MACHINE LEARNING (CSE343/ECE343)
ASSIGNMENT-1

Sourav Goyal
2020341

SECTION-A

Ans(1)

Assignment-2 -(ML)
CSE343 / ECE343

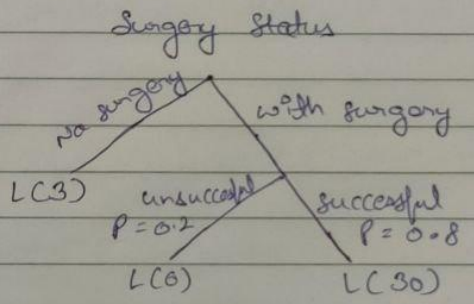
Sourav Goyal
2020341

Section A

Ans-① (a) Given:-
- survivor of the pandemic can die in 3days if not given surgery,
- survivor might live for 30 days post surgery.
- P(survivor doesn't survive the surgery) = 0.2.

∴ Let L(u) denotes the survivor's living function where u denotes the number of days survivor lives,

» So, decision tree will be,



(b) Given:- L(30) = 1 and L(0) = 0.

» Expected value of surgery:-

» E(with surgery) = 0.8 × L(30) + 0.2 × L(0)
= 0.8 × 1 + 0.2 × 0
= 0.8

∴ If L(3) < 0.8, then surgery should be performed.

» Now, we will use L(3) = 0.8 for next parts as given in this problem.

A-(C) Given :- P(Result of test is true) = 0.95
P(Result of test is false) = 0.05.

⇒ We have to find survivor's probability of having a successful surgery if the test is positive,

∴ P(successful surgery / test is +ve),

⇒ $P(C) = \frac{P(\text{test is +ve / successful surgery}) \times P(\text{succ. surg})}{P(\text{test is +ve})}$

⇒ $= \frac{P(\text{True positive Rate}) \times P(\text{succ. surg})}{P(\text{test is +ve})}$ [by bayes theorem]

⇒ $= \frac{0.95 \times 0.8}{0.95 \times 0.8 + P(\text{test +ve / unsuccessful}) \times P(\text{unsuccessful})}$

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

$= \frac{0.76}{0.76 + 0.01} = 0.987.$

∴ P(successful surgery / test is +ve) = 0.987. Ans

A-(d) From (b) part we know $L(3) = 0.8$.

- We have to find that whether we have to performed surgery if the test is positive or not.
- So, Expected value of surgery performed with test is positive, call it as variable S.
- $E[S] = 0.987 \times 1$.

$= 0.987$.

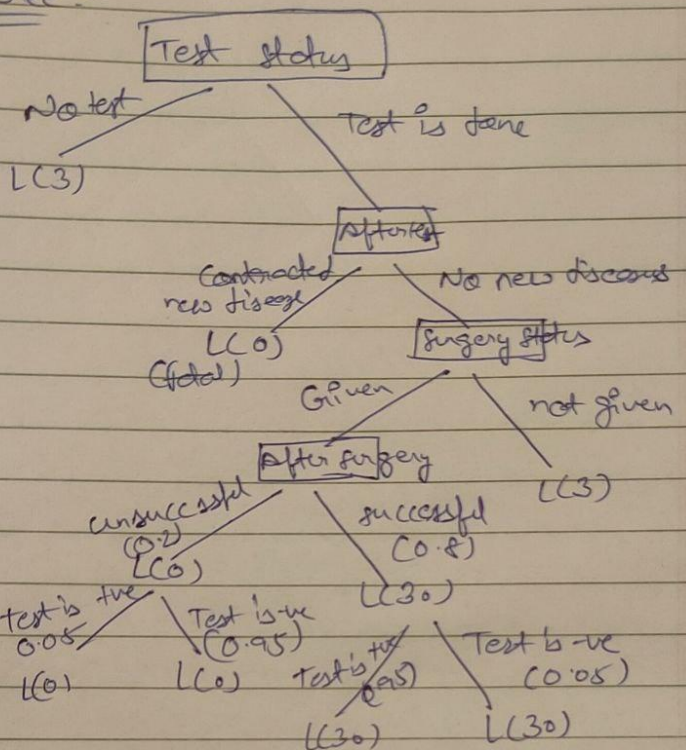
∴ Their is higher expected value of serving when we have test is positive which is 0.987 and for not take the surgery we have $L(3) = 0.8$.

- So, $0.987 > 0.8$, therefore surgery should be Performed.

Ans-(e) Let's take Probability p of patient can have new disease during the test,

Assumption :- If survivor wants a surgery then the test should be mandatory.

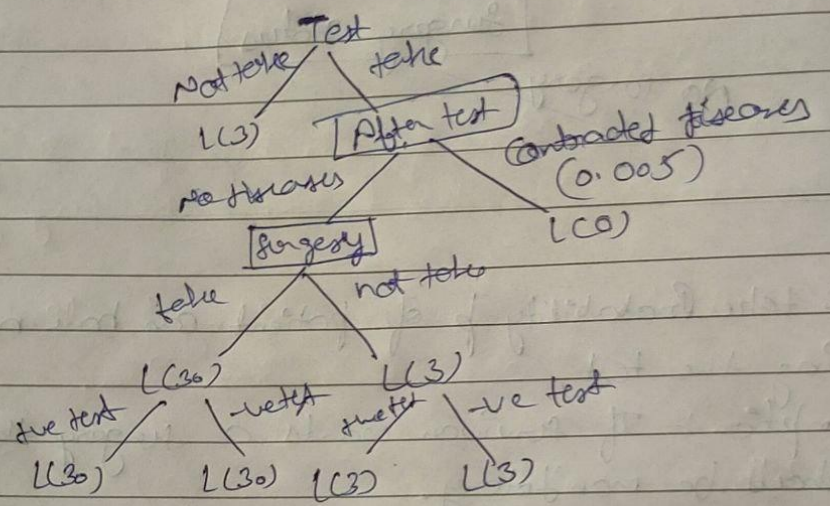
∴ Decision Tree :-



∴ Assumption :- If patient have new disease, ... they will fatal.

Ans (4) - Given :- Probability of contracting the new disease during the test is 0.005.

y Decision Tree :-



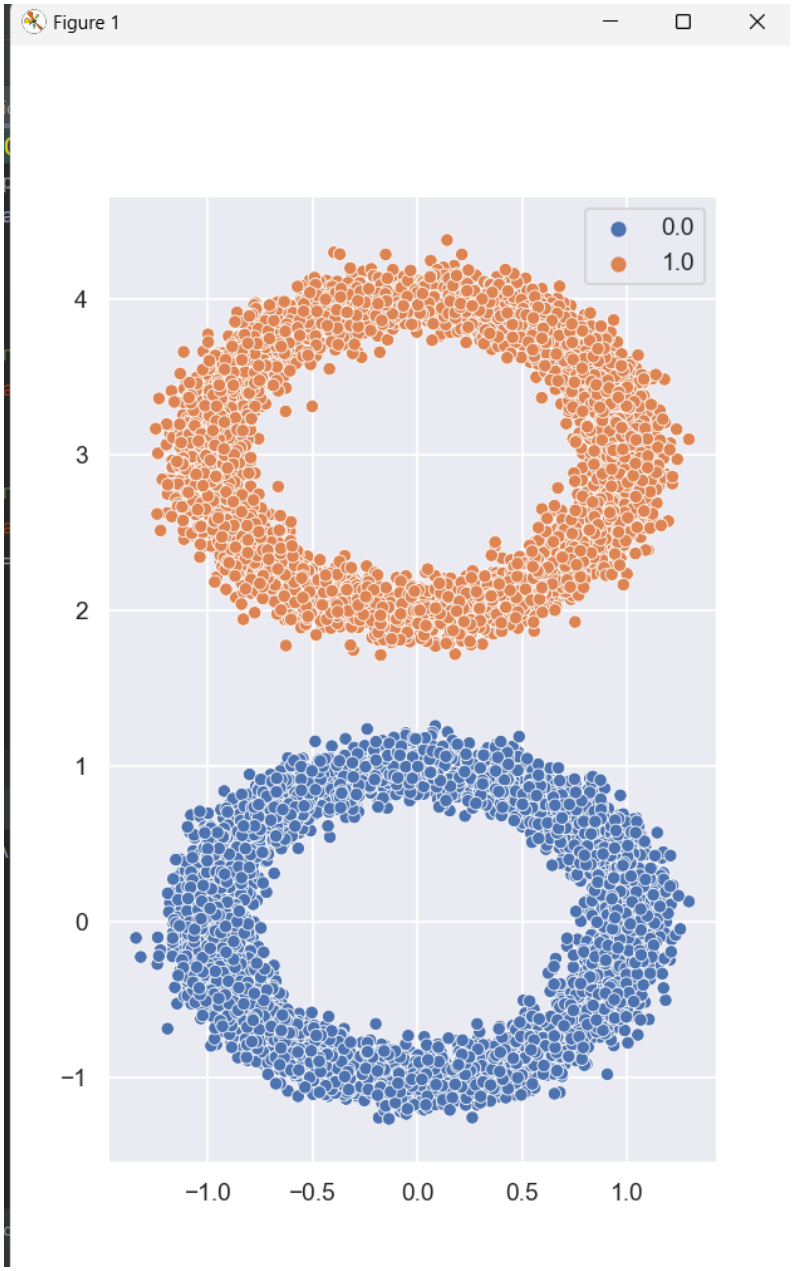
y We know that Probability of successful surgery given test is true is 0.997 which is greater than the Probability of having contracted diseases during test which is 0.005.

∴ Here also test be conducted prior to operation. Ans

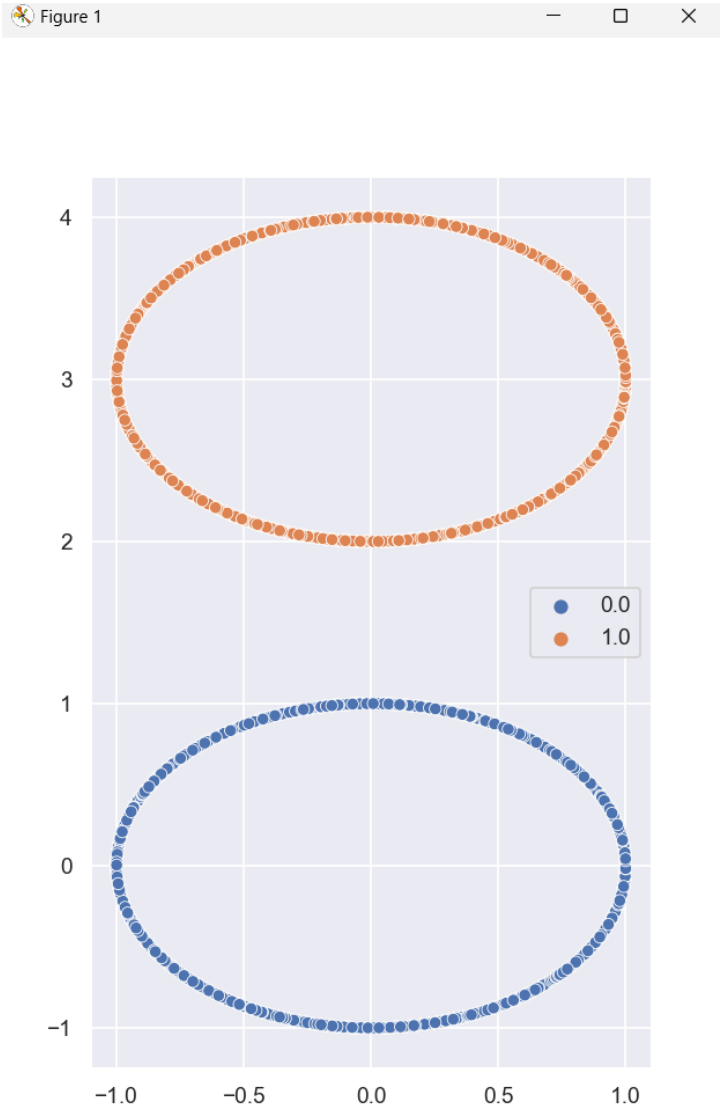
SECTION-B

Ans-2

2D plot with Noise :

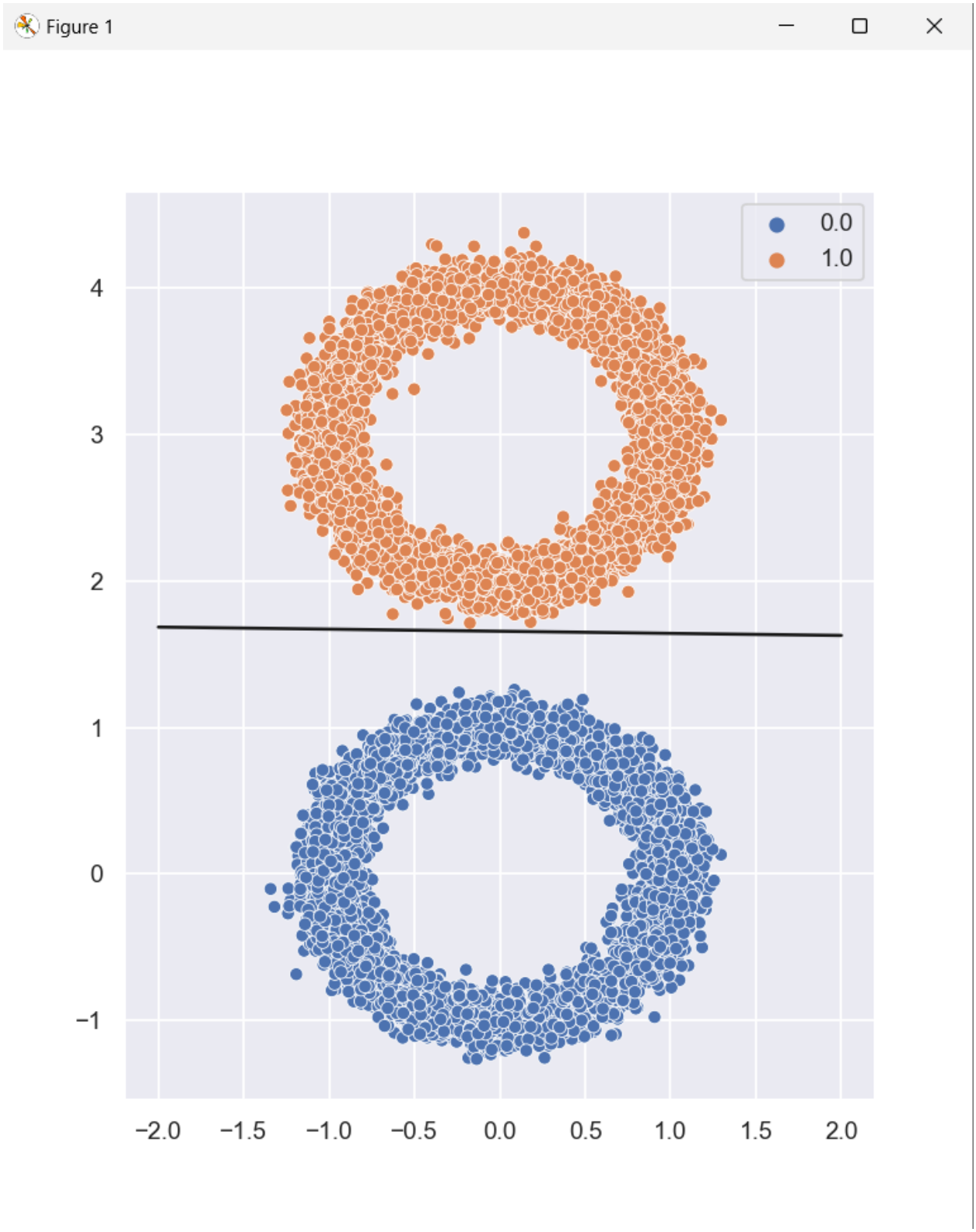


2D plot without Noise :-

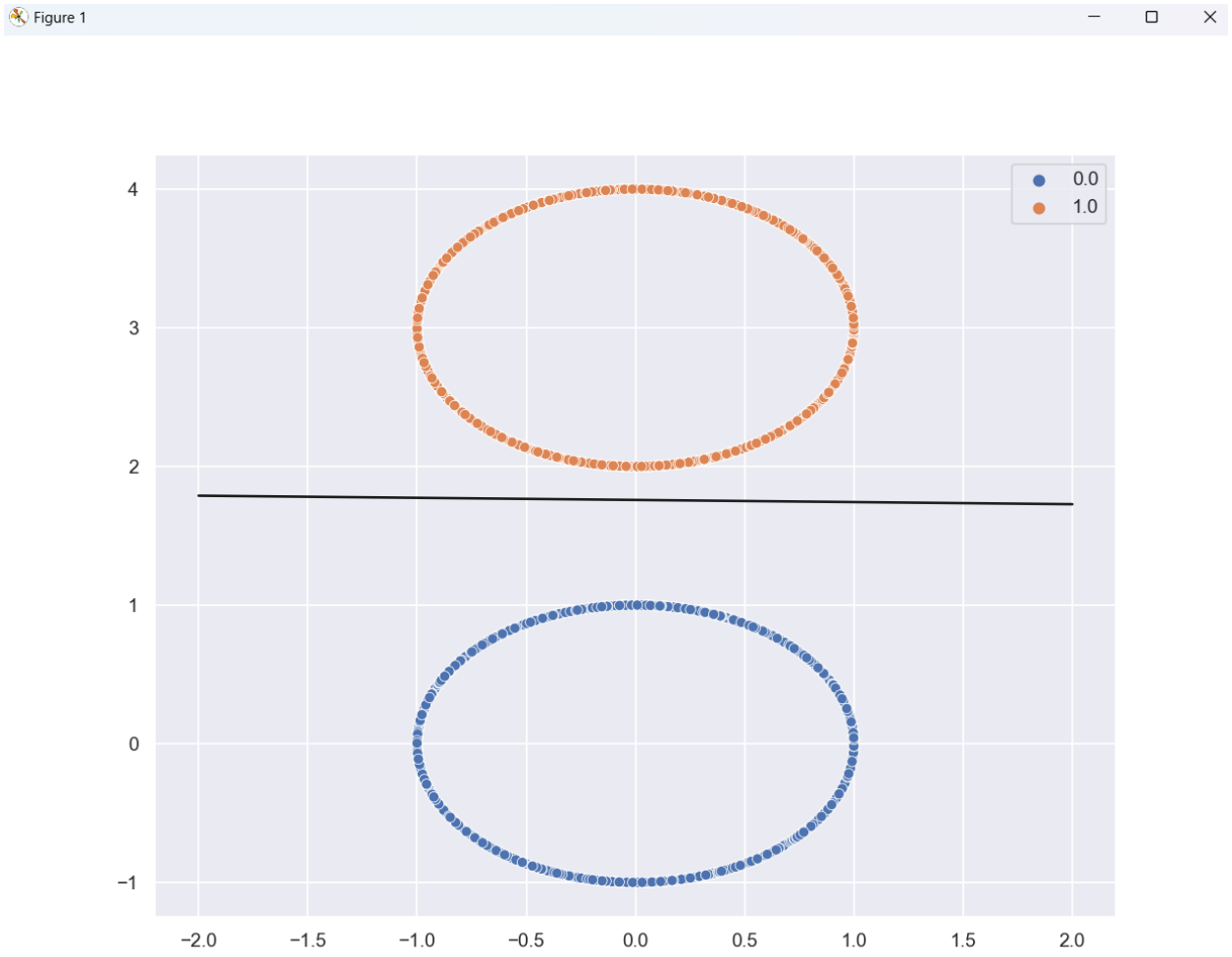


Ans-3

Decision Boundary for with Noise Data:-

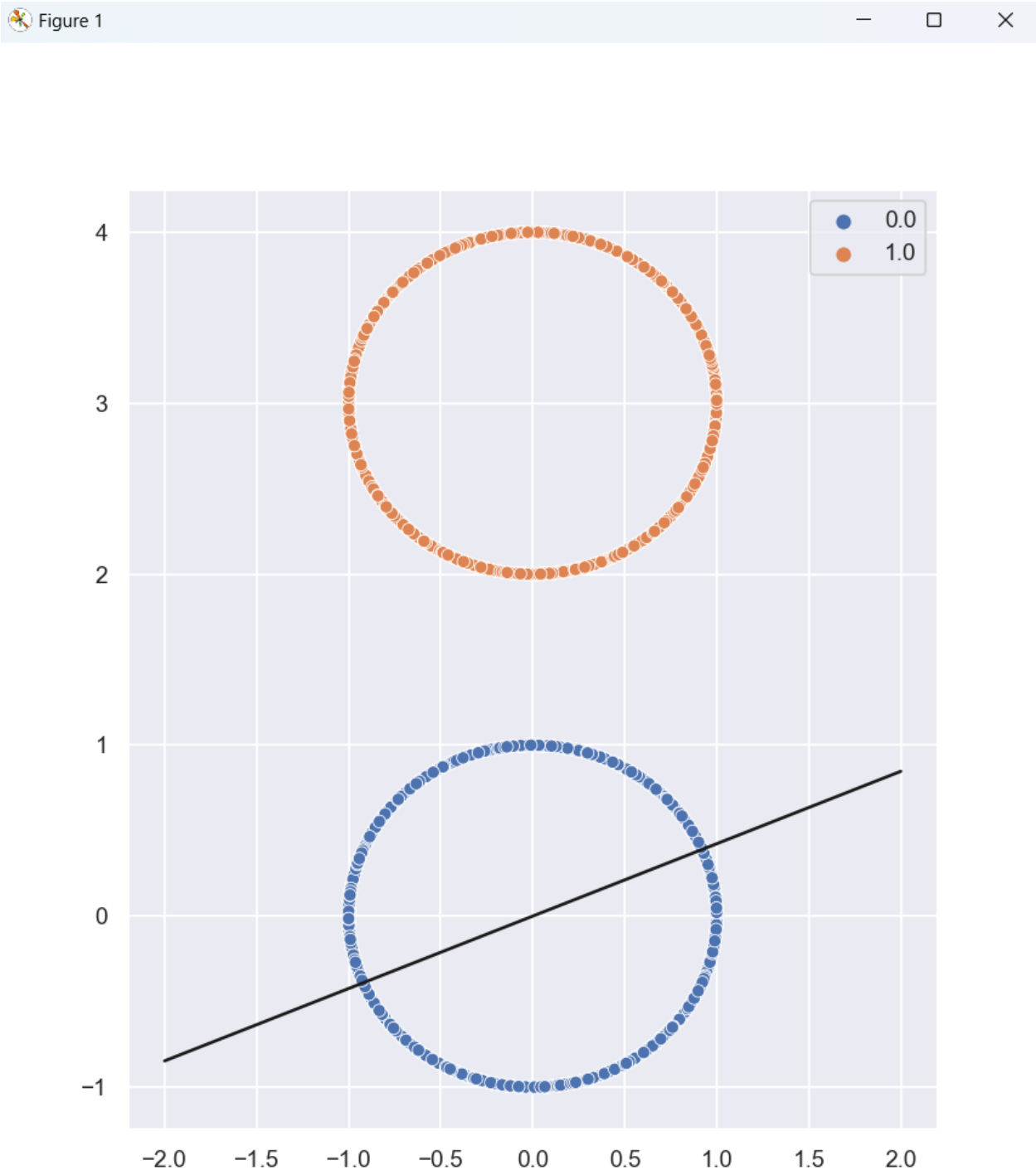


Decision Boundary for without Noise Data :-



Ans-4

Decision boundary for without noise and fixed bias :-



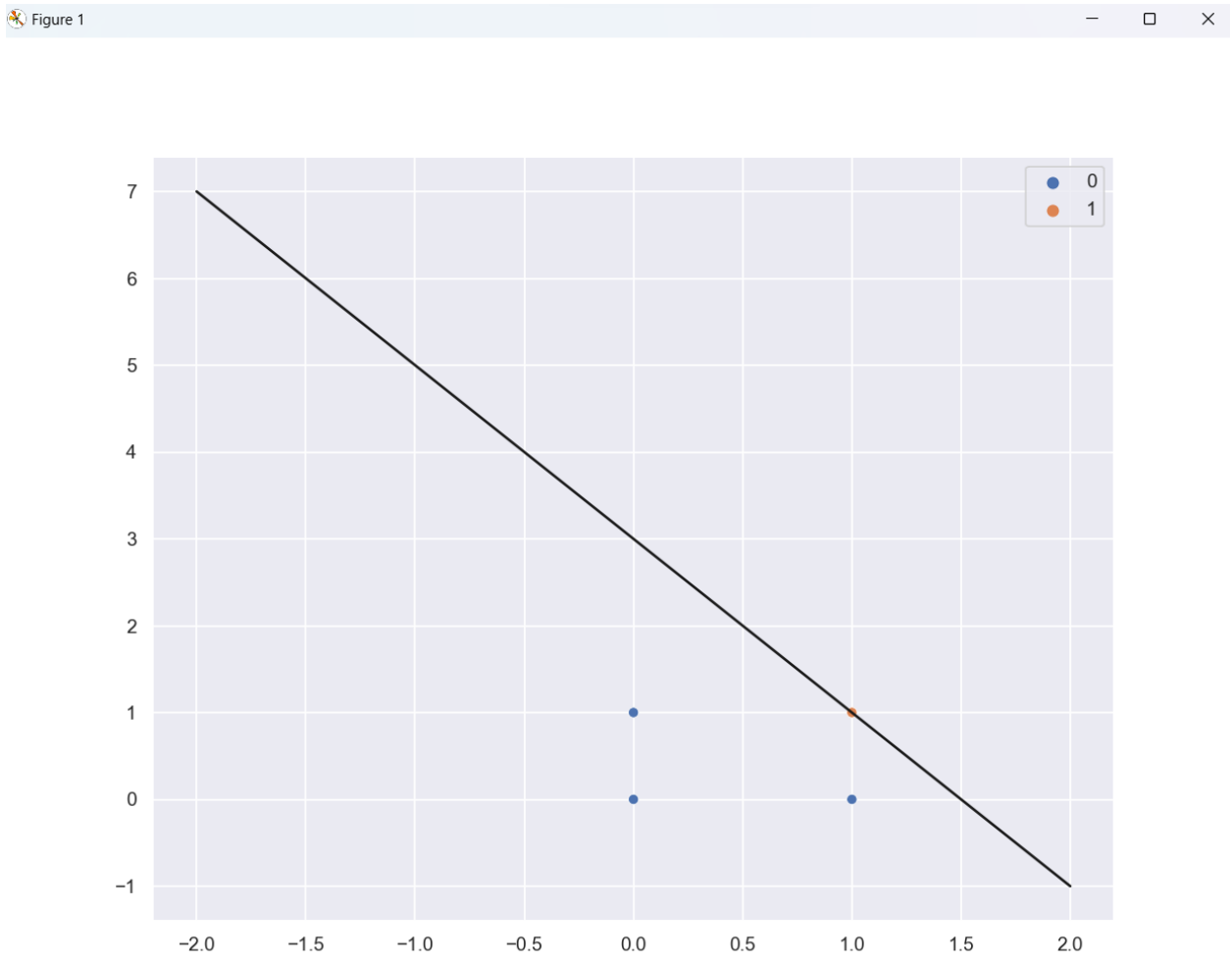
Report :-

We know that decision boundary helps us in classifying the data.So,In 2.3 we have a decision boundary for both types of data i.e., with noise and without noise but in this part we haven't any decision boundary because in 2.3 we use PTA with learnable bias and in this part we are fixed the bias. We know a kind of general equation in PTA which is w1x1 + w2x2 + bias = 0, now this is a equation of straight line if we fix the bias or we put bias = 0 then we have a equation w1x1 + w2x2 = 0, now this equation should always pass with the coordinate (0,0) so basically what happens w1 and w2 will rotate around (0,0) and in our both the data the class label 0 have circle shape in 2 D plot with center (0,0) and this shows that we can't make a decision boundary with

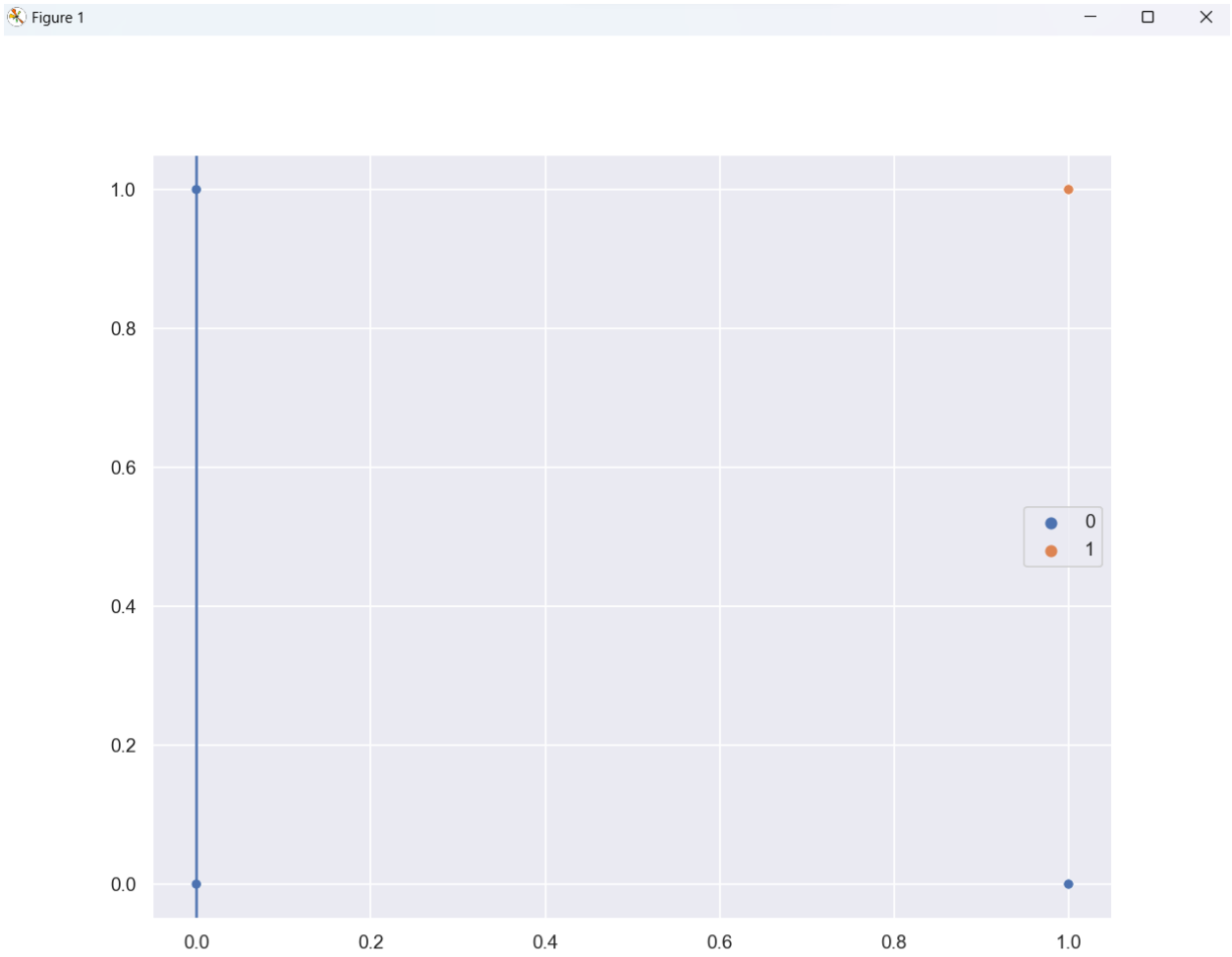
bias 0 because decision boundary with bias = 0 we get always cuts the class label 0 in 2 parts in which 1 part will surely lie on the side of class label 1 but in case of 2.3 we have a non-zero bias so bias will shift this point and we have a correct decision boundary in 2.3.

Ans-5

Decision Boundary for AND property with learnable bias :

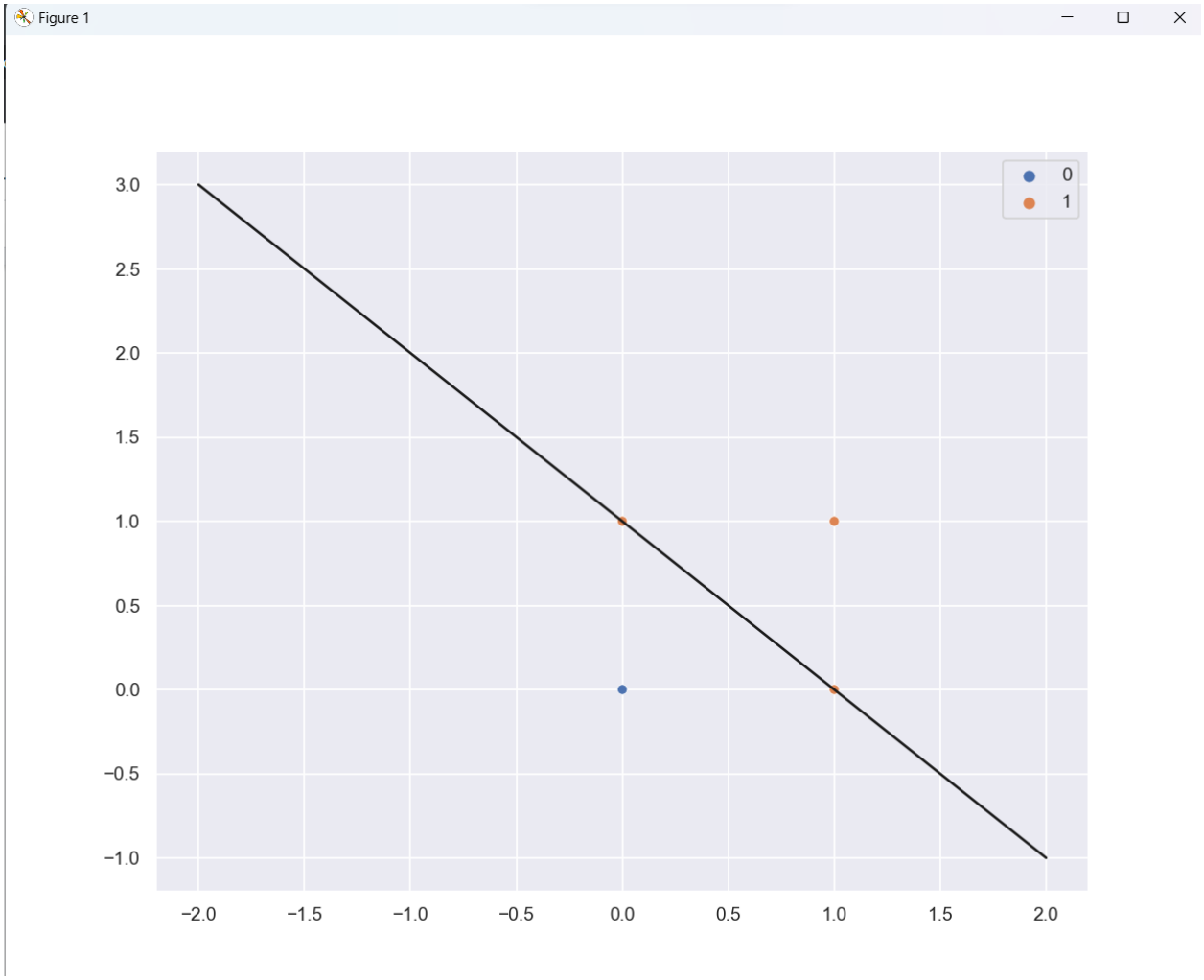


Decision Boundary for AND property with fixed bias :

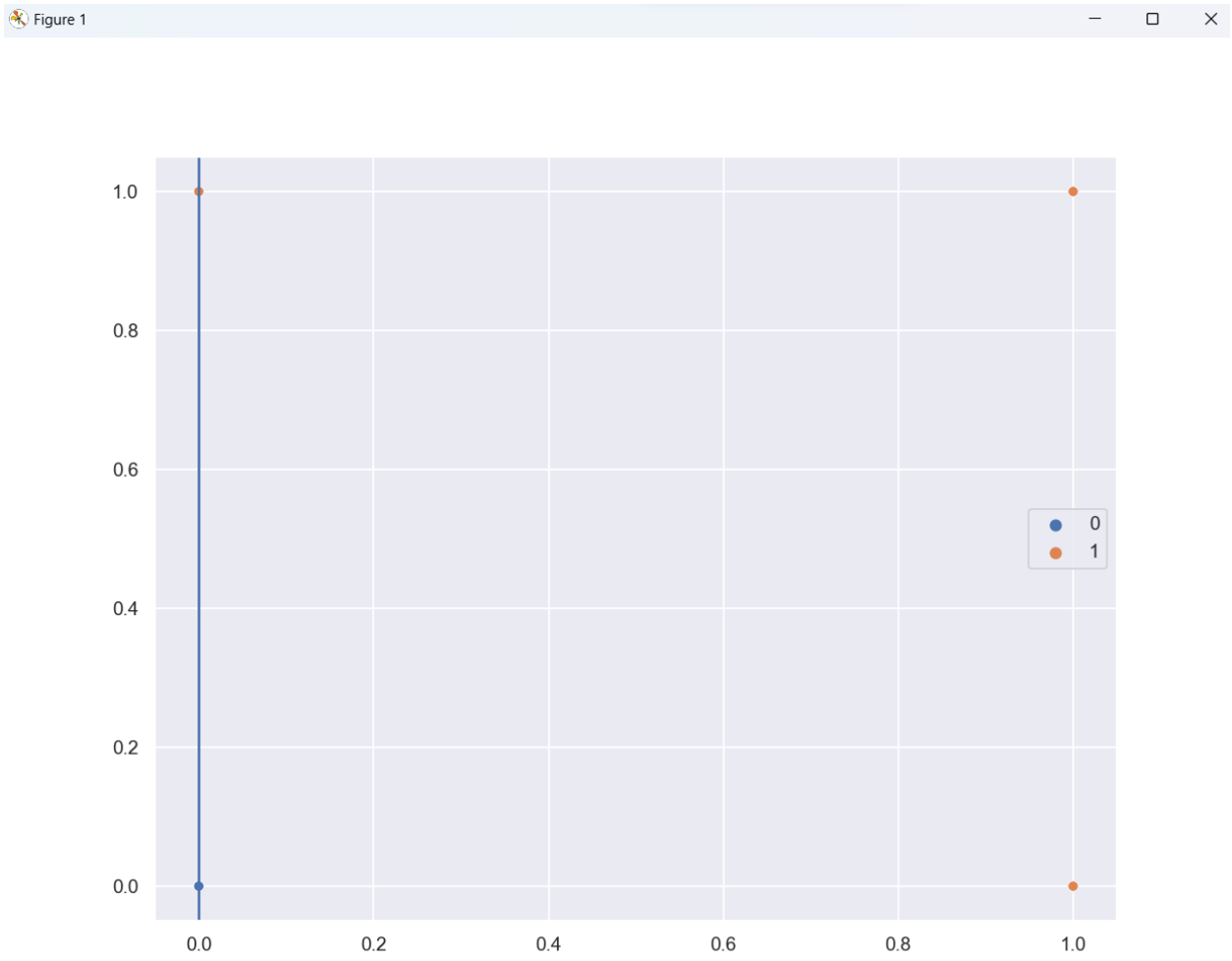


We can infer that if we fix the bias and apply perceptron on AND then we are unable to get a decision boundary for this.

Decision Boundary for OR property with learnable bias :

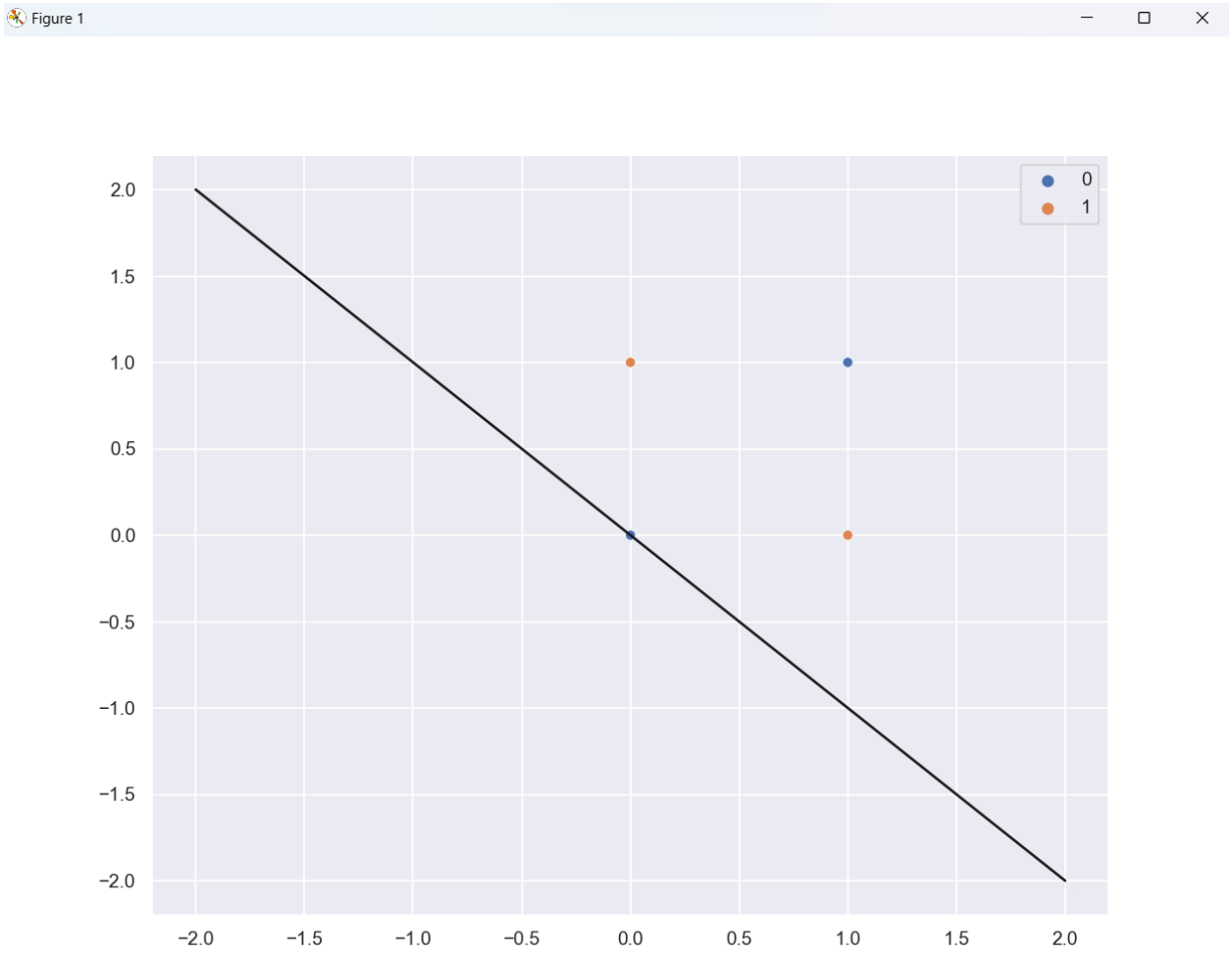


Decision Boundary for OR property with fixed bias :

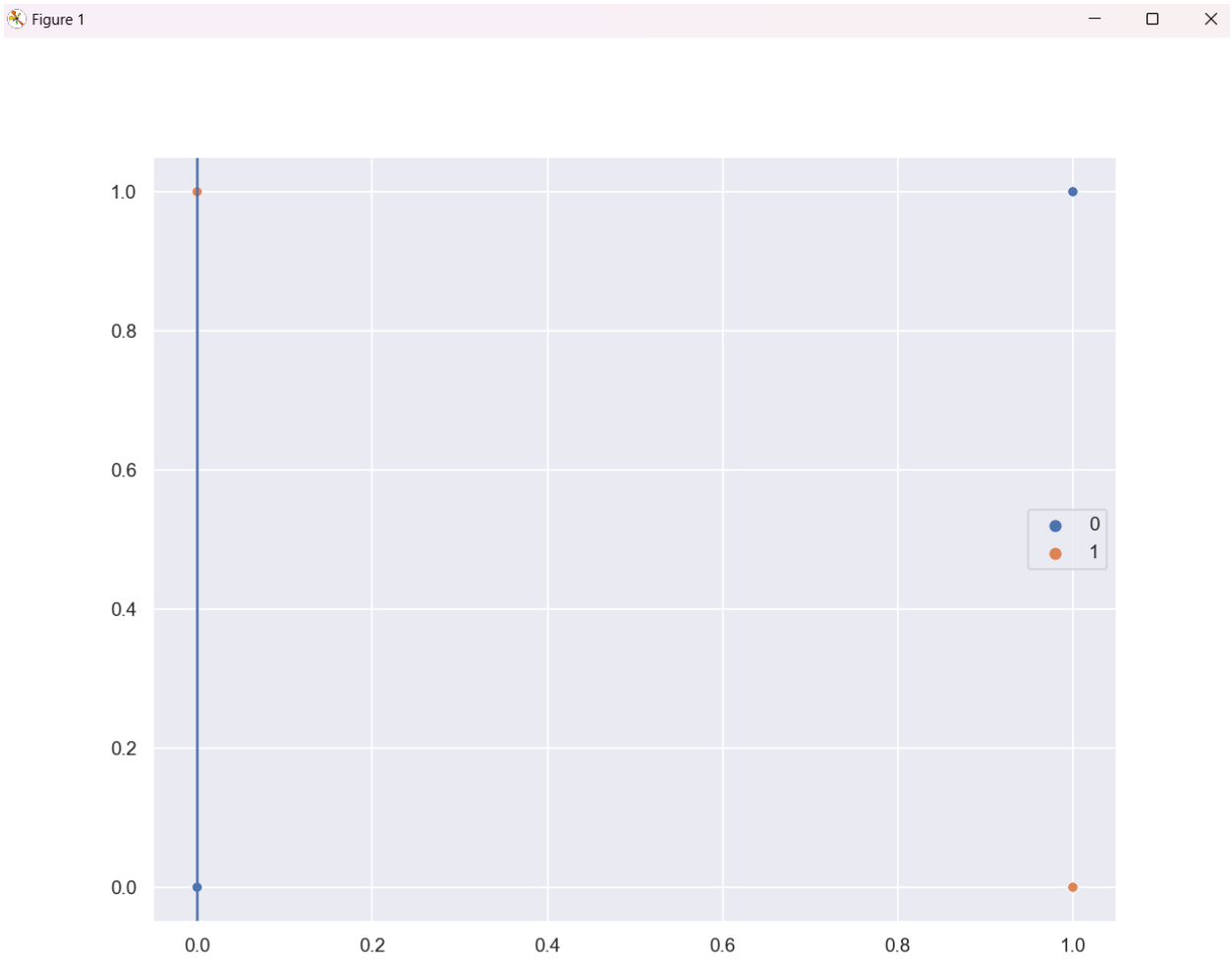


We can infer that if we fix the bias and apply perceptron on AND then we are unable to get a decision boundary for this.

Decision boundary for XOR property with fixed bias :



Decision boundary for XOR property with learnable bias :



We know that by single perceptron we can't classify the XOR and same here we get for both case i.e., when we use learnable bias and fixed bias we get an unclassified plot.

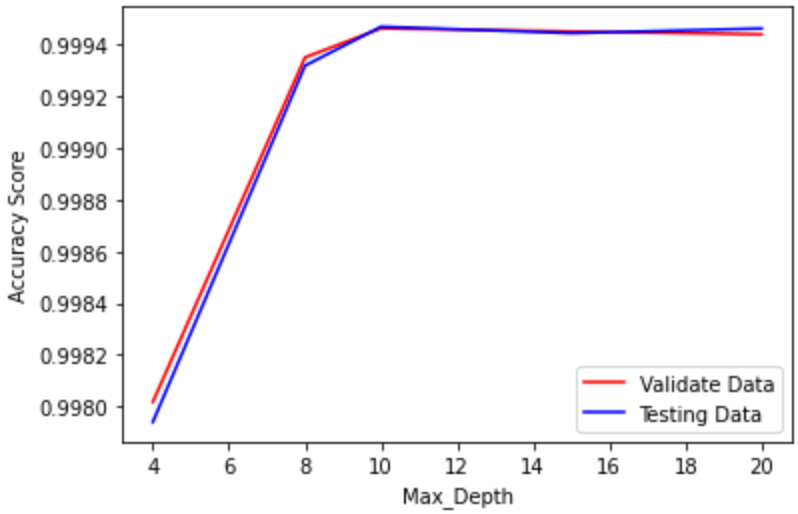
Ans-6 Given :- Hyperplane equation and point is given so,
Assumption :- We will say class 1 to >= 0 values which we get on putting the point in the hyperplane equation and vice versa for class 0.
We will put the point in the hyperplane equation and if we get a negative value then from our assumption it will classify class 0 and vice versa for class 1.

SECTION-C

Ans-1
Scores from Criterion = GINI :-

```
Accuracy Score For Validate (max depth = 4) : 0.9980183083621902
Precision Score For Validate (max depth = 4) : 0.9980183083621902
Recall Score For Validate (max depth = 4) : 0.9980183083621902
Accuracy Score For Test (max depth = 4) : 0.997940594964629
Precision Score For Test (max depth = 4) : 0.997940594964629
Recall Score For Test (max depth = 4) : 0.997940594964629

Accuracy Score For Validate (max depth = 8) : 0.9993531502497114
Precision Score For Validate (max depth = 8) : 0.9993531502497114
Recall Score For Validate (max depth = 8) : 0.9993531502497114
Accuracy Score For Test (max depth = 8) : 0.999318864927258
Precision Score For Test (max depth = 8) : 0.999318864927258
Recall Score For Test (max depth = 8) : 0.999318864927258

Accuracy Score For Validate (max depth = 10) : 0.9994582919052354
Precision Score For Validate (max depth = 10) : 0.9994582919052354
Recall Score For Validate (max depth = 10) : 0.9994582919052354
Accuracy Score For Test (max depth = 10) : 0.9994697203460532
Precision Score For Test (max depth = 10) : 0.9994697203460532
Recall Score For Test (max depth = 10) : 0.9994697203460532

Accuracy Score For Validate (max depth = 15) : 0.9994628632815625
Precision Score For Validate (max depth = 15) : 0.9994628632815625
Recall Score For Validate (max depth = 15) : 0.9994628632815625
Accuracy Score For Test (max depth = 15) : 0.9994674346578897
Precision Score For Test (max depth = 15) : 0.9994674346578897
Recall Score For Test (max depth = 15) : 0.9994674346578897

Accuracy Score For Validate (max depth = 20) : 0.9994605775933989
Precision Score For Validate (max depth = 20) : 0.9994605775933989
Recall Score For Validate (max depth = 20) : 0.9994605775933989
Accuracy Score For Test (max depth = 20) : 0.9994262922709455
Precision Score For Test (max depth = 20) : 0.9994262922709455
Recall Score For Test (max depth = 20) : 0.9994262922709455
```

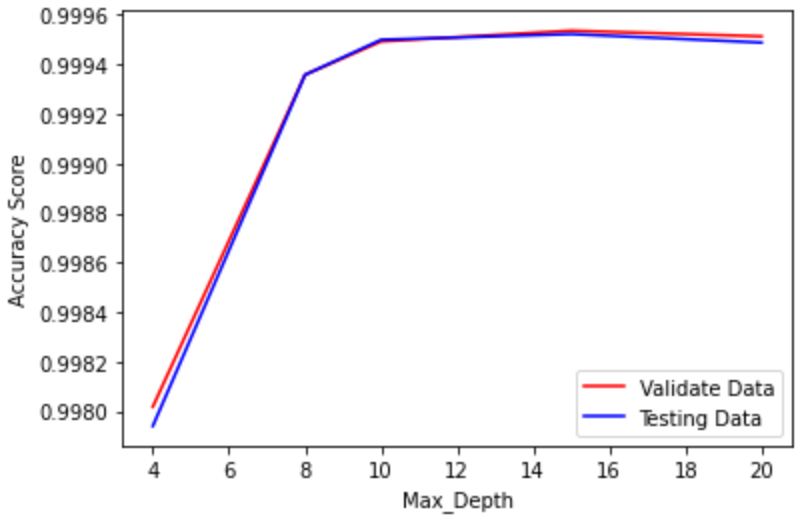
PLOT FOR GINI ACCURACY SCORE ON TEST AND VALIDATE DATA :-



Scores from Criterion = Entropy :

```
Accuracy Score For Validate (max depth = 4) : 0.9980183083621902
Precision Score For Validate (max depth = 4) : 0.9980183083621902
Recall Score For Validate (max depth = 4) : 0.9980183083621902
Accuracy Score For Test (max depth = 4) : 0.997940594964629
Precision Score For Test (max depth = 4) : 0.997940594964629
Recall Score For Test (max depth = 4) : 0.997940594964629

Accuracy Score For Validate (max depth = 8) : 0.9993577216260385
Precision Score For Validate (max depth = 8) : 0.9993577216260385
Recall Score For Validate (max depth = 8) : 0.9993577216260385
Accuracy Score For Test (max depth = 8) : 0.9993577216260385
Precision Score For Test (max depth = 8) : 0.9993577216260385
Recall Score For Test (max depth = 8) : 0.9993577216260385

Accuracy Score For Validate (max depth = 10) : 0.9994925772276888
Precision Score For Validate (max depth = 10) : 0.9994925772276888
Recall Score For Validate (max depth = 10) : 0.9994925772276888
Accuracy Score For Test (max depth = 10) : 0.9994994342921795
Precision Score For Test (max depth = 10) : 0.9994994342921795
Recall Score For Test (max depth = 10) : 0.9994994342921795

Accuracy Score For Validate (max depth = 15) : 0.9995360053027965
Precision Score For Validate (max depth = 15) : 0.9995360053027965
Recall Score For Validate (max depth = 15) : 0.9995360053027965
Accuracy Score For Test (max depth = 15) : 0.9995222911738152
Precision Score For Test (max depth = 15) : 0.9995222911738152
Recall Score For Test (max depth = 15) : 0.9995222911738152

Accuracy Score For Validate (max depth = 20) : 0.9995131484211609
Precision Score For Validate (max depth = 20) : 0.9995131484211609
Recall Score For Validate (max depth = 20) : 0.9995131484211609
Accuracy Score For Test (max depth = 20) : 0.9994880058513617
Precision Score For Test (max depth = 20) : 0.9994880058513617
Recall Score For Test (max depth = 20) : 0.9994880058513617
```

PLOT FOR ENTROPY ACCURACY SCORE ON TEST AND VALIDATE DATA :-



We can see that we are getting maximum accuracy for max_depth = 15 with entropy criterion so we will use entropy criterion for next parts.

Ans-2:
Scores from Ensembling :-

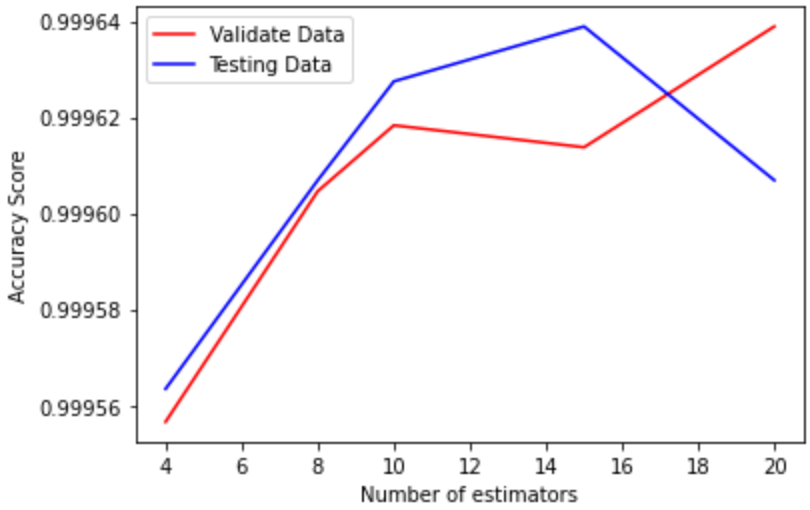
```
Accuracy Score For Validate : 0.9978217391801236
Precision Score For Validate : 0.9978217391801236
Recall Score For Validate : 0.9978217391801236
Accuracy Score For Test : 0.9976503125678564
Precision Score For Test : 0.9976503125678564
Recall Score For Test : 0.9976503125678564
```

Comparison between 1 and 2 :-
In Ensembling we get higher accuracy for validation as well as for testing data than maximum accuracy we get from entropy model that is for max_depth = 15.So ensembling is performing better than decision classifier.

Ans-3:
Scores of all Adaboost model :-

```
Accuracy Score For Validate (number of estimators = 4) : 0.9995565764962686
Precision Score For Validate (number of estimators = 4) : 0.9995565764962686
Recall Score For Validate (number of estimators = 4) : 0.9995565764962686
Accuracy Score For Test (number of estimators = 4) : 0.9995634335607593
Precision Score For Test (number of estimators = 4) : 0.9995634335607593
Recall Score For Test (number of estimators = 4) : 0.9995634335607593

/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:993: DataConve
  y = column_or_1d(y, warn=True)
Accuracy Score For Validate (number of estimators = 8) : 0.9996045759477035
Precision Score For Validate (number of estimators = 8) : 0.9996045759477035
Recall Score For Validate (number of estimators = 8) : 0.9996045759477035
Accuracy Score For Test (number of estimators = 8) : 0.999606861635867
Precision Score For Test (number of estimators = 8) : 0.999606861635867
Recall Score For Test (number of estimators = 8) : 0.999606861635867

/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:993: DataConve
  y = column_or_1d(y, warn=True)
Accuracy Score For Validate (number of estimators = 10) : 0.9996182900766848
Precision Score For Validate (number of estimators = 10) : 0.9996182900766848
Recall Score For Validate (number of estimators = 10) : 0.9996182900766848
Accuracy Score For Test (number of estimators = 10) : 0.9996274328293391
Precision Score For Test (number of estimators = 10) : 0.9996274328293391
Recall Score For Test (number of estimators = 10) : 0.9996274328293391

/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:993: DataConve
  y = column_or_1d(y, warn=True)
Accuracy Score For Validate (number of estimators = 15) : 0.9996137187003578
Precision Score For Validate (number of estimators = 15) : 0.9996137187003578
Recall Score For Validate (number of estimators = 15) : 0.9996137187003578
Accuracy Score For Test (number of estimators = 15) : 0.9996388612701569
Precision Score For Test (number of estimators = 15) : 0.9996388612701569
Recall Score For Test (number of estimators = 15) : 0.9996388612701569

/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:993: DataConve
  y = column_or_1d(y, warn=True)
Accuracy Score For Validate (number of estimators = 20) : 0.9996388612701569
Precision Score For Validate (number of estimators = 20) : 0.9996388612701569
Recall Score For Validate (number of estimators = 20) : 0.9996388612701569
Accuracy Score For Test (number of estimators = 20) : 0.999606861635867
Precision Score For Test (number of estimators = 20) : 0.999606861635867
Recall Score For Test (number of estimators = 20) : 0.999606861635867
```

PLOT BETWEEN ACCURACY AND NUMBER OF ESTIMATORS ON VALIDATE AND TEST DATA:



Compare Between Adaboost and RF :-

RF perform better than Adaboost on the basis of accuracy score we get and this is may be happen because adaboost aims to decrease bias not variance but on the other side Random forest aims to decrease variance not bias so that's why we have testing accuracy in random forest is higher than adaboost accuracy.