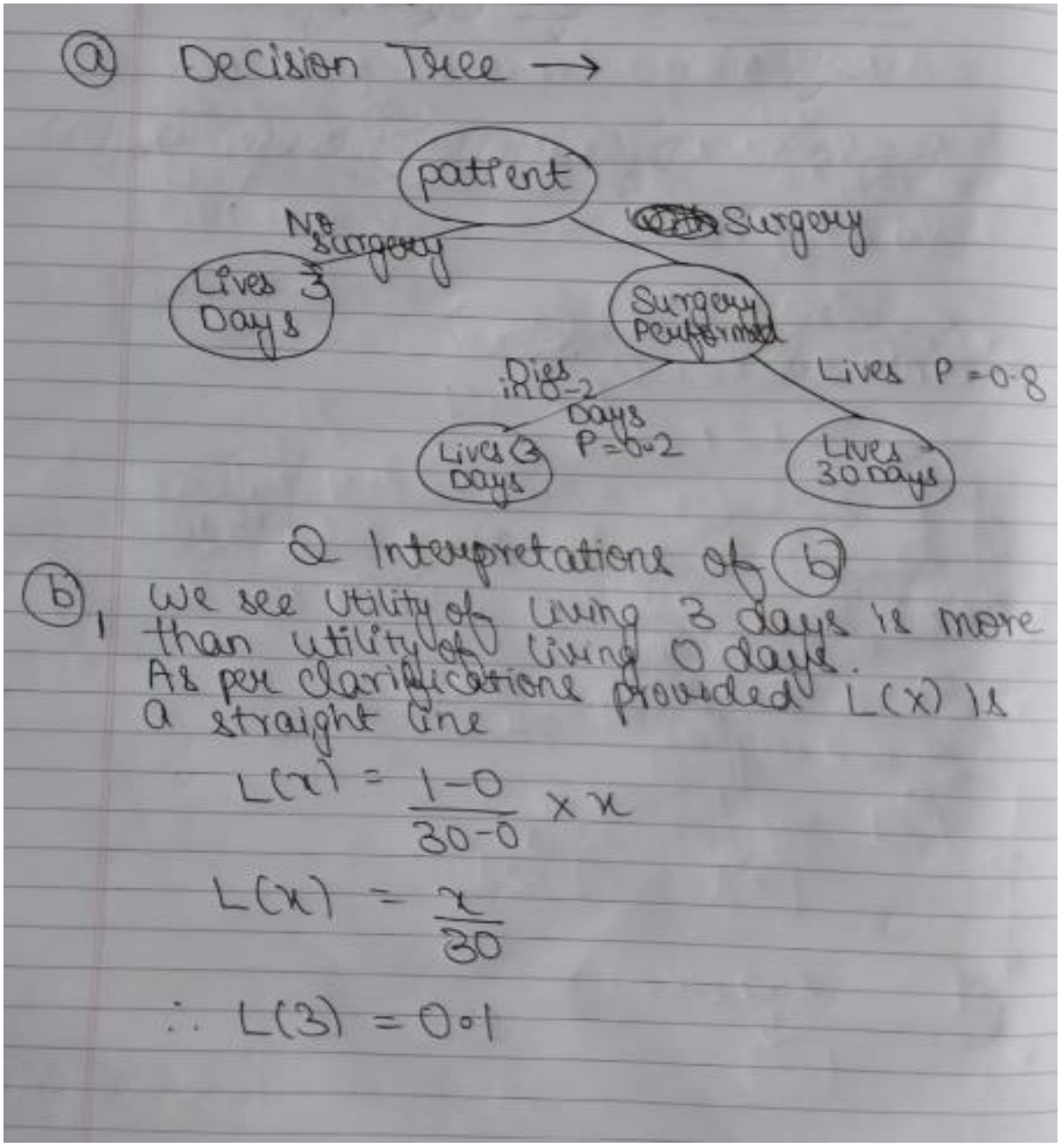
Assignment 2

Report

Name: Mohammad Aflah Khan

Roll No.: 2020082

Section A



(B2) P(Surviving Surgery) = 0.8

∴ If $L(3) < 0.8$ we take a surgery as it gives higher survival chances

If $L(3) > 0.8$ the patient won't take the surgery as without it the patient has higher survival chances

Hence patient will take surgery even when $L(3)$ goes down to 0

Truly both interpretations lead to same tending to 0 (lower bound as $L(0) < L(3)$ hence it can't be 0

(c) P(Test positive | Survival) = 0.95

P(Test positive | Not Survive) = 0.05

P(Survive | Test Positive) = P(Test Positive | Survival) * P(Survival) / P(Test Positive)

Survive = Successful Surgery

$$= \frac{0.95 \times 0.8}{0.95 \times 0.8 + 0.05 \times 0.2}$$

$$= 0.987$$

We get P(Test Positive) via formula of total probability where

P(Test Positive) = P(Test Positive | Survival) * P(Survival) + P(Test Positive | Not Survive) * P(Not Survive)

(d) If test results are positive we see
P(Successful Surgery | Test +ve) = 0.987
meaning there is a 98.7% chance of survival (also it is more than L(3))

(e)

$$P(\text{Survive} \mid -\text{ve Test}) = \frac{P(+\text{ve Test} \mid \text{Survive}) \cdot P(\text{Survive})}{P(-\text{ve Test})}$$

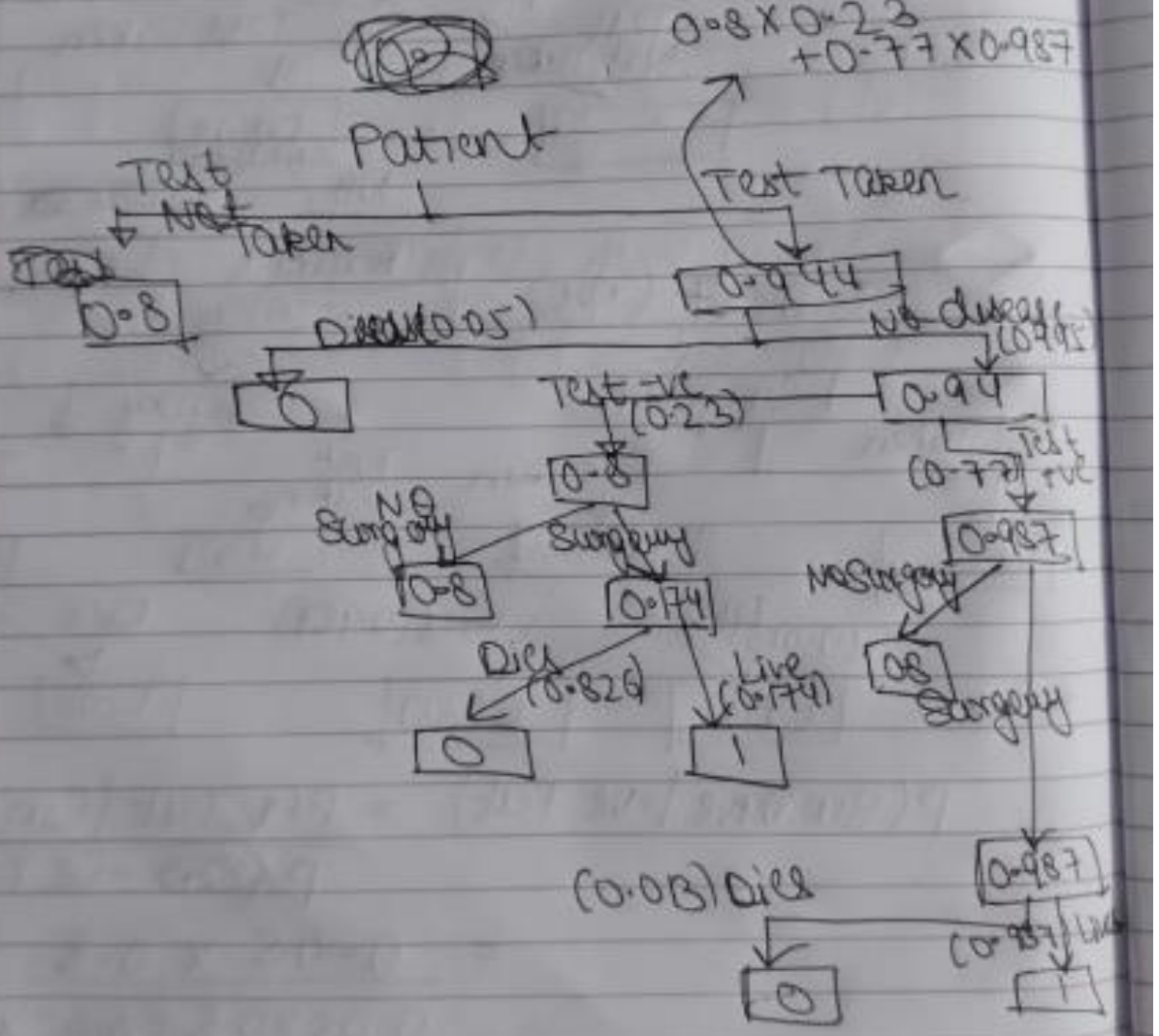
$$= \frac{0.05 \times 0.8}{0.05 \times 0.8 + 0.95 \times 0.2}$$

$$= 0.174$$

(f) P(contracting Disease) = 0.005

So now we can compute some more probs.

Note: Each box has P(survival) at that junction with best case branch estimates

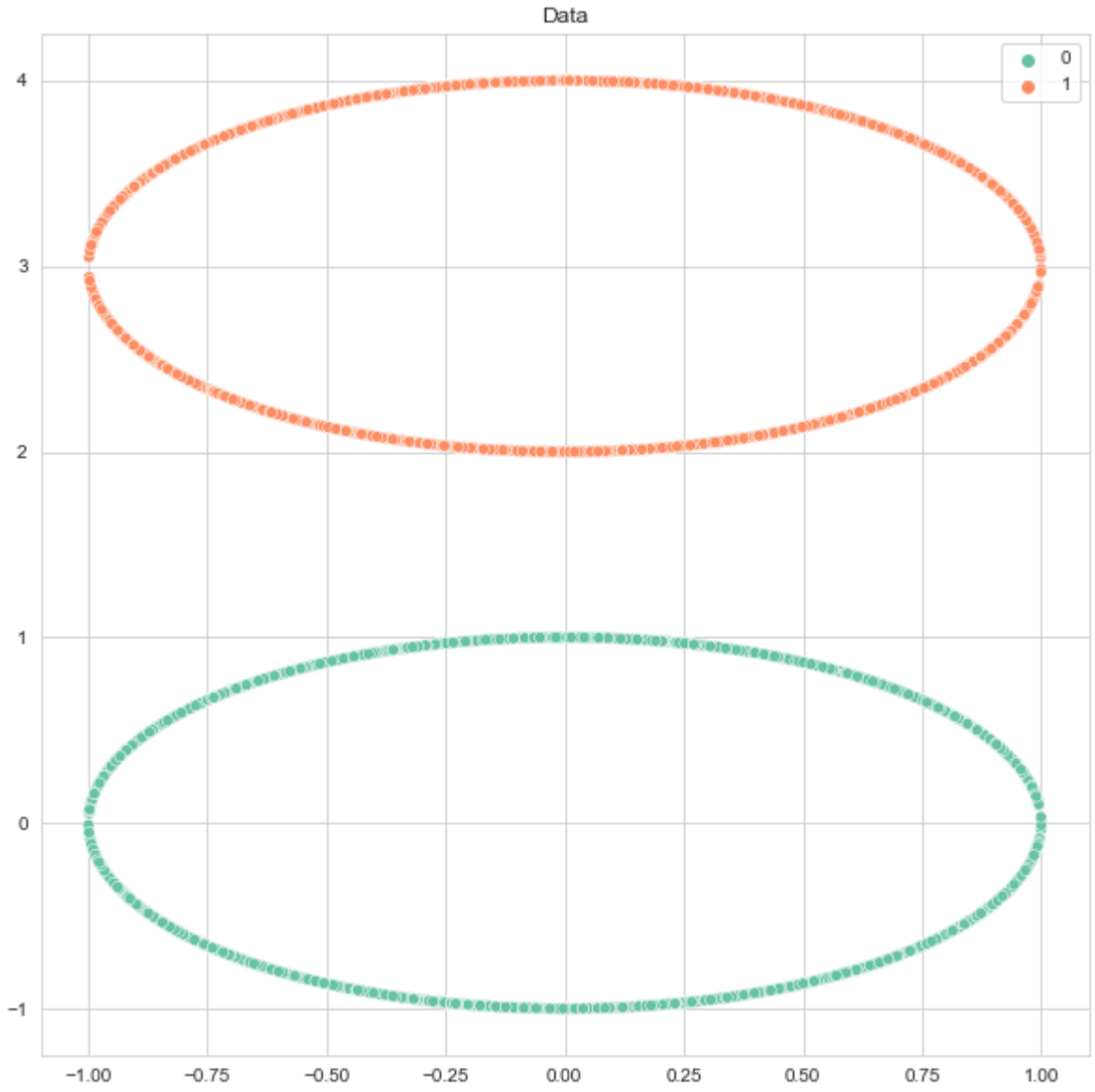


Section B

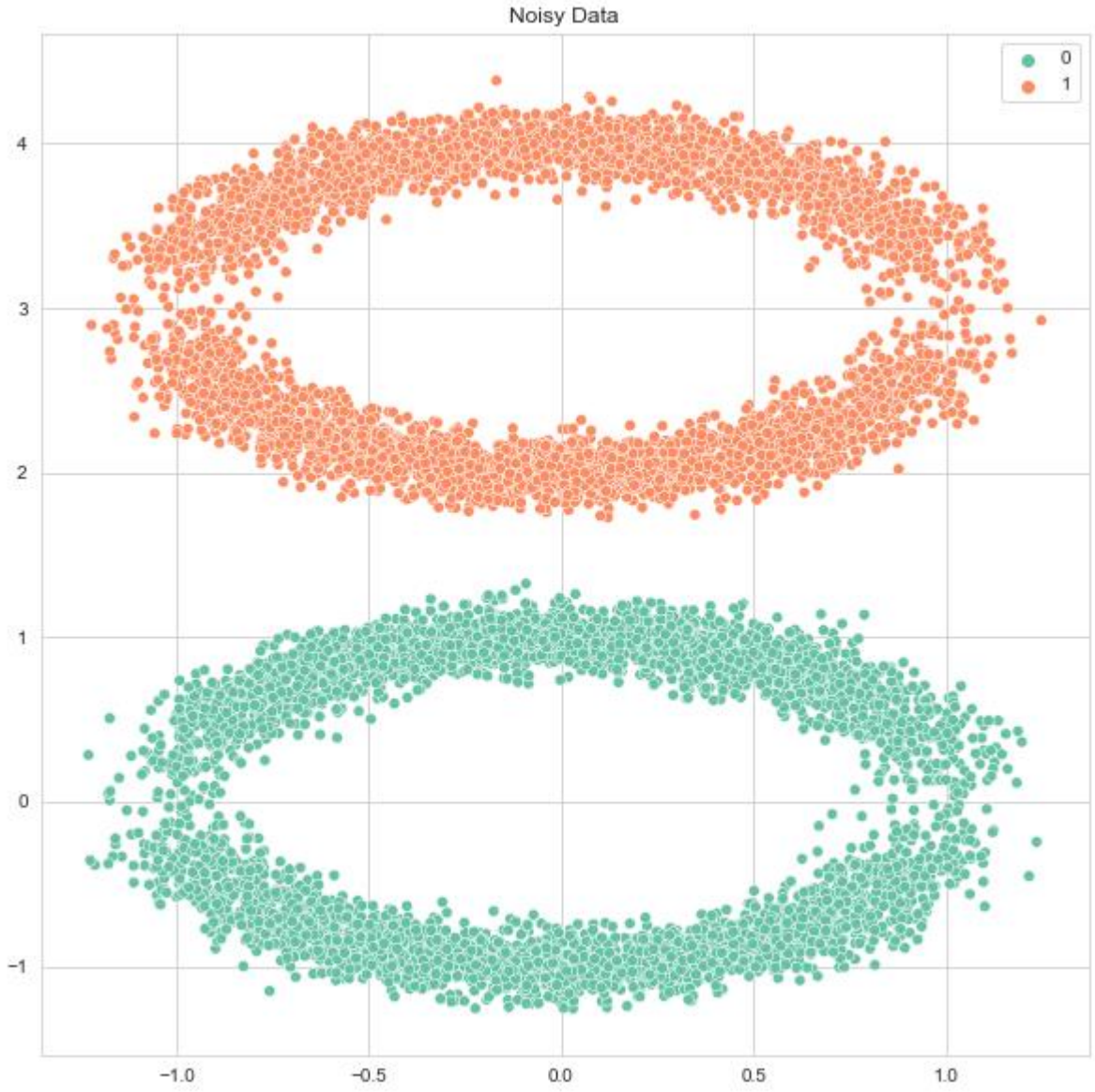
Ans 2)

Dataset Plots-

Without Noise:



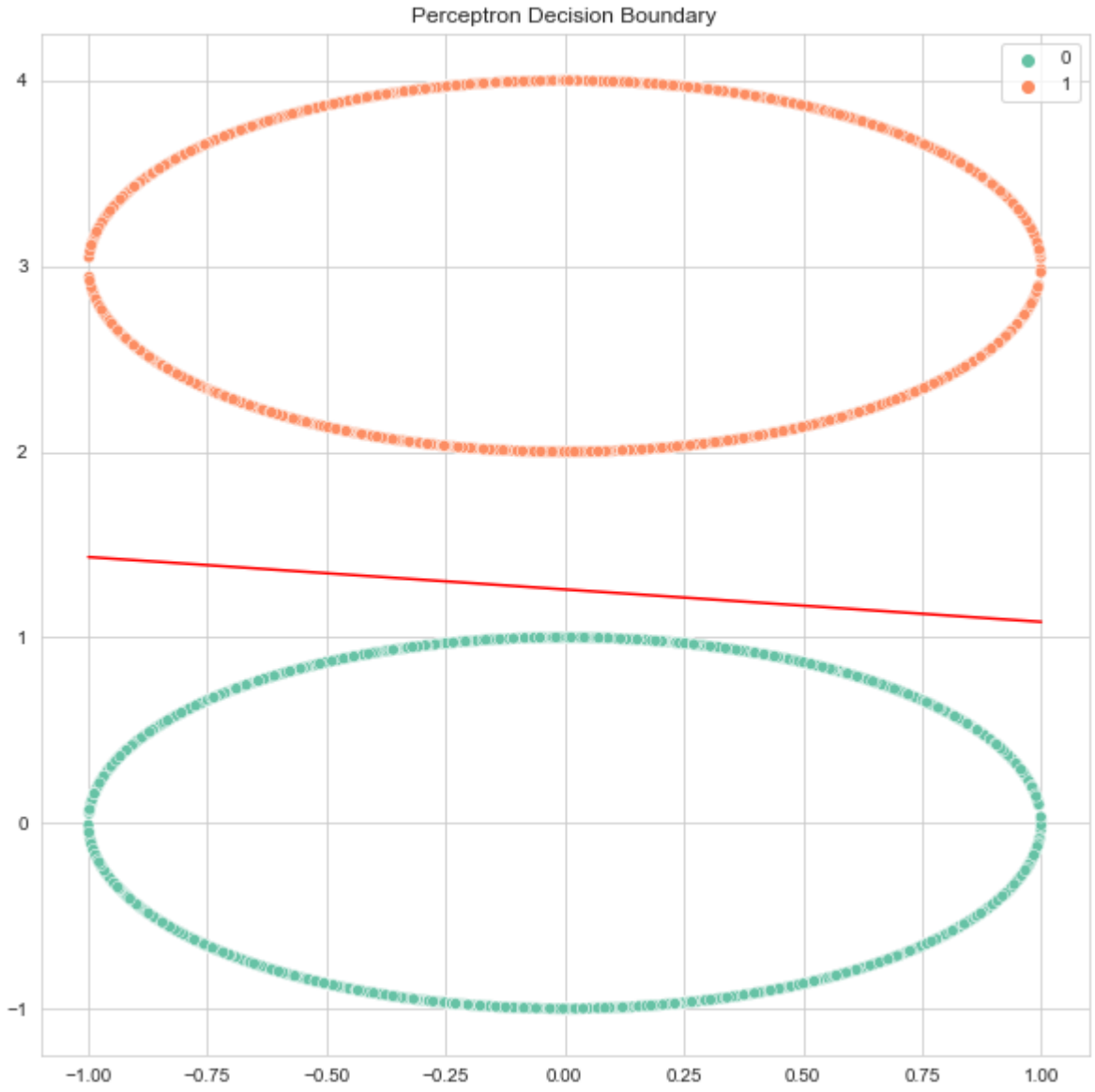
With Noise:



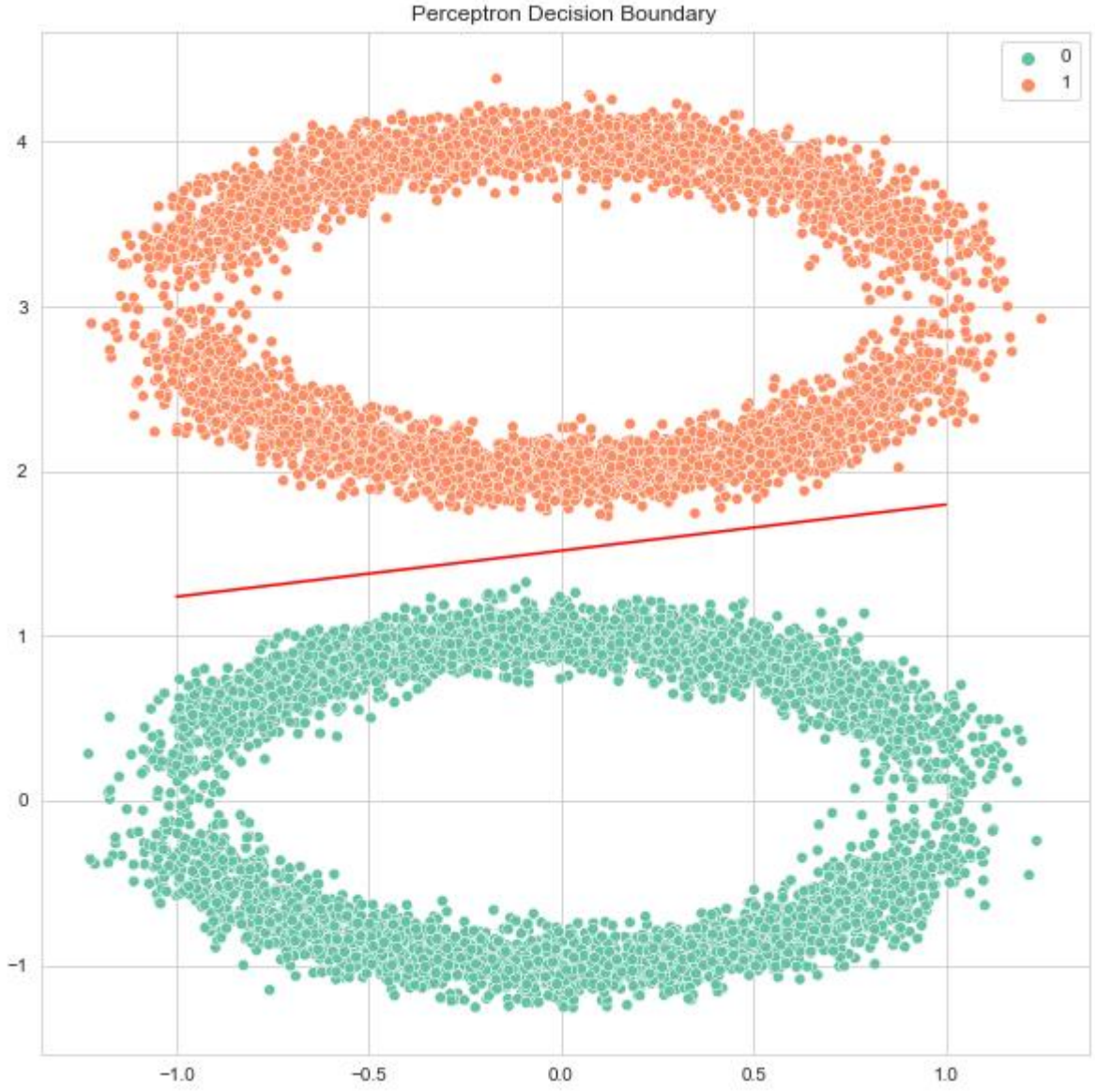
Ans 3)

In all subsequent plots the red line denotes the decision boundary.

PTA Training without Fixing Bias on Noise Free Data –



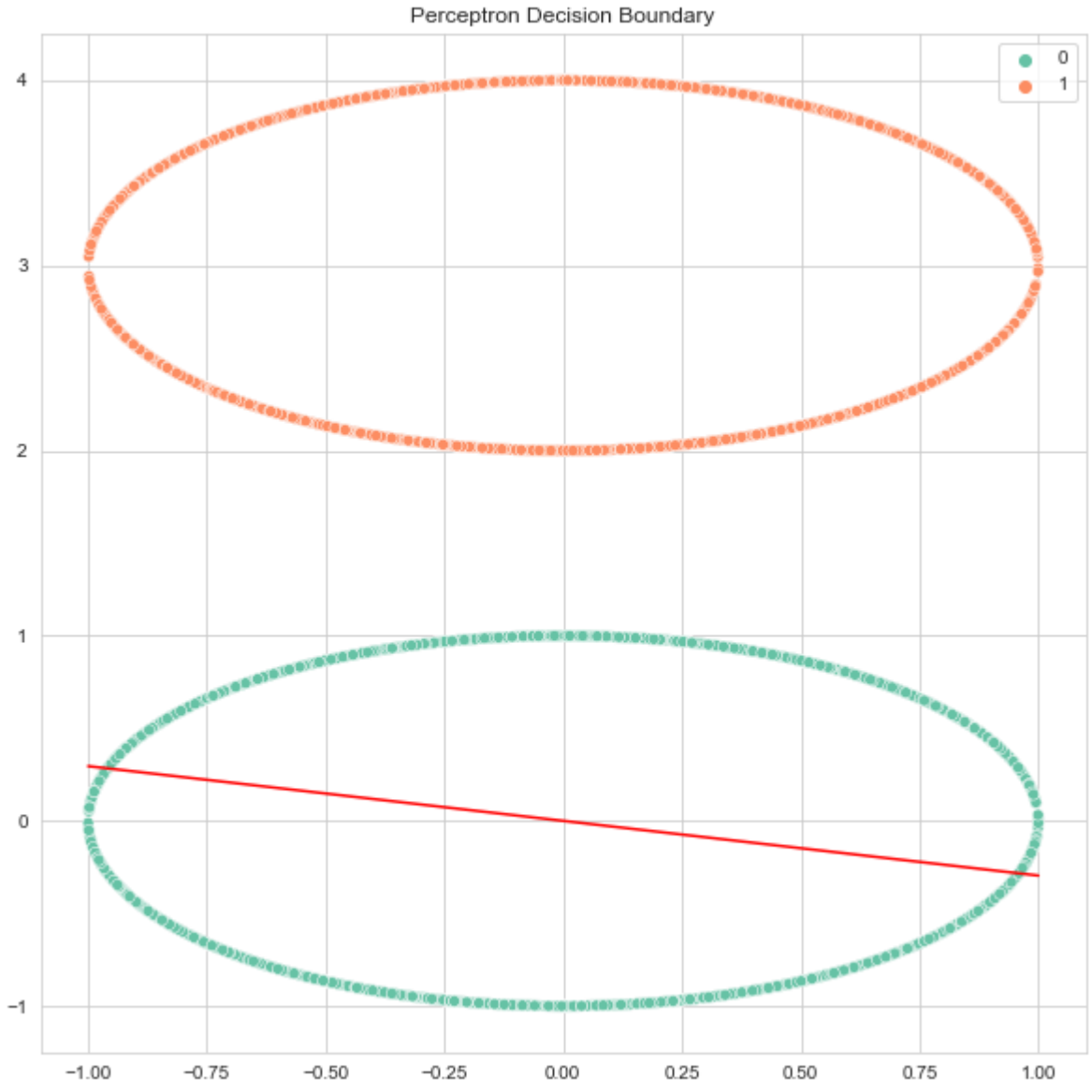
PTA Training with Fixed Bias on Noisy Data -



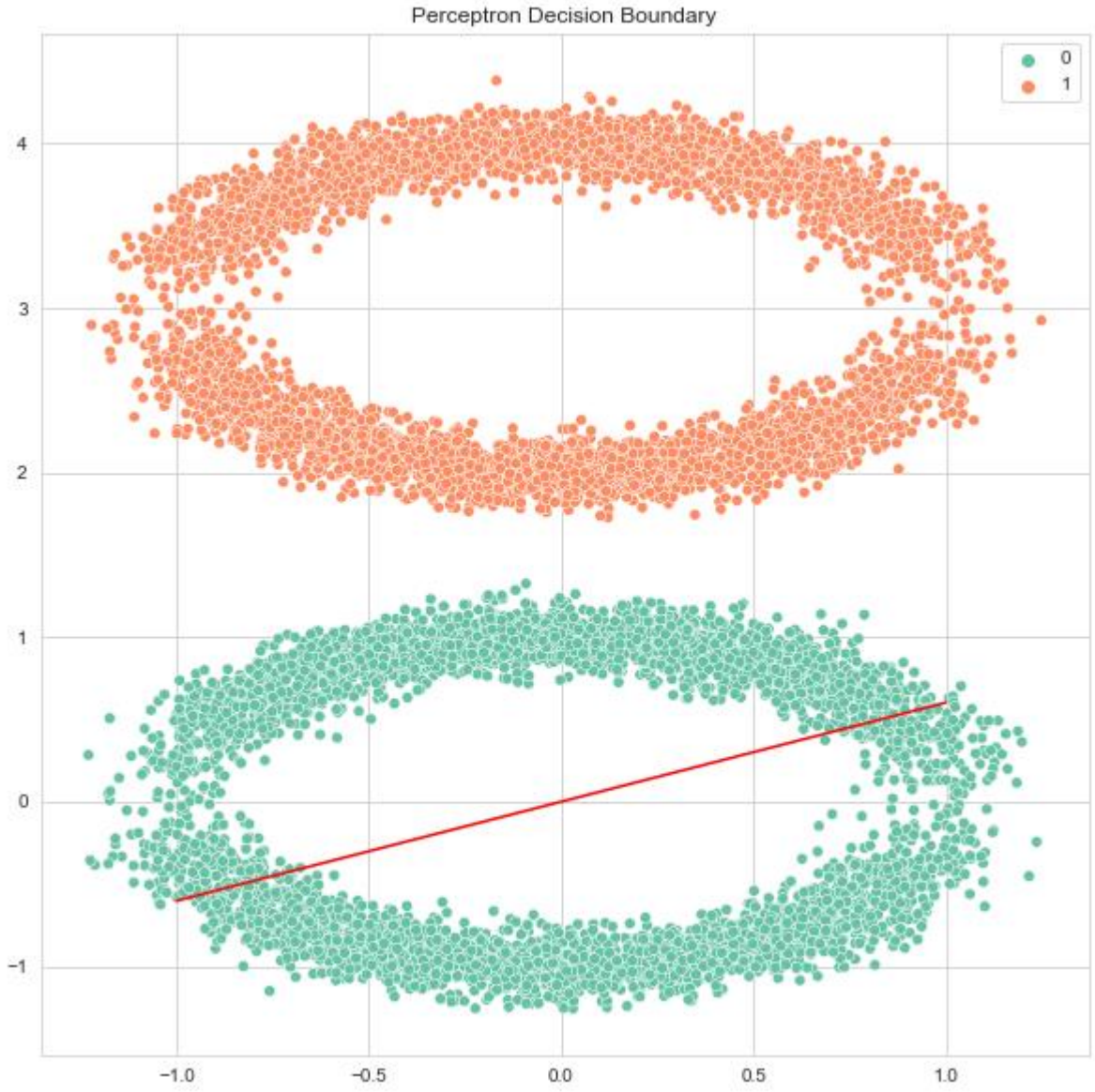
A decision boundary exists in both cases as we can see the 2 classes of points are linearly separable. In the case of noisy data, it is highly unlikely but possible that one the points might outlie by a huge margin as they are sampled from a gaussian distribution and the noisy case stops being linearly separable. However as mentioned before it is highly unlikely and in almost all sampling experiments the noisy data is also separable while the noise free data is always linearly separable.

Ans 4)

PTA on Data Without Noise with Fixed Bias = 0:



PTA on Data with Noise with Fixed Bias = 0:



On Comparing with 2.3 we notice that our model hasn't learnt a good decision boundary. This is because we've constrained it heavily by fixing our bias which forces it to pass via the origin. For the points belonging to class 0 i.e. which lie on the circle with center at origin and radius 1 this line would infact be the diameter as it passes through it's centre and hence cuts it in half. Since the line is the diameter for even one circle it can't even let points of the same class lie on the same side of the line leave alone becoming a decision boundary between 2 classes. This just goes to show how powerful the bias term is. In 2D where we only have 2 parameters to manipulate restricting the bias can also be thought of restricting one degree of freedom as now we can only change the slope of the line but not the intercept.

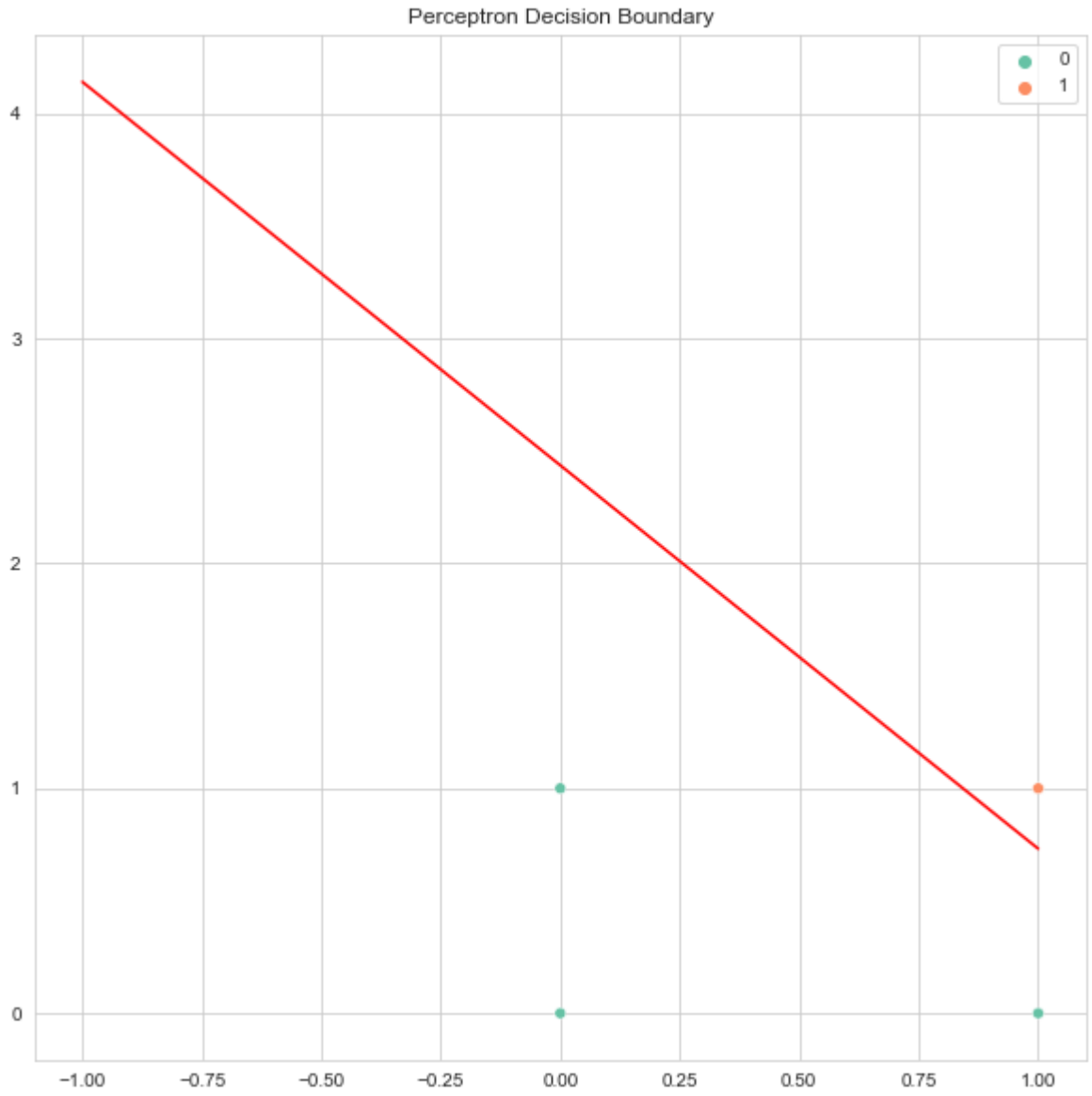
Note: Interestingly there is a non 0 but infinitesimally small chance that the noisy case would infact become solvable even with 0 bias using the same reasoning which we used in last part to say it wouldn't be solvable had one point been outlying by a huge margin and end up inside the other circle. Here it would be solvable if a lot of points outlie by a huge margin due to the noise to make

them linearly separable but again this is just a theoretical consideration and effectively has a zero chance of happening in practice.

Note: Since the algorithm won't converge it keeps running hence I've set a hard limit to stop training at 1000 epochs even if convergence does not happen.

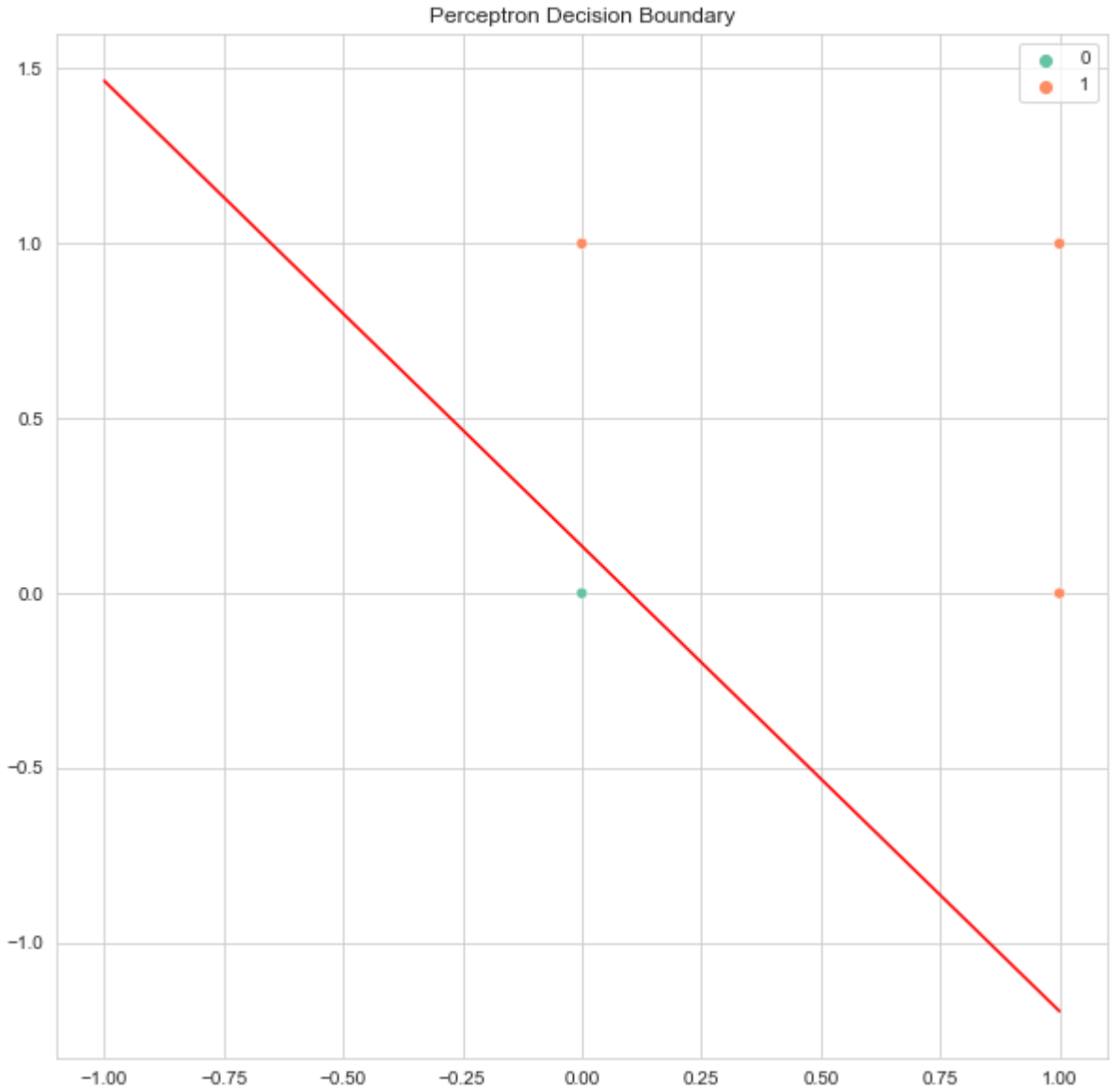
Ans 5)

Solving AND with Learnable Bias –



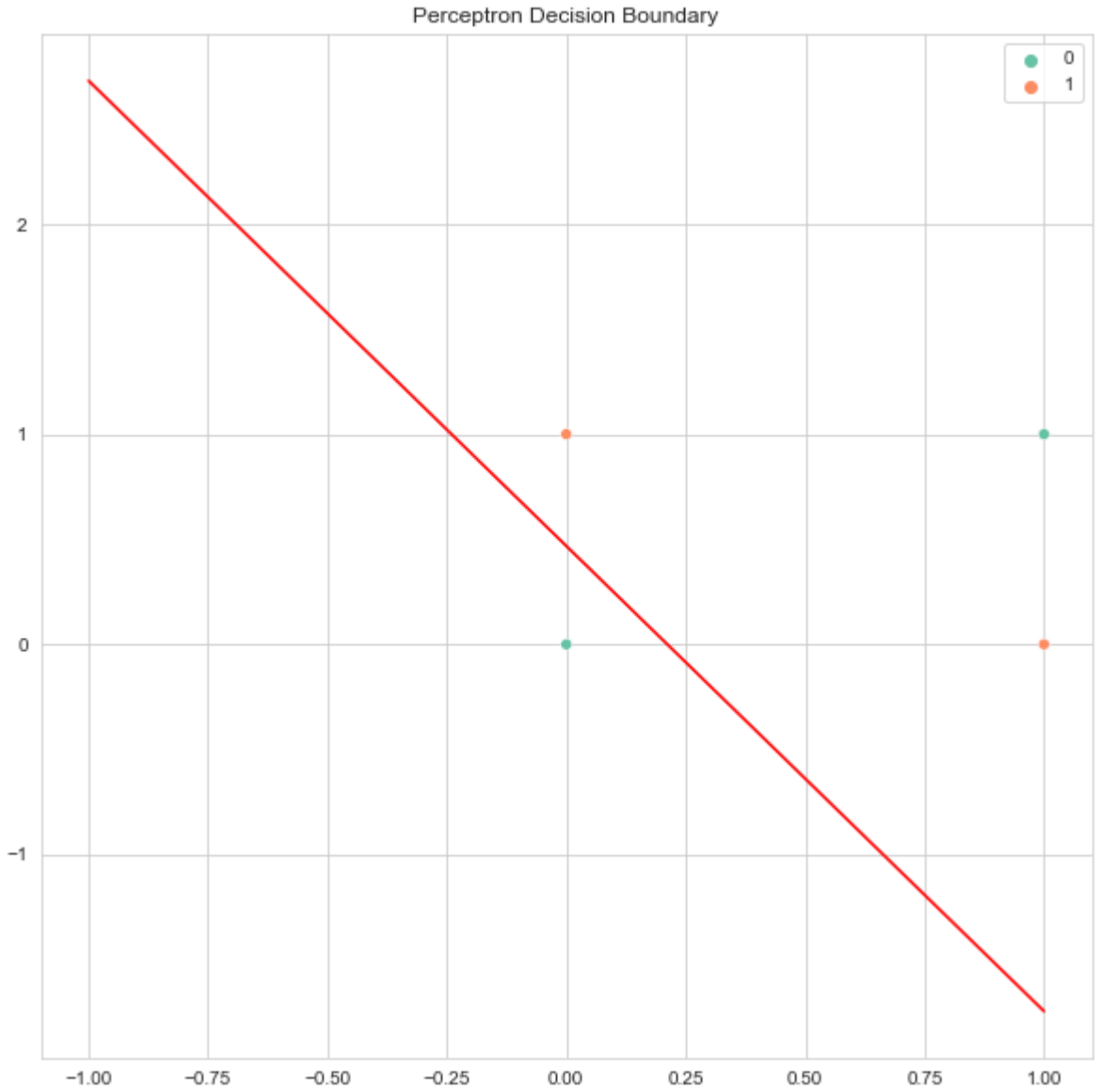
We can see we get a linear boundary which distinguishes between the 2 classes. Hence AND is solvable with a perceptron where bias can be trained.

Solving OR with Learnable Bias –



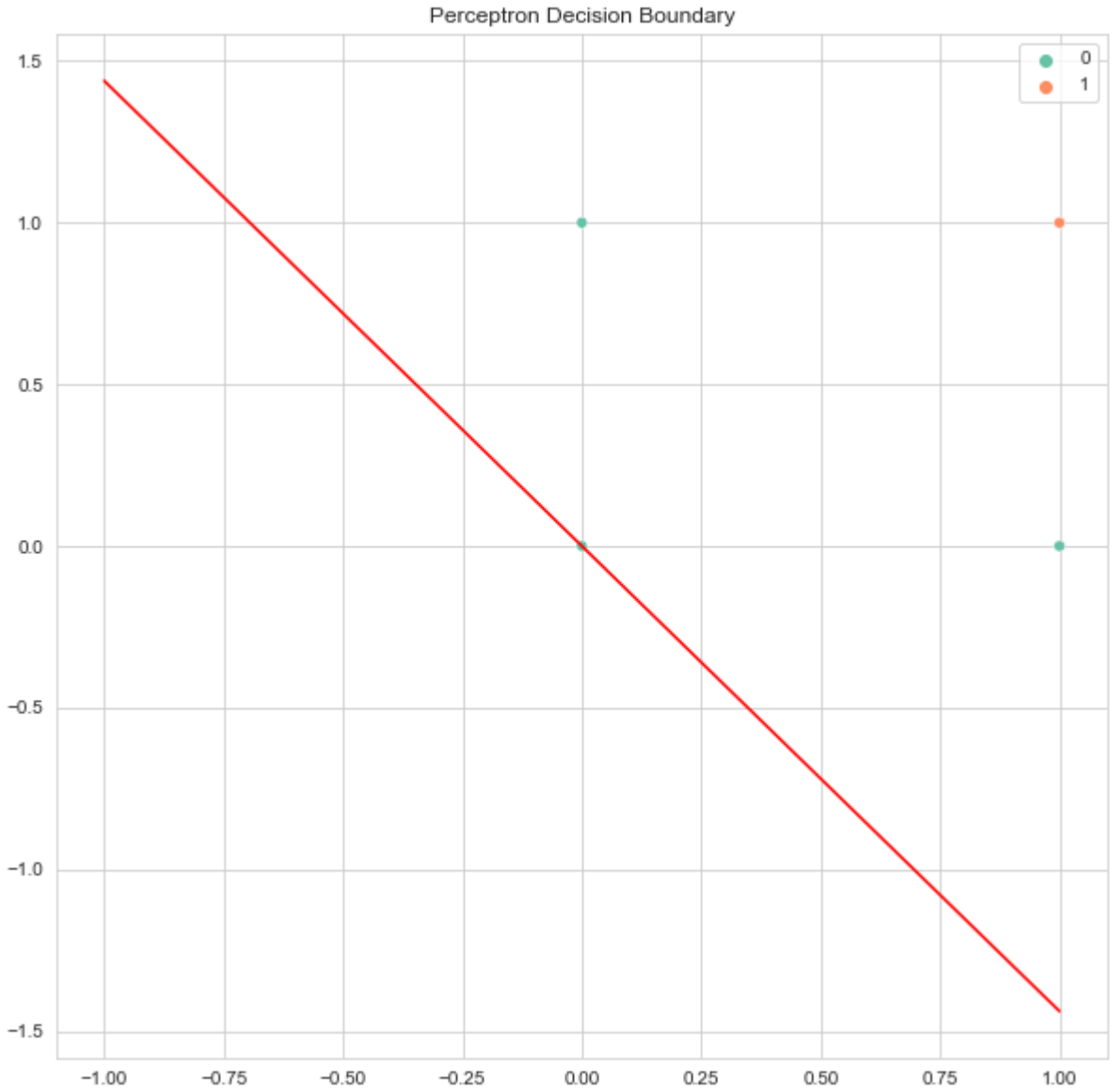
We can see we get a linear boundary which distinguishes between the 2 classes. Hence OR is solvable with a perceptron where bias can be trained.

Solving XOR with Learnable Bias –



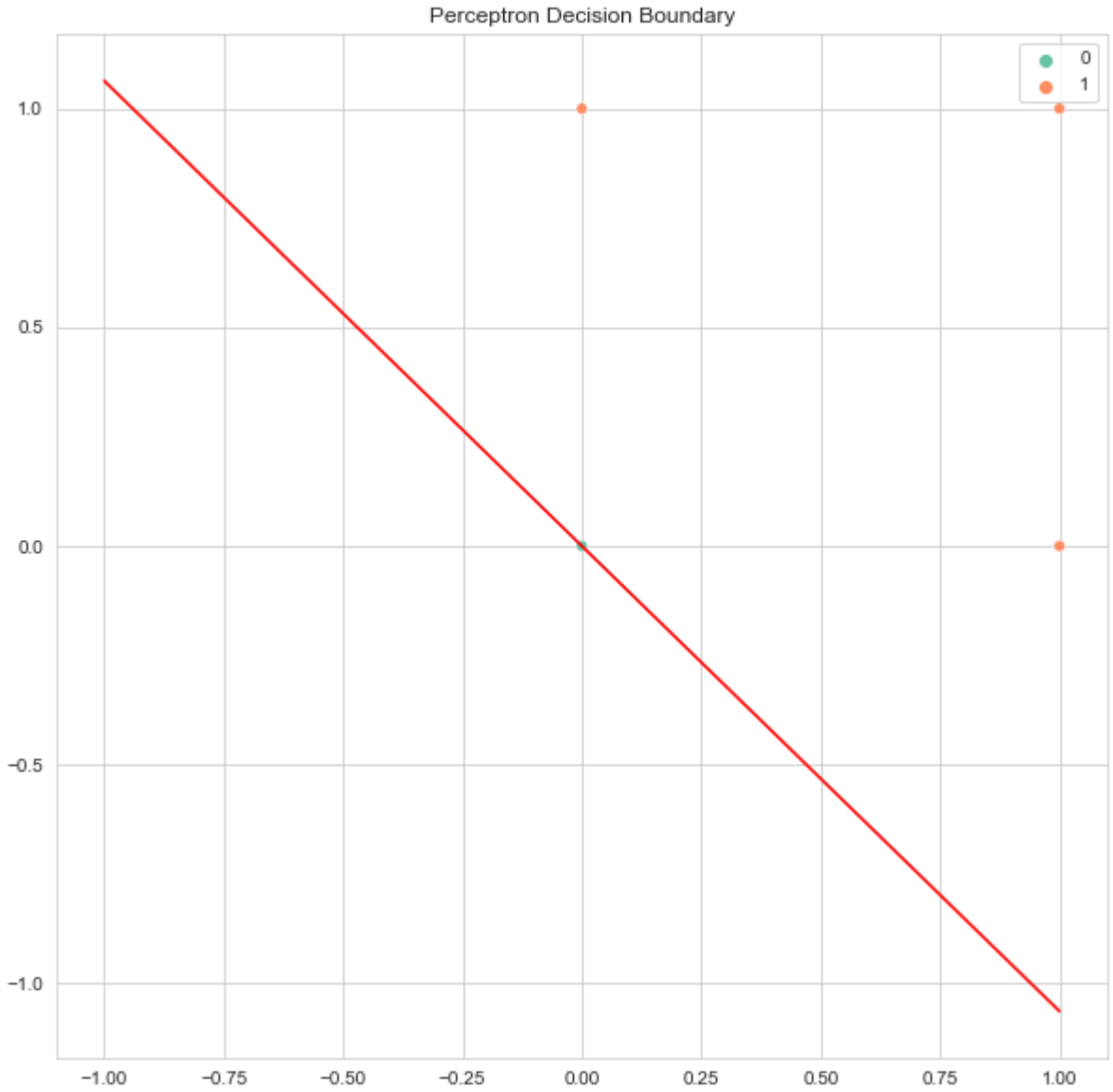
We can see we do not get a linear boundary which distinguishes between the 2 classes. Hence XOR is unsolvable with a perceptron where bias can be trained.

Solving AND with Fixed Bias = 0 –



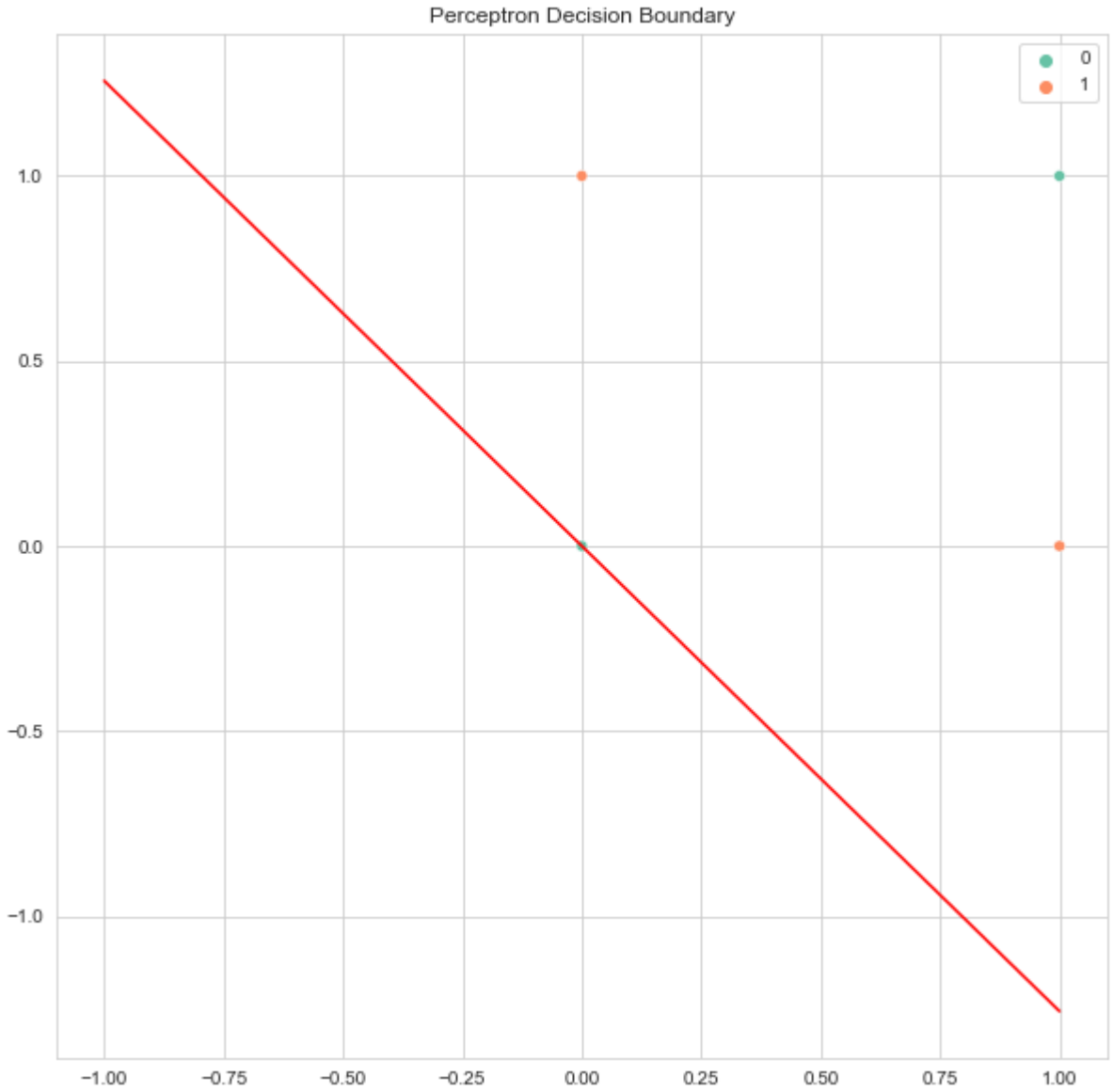
We can see we do not get a linear boundary which distinguishes between the 2 classes. Hence AND is unsolvable with a perceptron where bias cannot be trained.

Solving OR with Fixed Bias = 0 –



We can see we get a linear boundary which can solve the problem we just utilize an assumption that any point lying on the line or lying such that f(x) <= 0 is class 0 while the other is class 1.

Solving XOR with Fixed Bias = 0 –



We can see we do not get a linear boundary which distinguishes between the 2 classes. Hence XOR is unsolvable with a perceptron where bias cannot be trained.

6)

To predict which class It belongs to we can compute the value of f(x) where f is the function which applies the transformation on an input x to yield the output of the perceptron.

Now, if f(x) <= 0 we say it belongs to one class say 0 and if it's > 0 we say it belongs to the other class say 1. Here we need to decide which class label to give to points lying on the decision boundary so for that we can arbitrarily choose one convention and stick with it. In my case I've assumed points lying on the boundary that is those where f(x) = 0 to be class 0.

Here f(x) is an equation of a d-1 dimensional plane where d is the input space. So for 3d points it's a 2d plane etc.

Section C

Pre-processing Applied –

1. Dropped Index
2. Dropped Address Column as it was unique for most samples
3. Shuffled Data prior to splitting
4. Since there is no missing data, there was no need to handle it

General Observations and Methodology –

1. Due to heavy class imbalance the data is very skewed
2. Methods to add data to balance classes make the dataset much larger than what a normal machine can handle and also it's difficult to maintain the Independence and Identically Distributed Assumption while doing that.
3. Class balancing by cutting down samples would force us to lose over 95% of the data belonging to class 'white' hence it also doesn't make sense
4. Since the trees particularly the Random Forest Stumps and AdaBoost Trees take too long to train I've trained them once and saved them and for inference I load them however the code to train them can be uncommented and run again as well.

a)

Criteria	Depth	Accuracy – Validation Set	Accuracy – Test Set
Gini	4	98.5609307322202	98.58858755899934
Gini	8	98.62424429435093	98.65510108455904
Gini	10	98.66675809419321	98.70881475640279
Gini	15	98.77349973143164	98.79498520016914
Gini	20	98.65327253402818	98.6772722597456
Entropy	4	98.5609307322202	98.58858755899934
Entropy	8	98.59407321059189	98.62173003737101
Entropy	10	98.72801453697672	98.74401435412167
Entropy	15	98.80869932915053	98.8219563204992
Entropy	20	98.60710163312419	98.62447286316728

These values differ by small margins across different runs

The best Validation and Test Accuracy seems to be produced by the Decision Tree having depth 15 and using the Criteria of Entropy.

b)

For Validation Set - Accuracy for ensemble of 100 trees is 98.5609307322202 %

For Test Set - Accuracy for ensemble of 100 trees is 98.58858755899934 %

On Comparison with the best performing tree we can see that the performance is in the same neighbourhood and so is the variance. We do not see significant improvements due to 2 reasons:

1. The results are already pretty good without the random forest
2. The dataset is heavily class imbalanced meaning some 50% data subsets would only have 1 class making them practically single class predictors which don't give them much power

c)

AdaBoost Classifier was trained using Decision Tree Classifier as the base estimator with depth 15

Number of Estimators	Accuracy – Validation Set	Accuracy – Test Set
4	97.80048228020252	98.75041428097965
8	97.5264282693912	98.51315984960172
10	97.13763271276899	98.14859258751328
15	97.52688540702393	98.43476074559148
20	97.77968251791408	98.67132947052035

We can see AdaBoost works better than Random Forests even with much less number of estimators because it corrects itself as it is a meta-estimator by fixing places where it made error. The results stay fairly consistent as we increase the number of estimators as well. In the case of RF we hope different trees learn different things and correct each other's mistakes but in the case of AdaBoost that is explicitly a part of it's design giving it more power