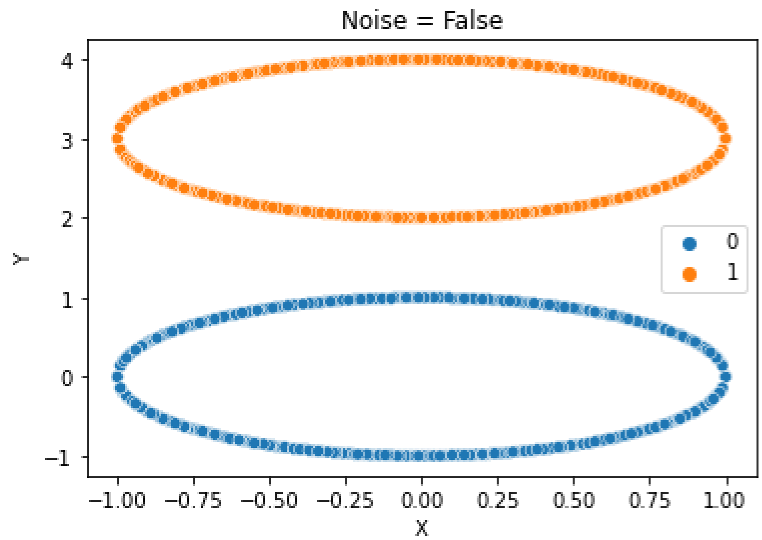
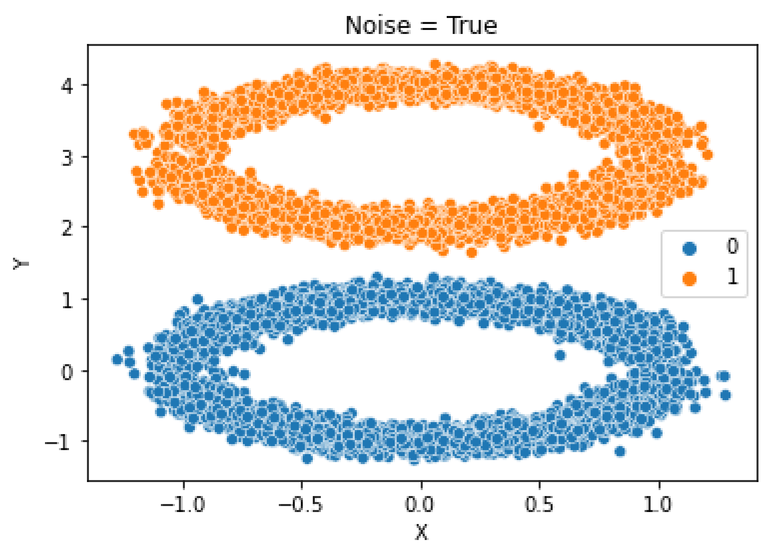
<u>ML Assignment</u>

Ishit Bajpai
2020380

Q2.
1.Code
2.

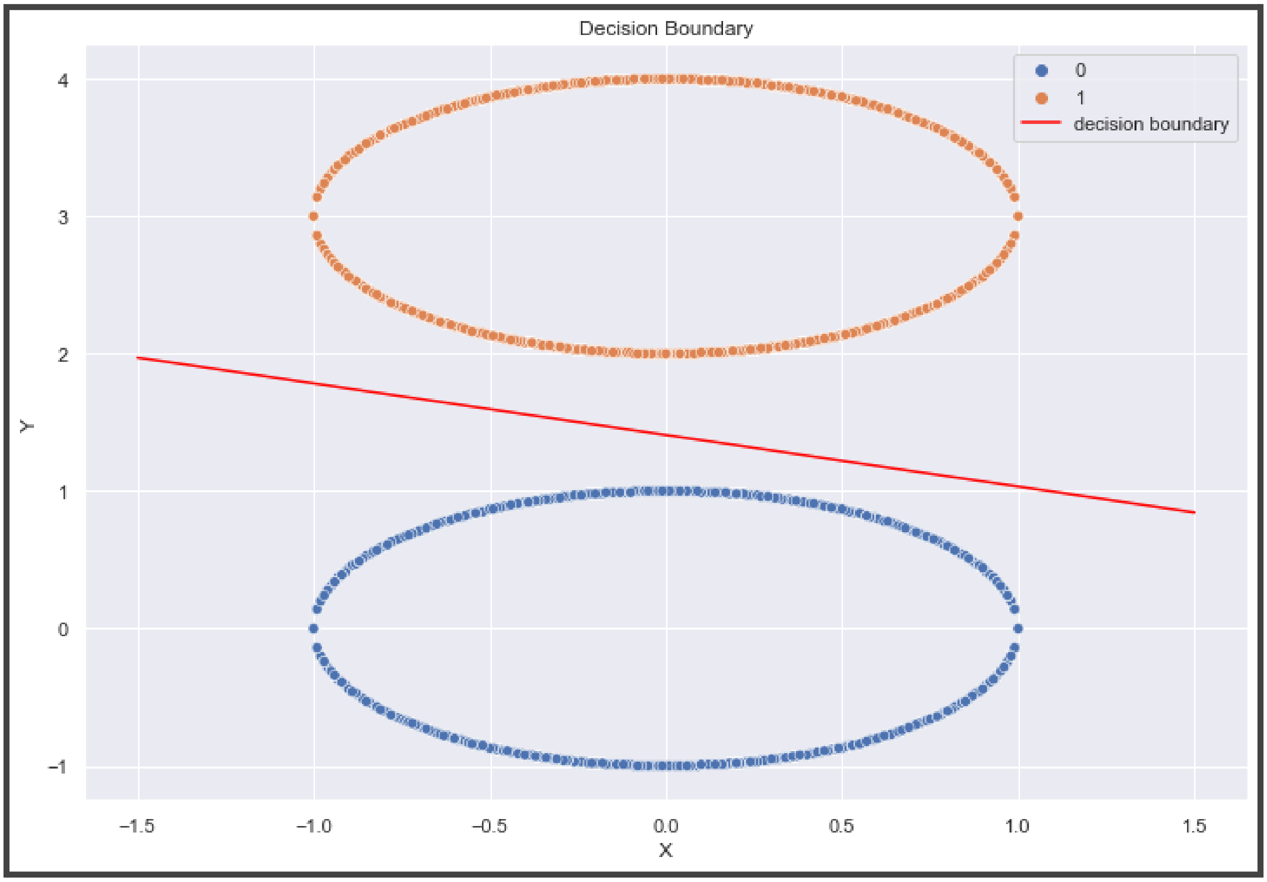




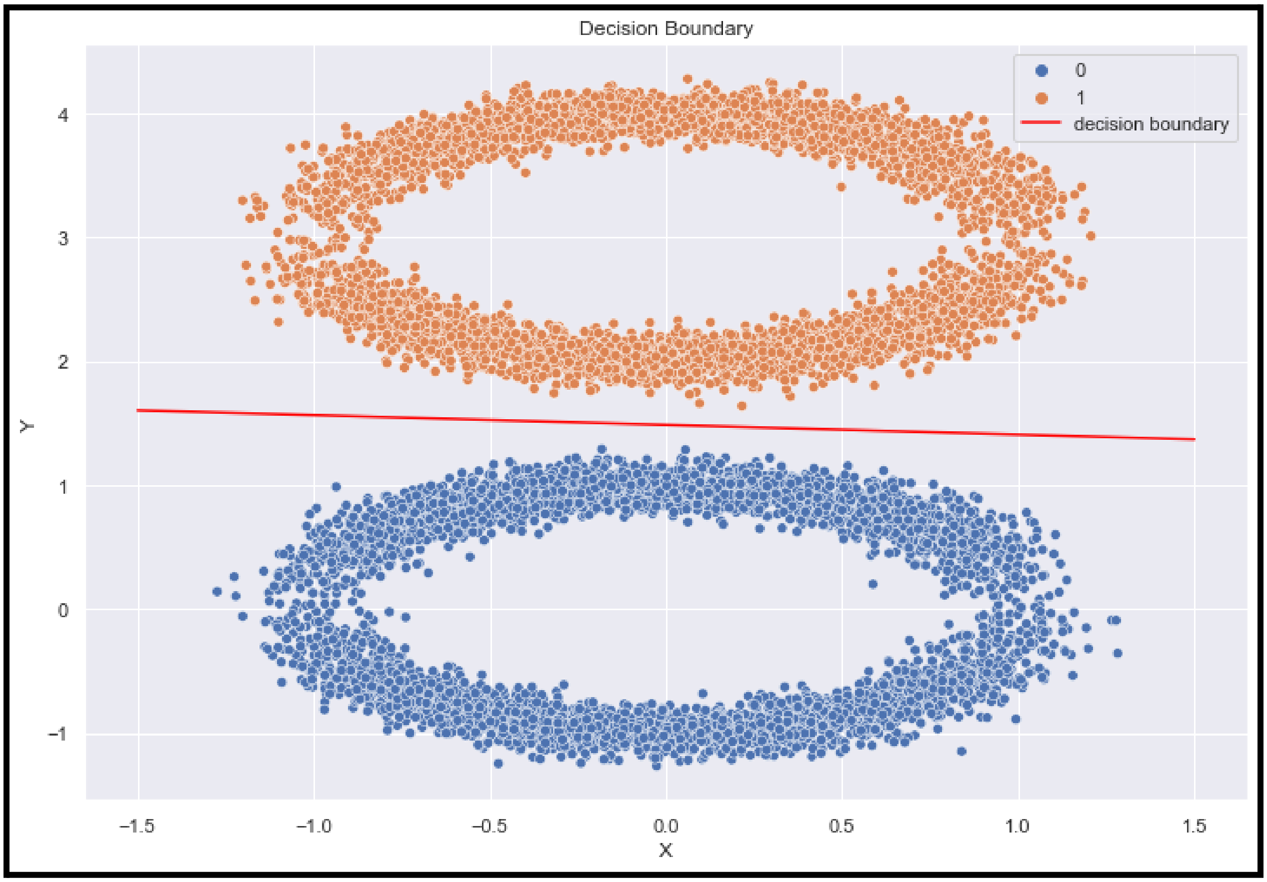
Without Noise : Points lie exactly on the equation of circle
With Noise of mean = 0 and standard deviation = 0.1 Points are not exactly on the equation of circle.

3.

Noise = False

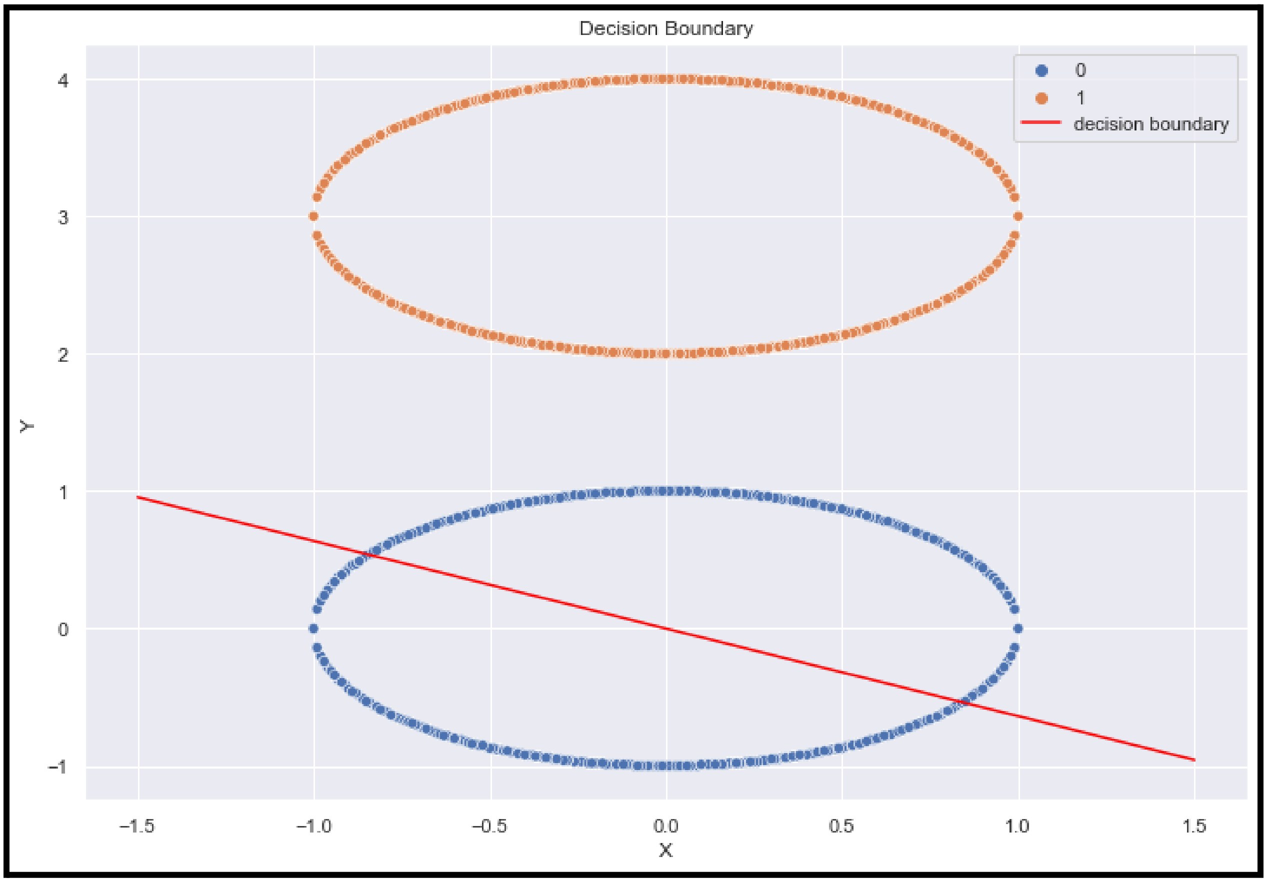


Noise = True



For both the cases there exists a decision boundary as labels are linearly separable.
In without noise two circles are plotted x^2 + y^2 = 1 and x^2 +(y-3)^2=1.Thus there is ample of space in the region y=[+1,+2] for a line to pass through dividing them into separate classes.
With noise standard deviation is 0.1(which is very small) and thus points will not be that far away from the original equation and will still leave a space in the region y=+1 to y+2 for the decision boundary to pass through , thus again making it linearly separable.

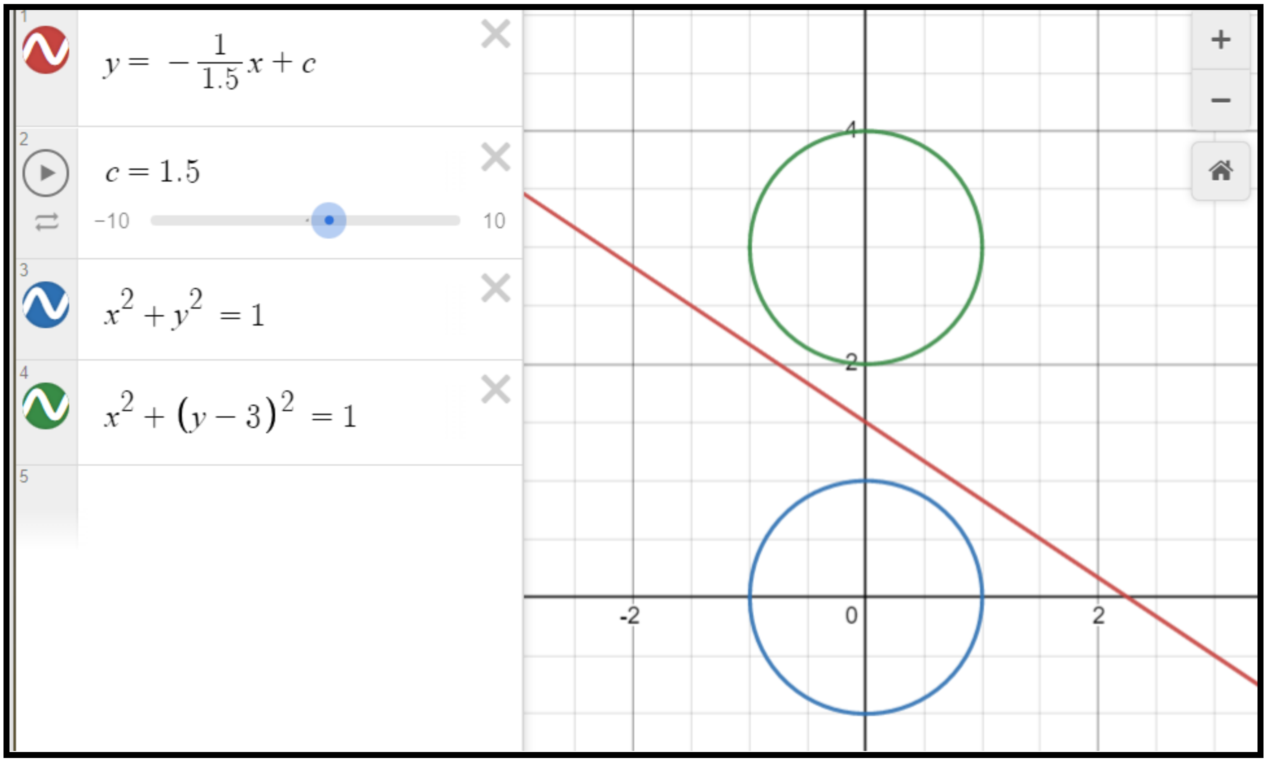
4.



In Q 2.3 , Decision boundary visibly linearly separated the data. However with fixed bias equal to 0 data is not linearly separable.

Bias is a necessary component as it helps us to add constant value to the function.
Without bias(fixed and equal to 0) we can only adjust/tune the weights of the function according to the given data.That is, we can only change the slope(angle) of the activation function.

If bias remains 0 our function will always pass through 0 as(y = w.x+0) and may not be tuned according to the data.
As shown in the above figure our data pass through (0,0) and misclassify many data points even when the data is linearly separable as shown below.



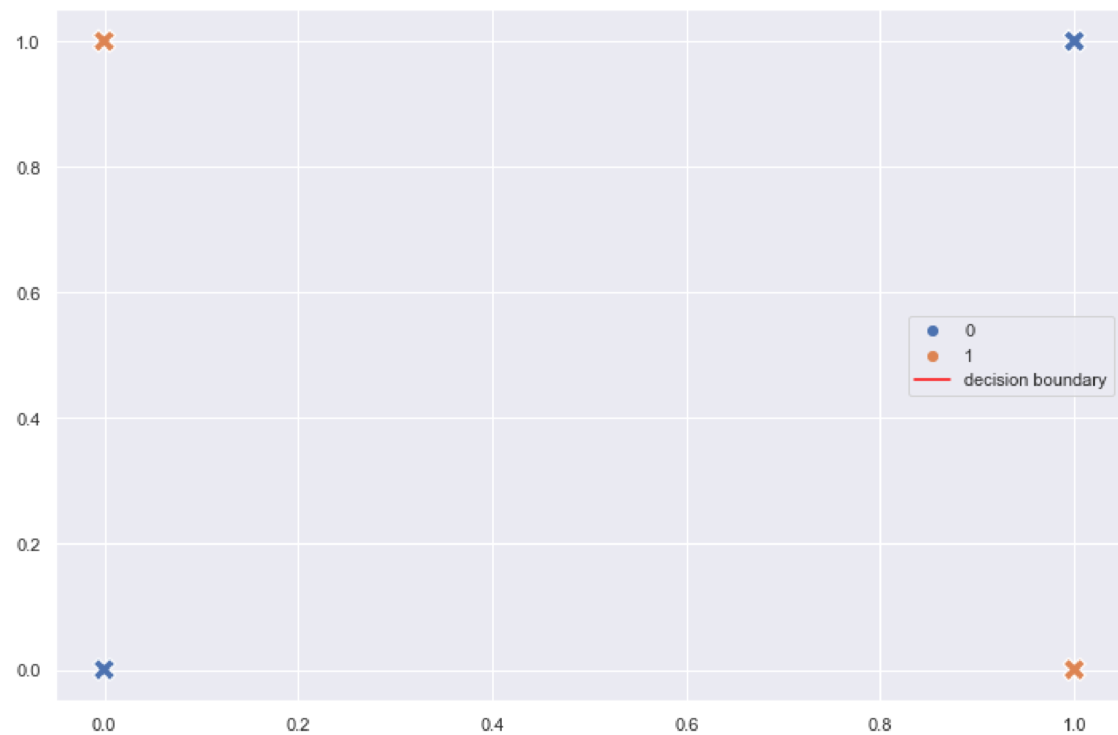
However if bias(bias =1.5) is added , activation function will be shifted upwards and data will be linearly separated.

Therefore bias is necessary to shift our activation function upwards and downwards.In other words it helps us to assign value to the activation function when all other inputs are zero.

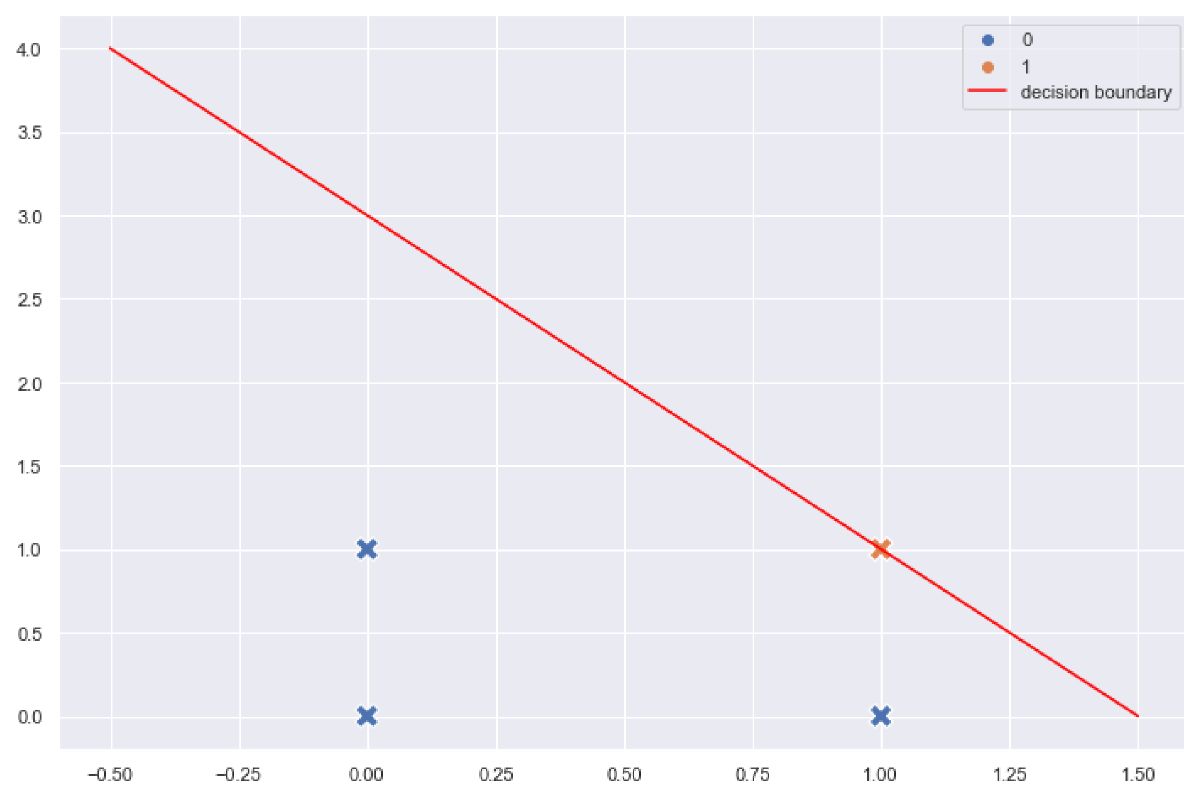
5.

Learnable

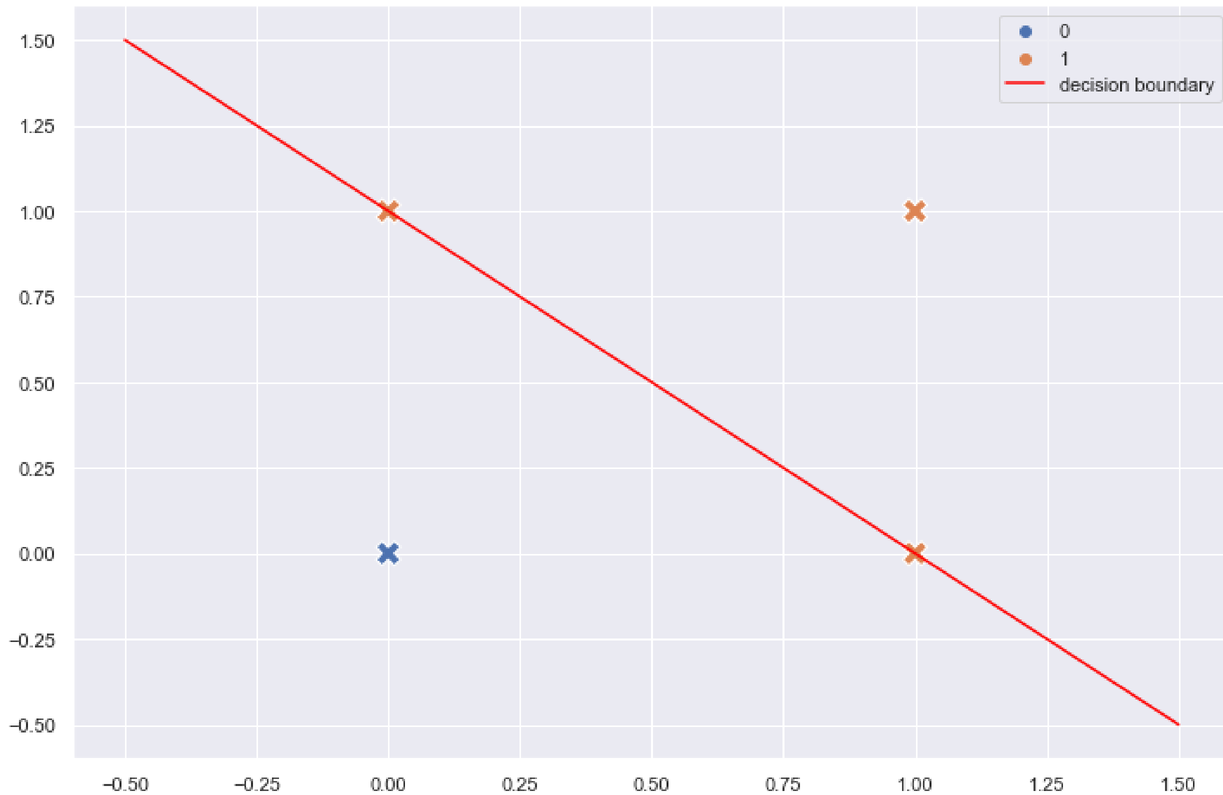
XOR(no decision boundary)



AND(decision boundary exists)

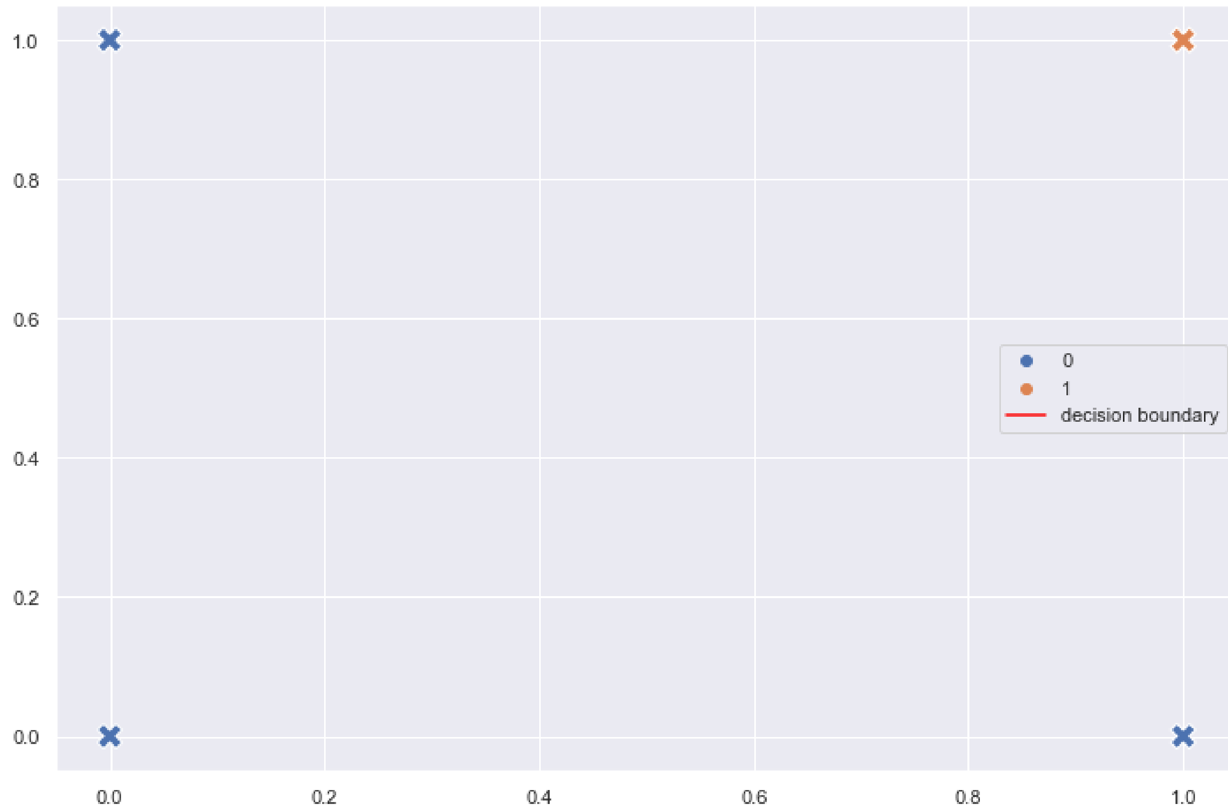


OR(decision boundary exists)

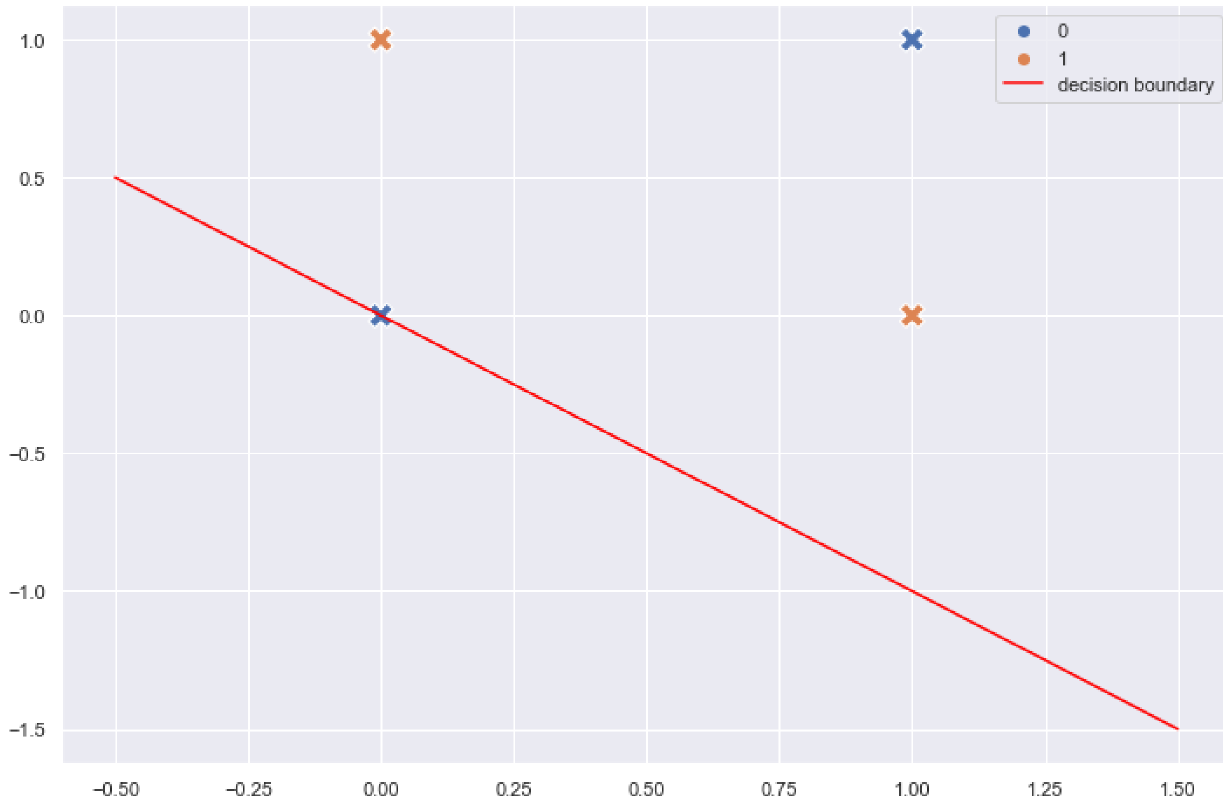


Fixed Bias

AND

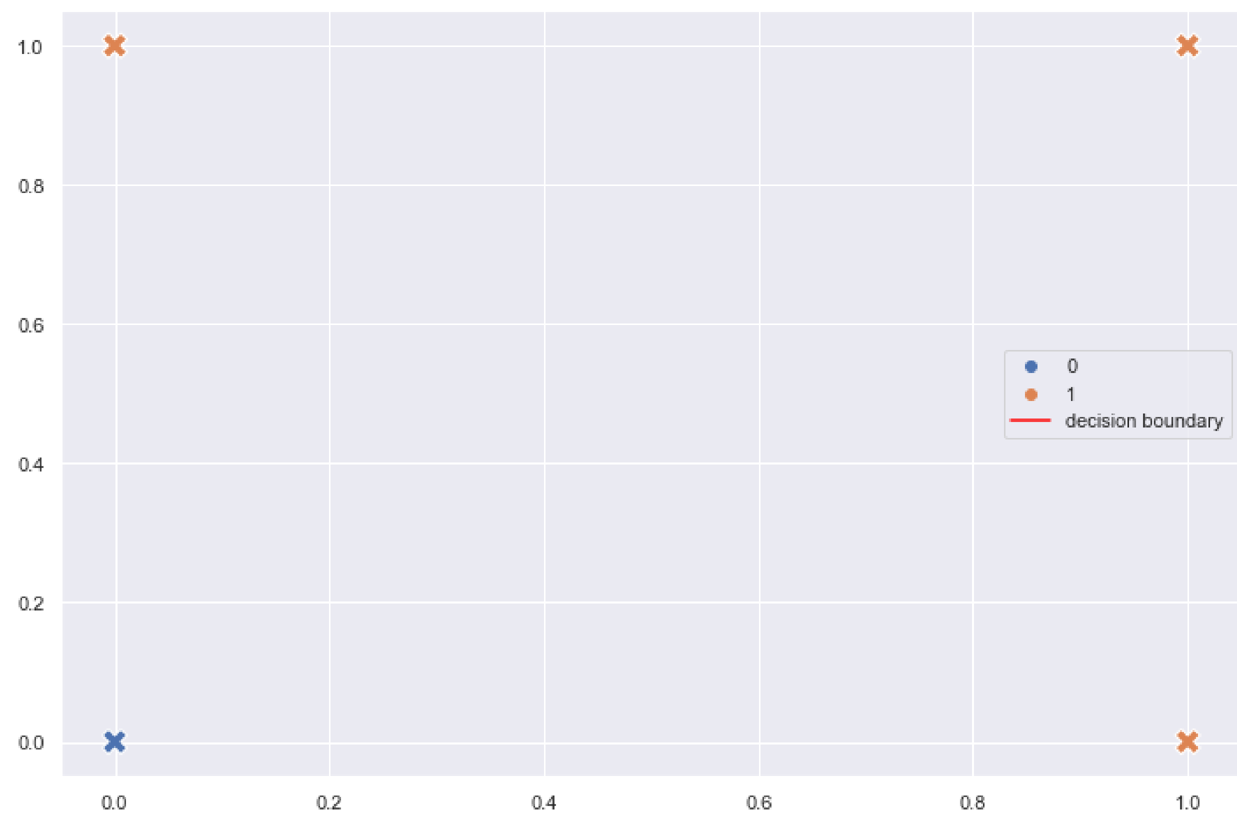


XOR



1.0
0.5
0.0
−0.5
−1.0
−1.5
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
1.25
1.50
0
1
decision boundary

OR



For fixed bias no decision boundary exists which can linearly separate the data.

6.

Let y = Σwi.xi + b , be the equation of hyperplane in the nth dimension.
Given a point (x) in R^n we will classify point as:
0 class if sgn(y) = -1 i.e y(x) < 0
1 class if sgn(y) = +1 i.e y(x) >= 0
Here sgn(v) = +1 if v>= 0
-1 if v<0
Assumption:y(x) < 0 will be classified as 0,while y(x)>=0 as 1.

Q3.

Label encoding is used for label and address columns(ohe not possible due to large range of values of address)

No null values were found during exploration.

(a)

Test(Accuracy Score)

Depth	Gini	Entropy
4	0.9858949861029842	0.9858309866881217
8	0.98650069485079	0.9861738406963136
10	0.9870424041837331	0.9873418300175542
15	**0.988237821825629**	**0.9889875292568754**
20	0.9879703956992393	0.9881761081041545
Average	0.9871292605324751	**0.9873020589526039**

On average, the accuracy of Entropy(98.73%) is better than Gini(98.71%).
Best accuracy score(98.898%) observed was of entropy and max depth = 15.

Val(Accuracy Score)

Depth	Gini	Entropy
4	0.9857235590988882	0.985648131217086
8	0.9865281231714453	0.9860686988004681
10	0.9869372622878877	0.9874401148332358
15	**0.9882926784669397**	**0.9889212441486249**
20	0.9877669689877121	0.9879018248976009
Average	0.9870497184025746	**0.9871960027794031**

On average, the accuracy of Entropy(98.72%) is better than Gini(98.70%).
Best accuracy score(98.892%) observed was of entropy and max depth = 15.

(b)Ensembling(RF)
Entropy is used

	Test	**Validation**
Accuracy Score	**0.9858309866881217**	**0.985648131217086**

Observation:
The Accuracy Score of test and validation are less than the Average accuracy score of part(a) by 0.14% and 0.15% respectively.
Even though we have used decision trees of max depth = 3 and trained it on only 50% of the training data we still get a good accuracy as shown in Table.
In part(a) all parts had depth greater than 3.
If we compare values of Dept = 4 in part a we find that they are approximately the same as the ensembled ones.
Therefore with the help of large number of small(in terms of depth) decision trees we are able to predict with a good accuracy.(and approximately same as trees with larger depth)

(c)ADA BOOST
Base Estimator : Decision tree with max depth = 15

Estimators	Test	Validation
4	**0.9885212478057344**	0.9870675468110006,
8	0.9865761227325922	0.9861235554417788,
10	0.9864024100351082	0.9871818314803978,
15	0.9874172578993563	0.9870698325043885,
20	0.987684684025746	**0.9878515396430662**

Average	0.9873203444997074	0.9870588611761264

Observation:
Adaboost gives an accuracy of 98.7320% and 98.7058% for the Test and Val respectively.
Adaboost's performance is better than Random forest for both cases.
Adaboost outperformed RF on the test set by 0.001489(or 0.149%) accuracy score and on the validation set by 0.00141(or 0.141%).
Ada boost performance was better because boosting reduces bias and variance whereas in RF only variance was reduced.
AdaBoost is also better as it tries to learn from the previous errors and makes classification in successive attempts to correct those errors.