, or from cumulative distribution, or best from random pairs.

Select parents (using biased roulette wheel or draw 2)

Meant to select from the fittest to pass on genes.

Crossover: Slice bits at random position and combine

Mutation: “Flip random bits” algorithm with hyperparameter ‘m’ = 1/12 probability (12 since this is the genotype size). Per bit in the genotype, draw randomly ‘v’ from U(0, 1), then if “v < m” flip the bit

Meant to allow children to explore the search space for better performing values than parents.

1. Fitness: Negative Distance to target, as maximising this metric achieves our desired goal (0 distance = touching target) *Potentially adding special cases for phenotypes that are invalid and forcing them to –inf so that they are discarded in next gen*
2. 1. Generate N random population

2. Evaluate fitness

3. Use selection operator to select U best

4. (Optional) Keep X best performing (Elitism)  
5. Pick randomly from selection pool, use crossover to generate N new

6. Perform mutation on each child

7. Go back to step 2 until convergence

Define hyperparameters:

* + Number of epochs till convergence
  + Probability to flip during the mutation phase
  + U – how many parents will you select during the selection operator