Measuring Classification performance through AUC-ROC curve

TPR

	C +	H −
Cancer +	5 TP	0 FN
H −	1 FP	4 TN

FPR

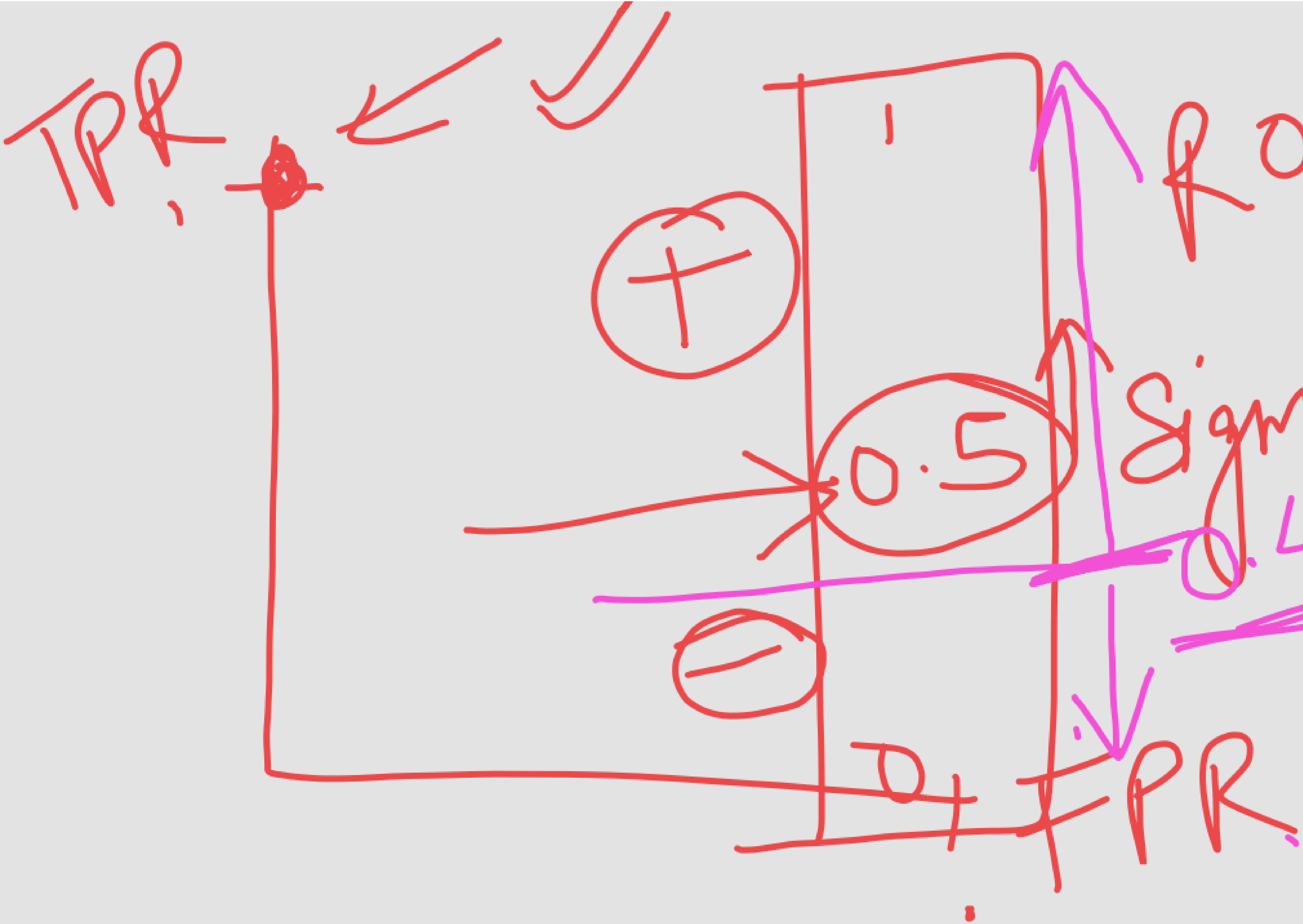
Precision
Recall
F1-score
Accuracy

90%

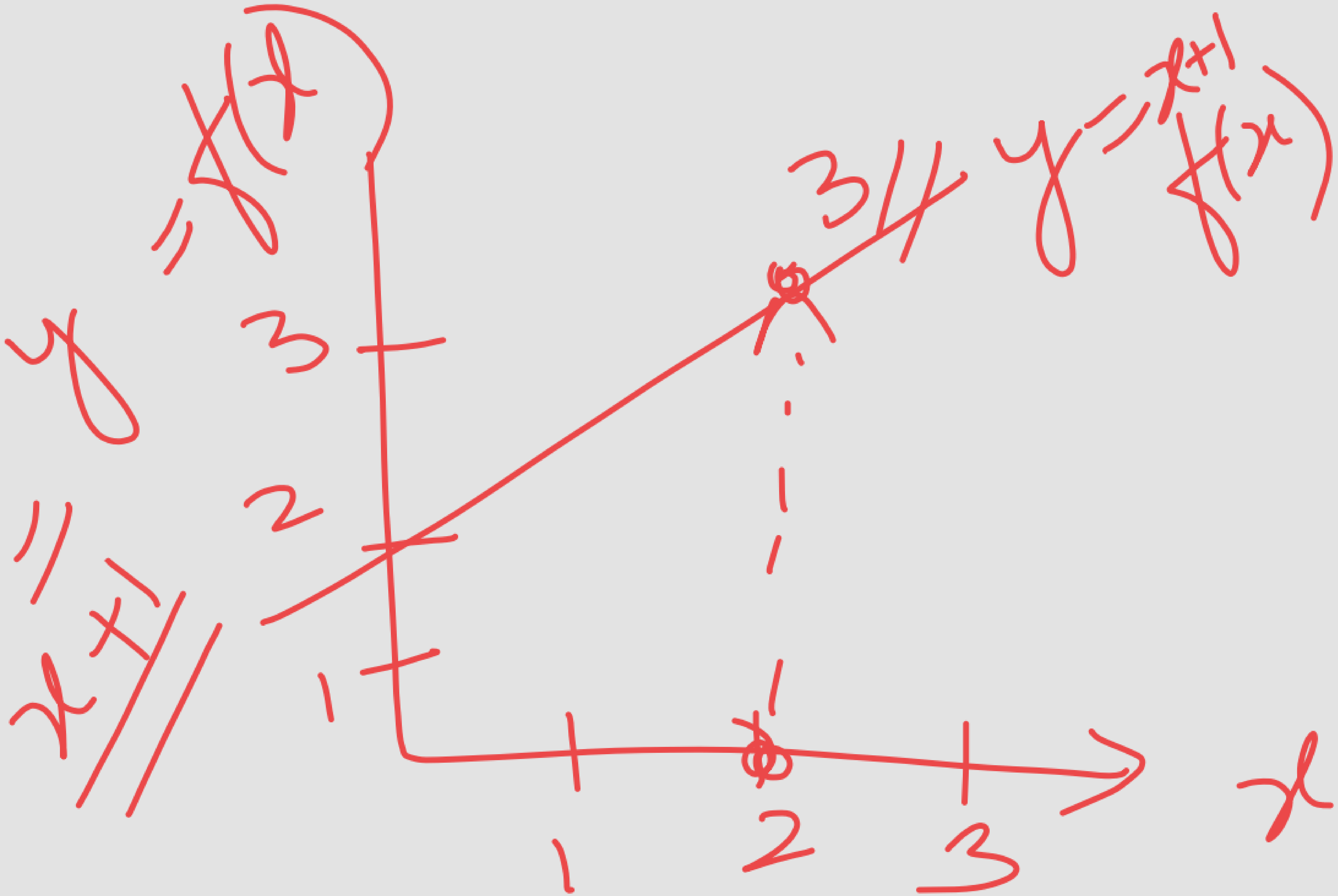
10
Test Part

5C
5H

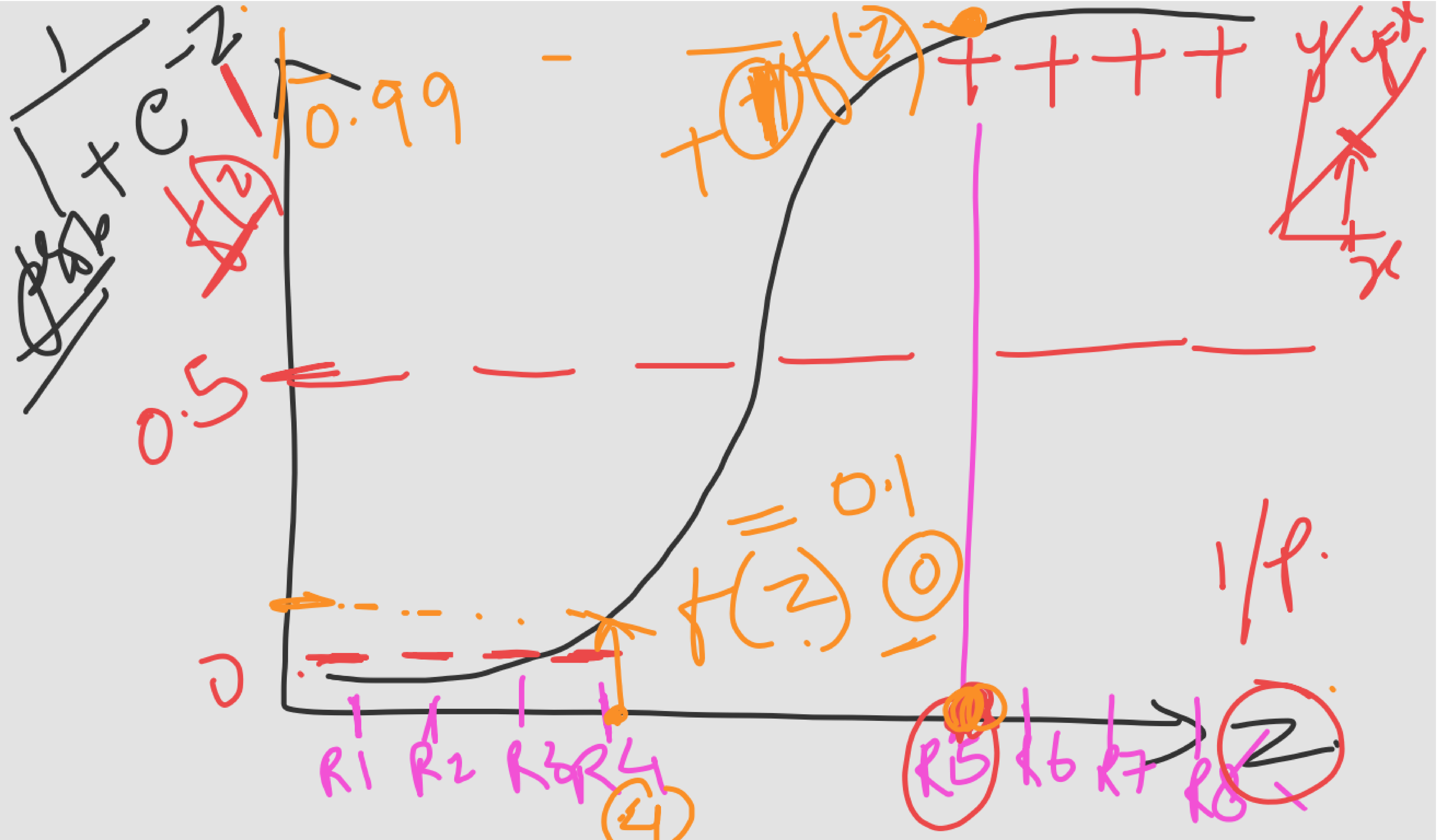
Y Y



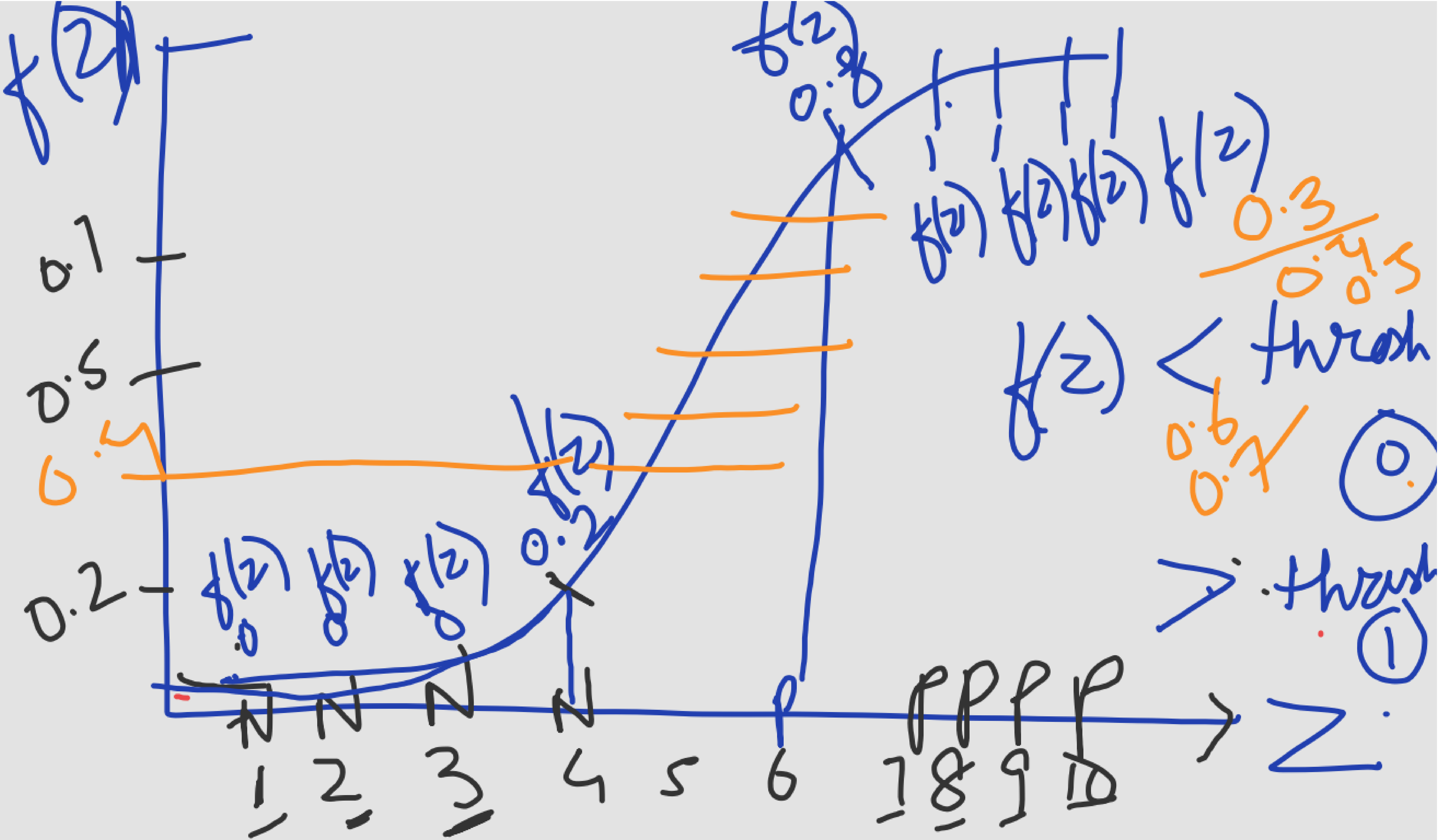
TPR
1
+
0.5
Sigmoid
0.4
−
0
FPR
ROC



y = f(x)
y = x+1
3
2
1
1
2
3
x
3
y = x+1
f(x)



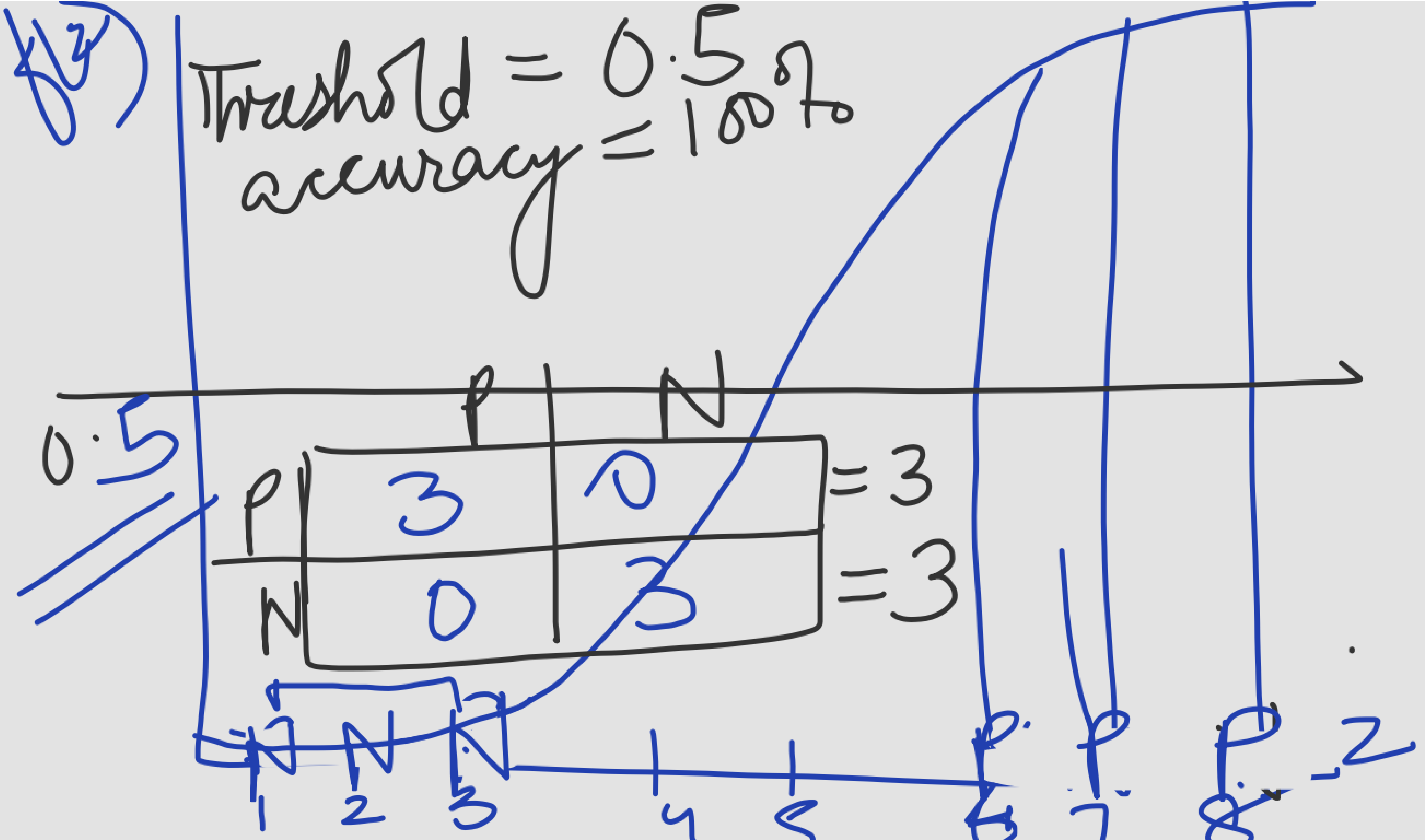
0.99
0.5
0
0.1
R1
R2
R3
R4
R5
R6
R7
R8
z
1/p
y
x



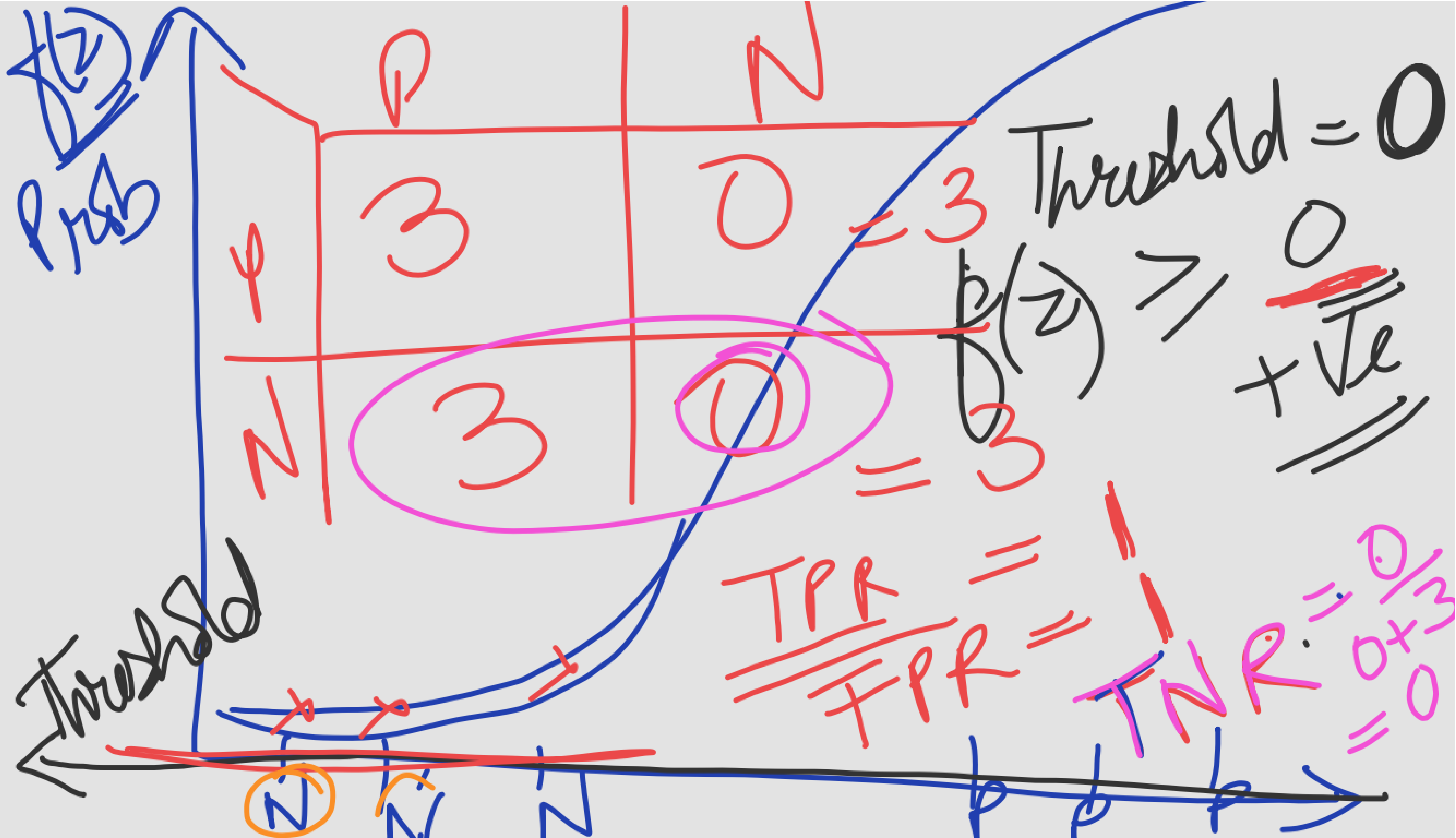
f(z)
0.1
0.5
0.4
0.2
f(z) 0
f(z) 0.2
f(z) 0.8
0.3
0.4
0.5
0.6
0.7
f(z) < thresh 0
> thresh 1
z
N N N N P P P P P
1 2 3 4 5 6 7 8 9 10

Threshold = 0.5

accuracy = 100%



	P	N	
P	3	0	= 3
N	0	3	= 3


f(z)
Prob
Threshold
P
N
P
N
3
0
3
0
= 3
= 3
Threshold = 0
f(z) ≥ 0
+ve
TPR = 1
FPR = 1
TNR = 0/(0+3) = 0

	+	−	
+	0	3	3
−	3	0	3

0.5

100%

Prob > 50% : +ve

< 50% : -ve

0.1 0.1 0.1

P P P

1 2 3

N N N

Area under curve

Receiver Operating Characteristic

AUC - ROC Curve

- To evaluate model's performance

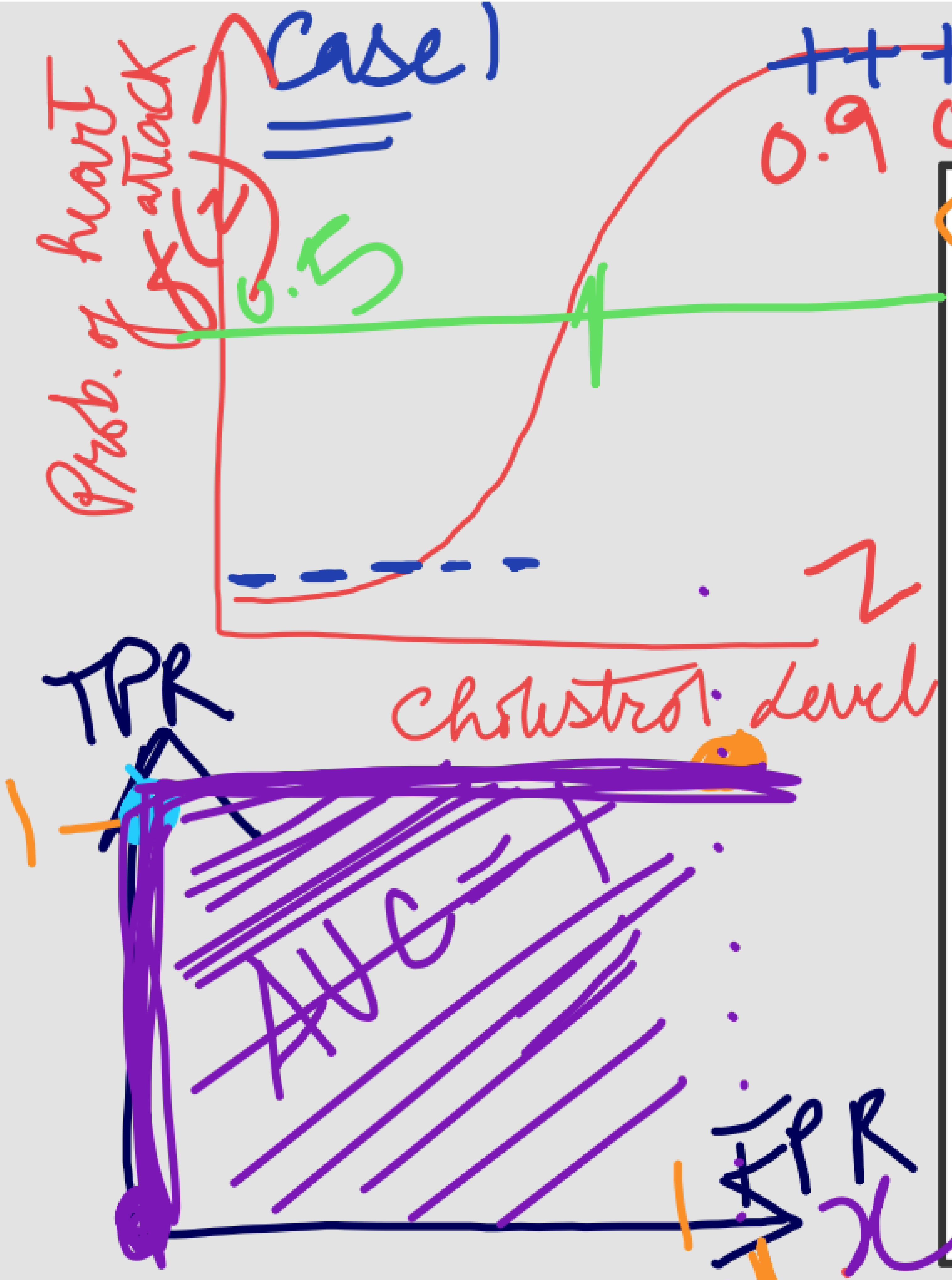
or

Performance of Classification Model at all classification thresholds.

→ Probability curve showing tradeoff b/w TPR vs FPR (ROC)

Best Value of AUC : 1

Model perfectly able to distinguish b/w + & class



Threshold 0

TPR = 1

FPR = 1

	+	-	
+	5	0	5
-	5	0	5

Threshold 0.5

TPR = 1

FPR = 0

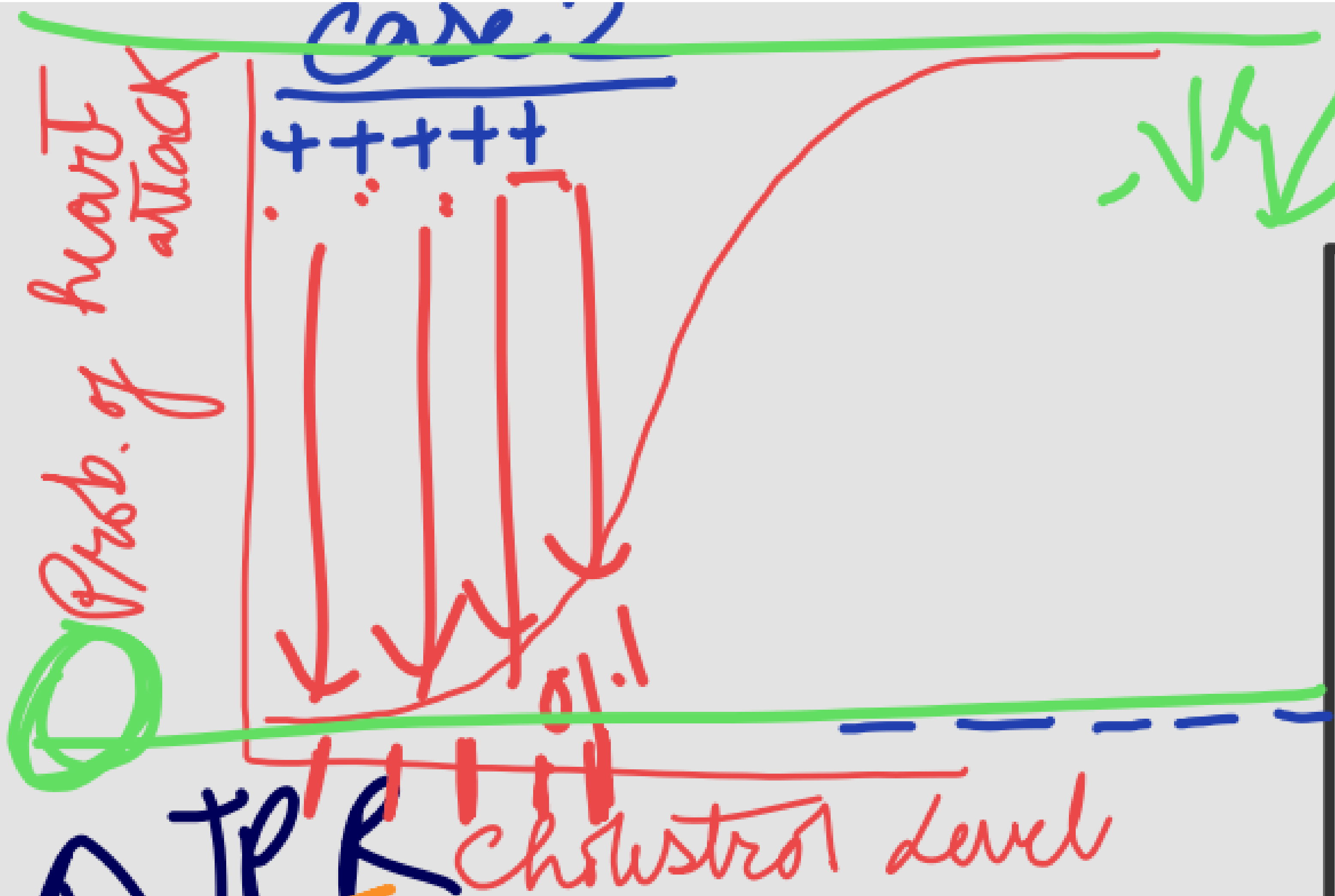
	+	-	
+	5	0	=5
-	0	5	=5

Threshold 1

TPR = 0

FPR = 0

	+	-	
+	0	5	5
-	0	5	5



Prob $\geq$ 0 → +

Threshold 0

	+	−
+	5	0
−	5	0

5, 5

TPR = 1
FPR = 1

Threshold 0.5

Prob > 0.5 +

	+	−
+	0	5
−	5	0

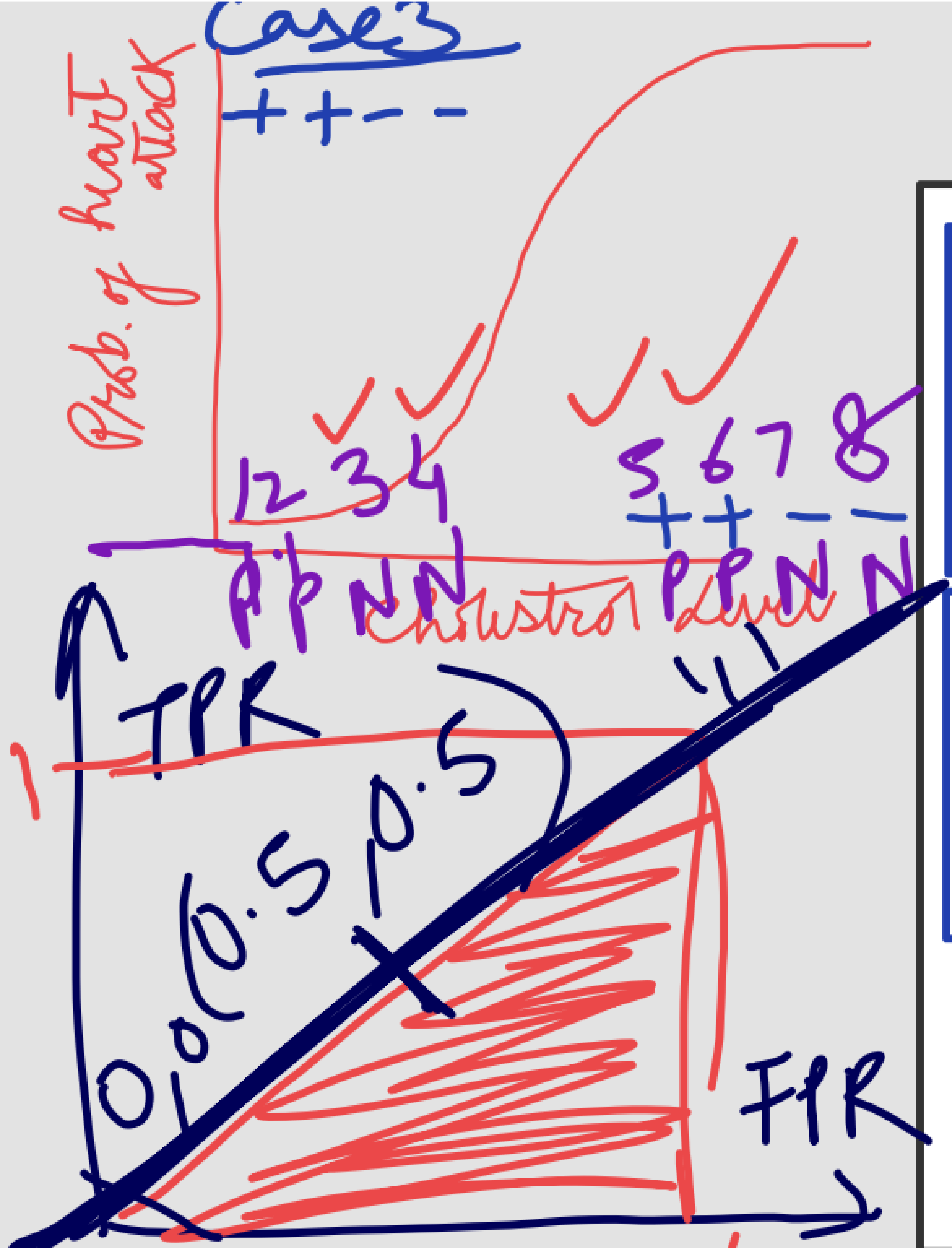
TPR = 0
FPR = 1

Threshold 1

Prob > 1 +

	+	−
+	0	5
−	5	5

TPR = 0
FPR = 0



Threshold 0

1,1

	P	N
P	4	0
N	4	0

Threshold 0.5

TNP = 0.5

FPR 0.5

Threshold 1

0,0

	P	N
P	0	4
N	0	4

for data = K-fold CrossValidat(X, Y)

[x_train, y_train, x_test, y_test]

2-5 / 1

[x train ytrain x_test, y-test

1,3-5 / 2

LR()

fit(x_train ytrain) in data

predict(x_test) y-test

acc → CM → P, R, F1

AUC value

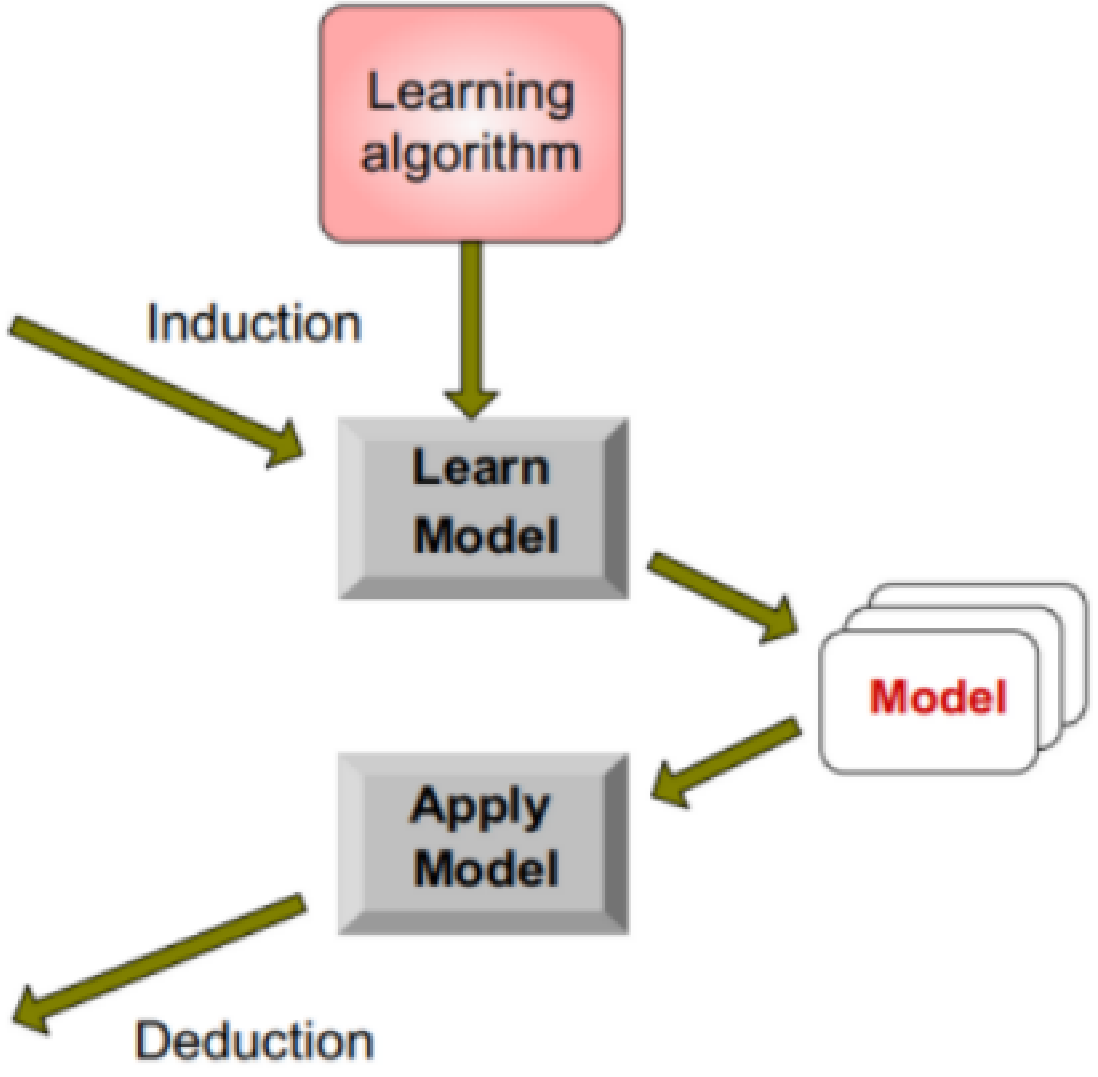
Evaluating the Performance of a Classifier

- Often useful to measure the performance of the model on the test set because such a measure provides an unbiased estimate of its generalization error.
- The accuracy or error rate computed from the test set can also be used to compare the relative performance of different classifiers on the same domain.
- Methods for evaluating classifier performance:
  1. Holdout Method
  2. Random Subsampling
  3. Cross-Validation
  4. Bootstrap

Stratified Cross Validation

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

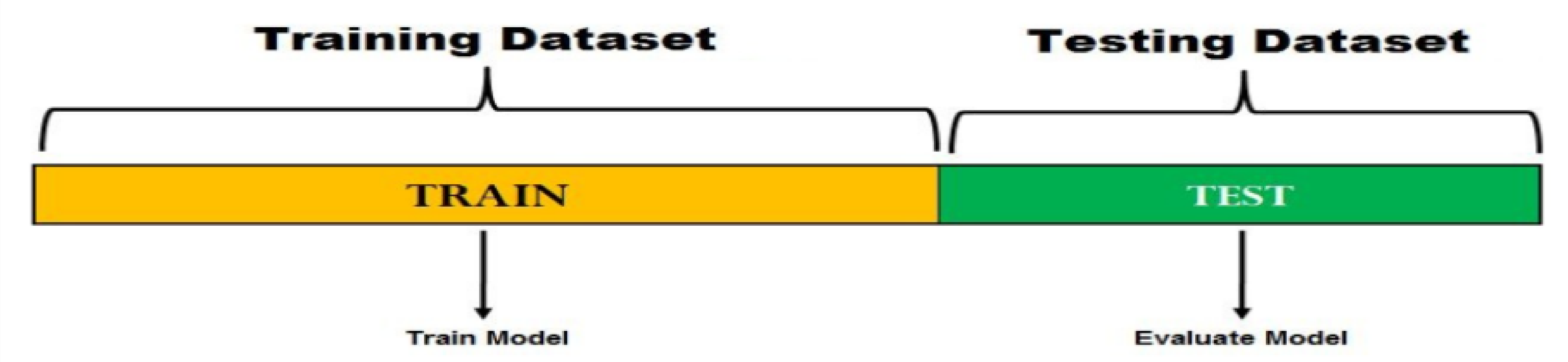


Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

1. Holdout Set

- In the holdout method, the original data with labeled examples is partitioned into two disjoint sets, called the training and the test sets, respectively.
- A classification model is then induced from the training set and its performance is evaluated on the test set.
- The proportion of data reserved for training and for testing is typically at the discretion of the analysts (e.g., 50-50 or two-thirds for training and one-third for testing).
- The accuracy of the classifier can be estimated based on the accuracy of the induced model on the test set.



Issues with the "Holdout" Method

- Separate sets for training and validation/testing

1. Fewer labeled examples are available for training because some records are held out for testing
2. Model may be highly dependent on composition of training and testing sets
   - *Small training sets will have greater variance*
   - *Small testing sets will be less reliable (will have wider confidence intervals)*

2. Random Sampling

- The holdout method can be repeated several times to improve the estimation of a classifier's performance.

Let acc_i be the model accuracy during the i^{th} iteration. The overall accuracy is given by $acc_{\text{sub}} = \sum_{i=1}^{k} acc_i/k$. Random subsampling still encounters some

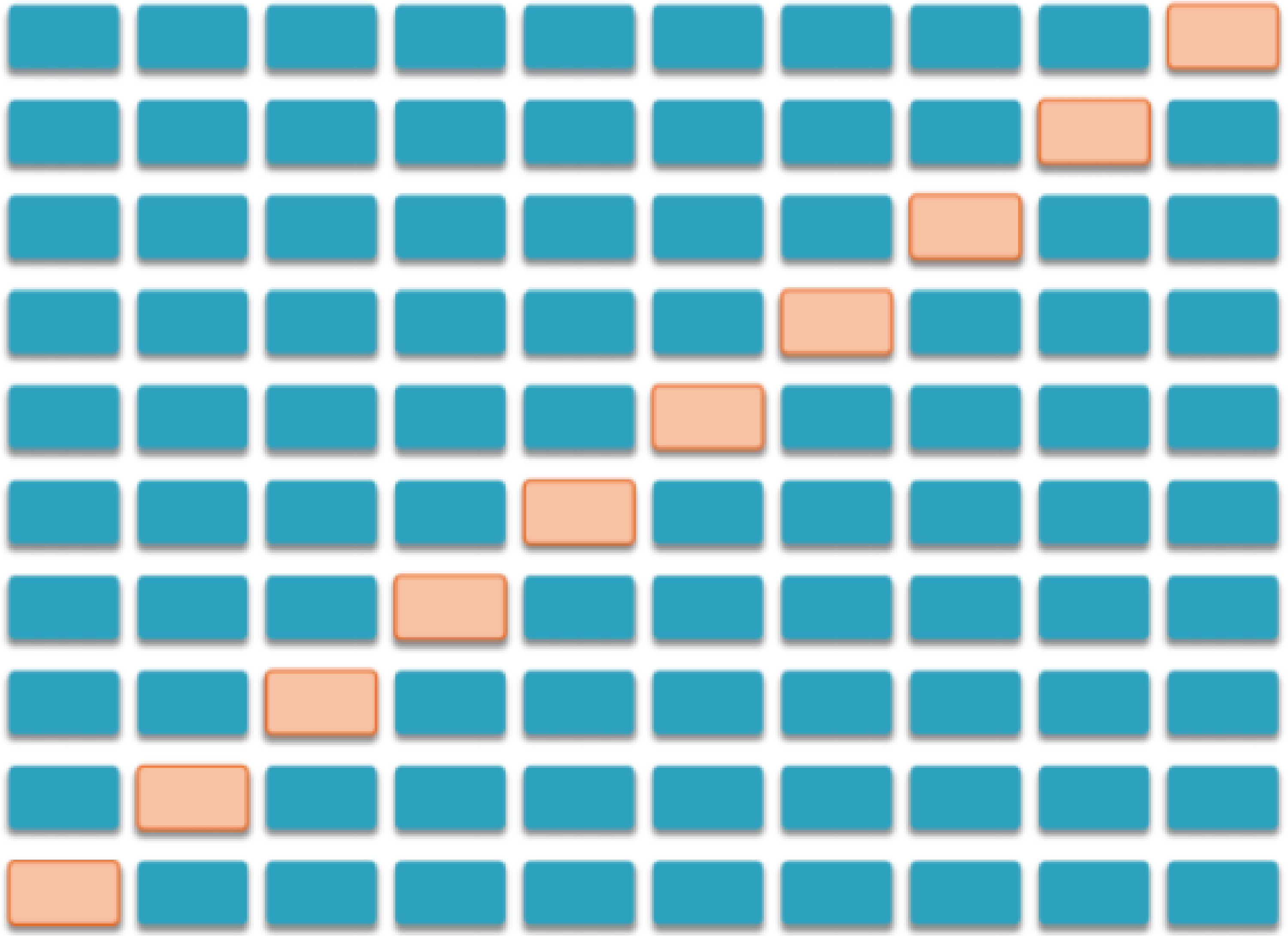
- ☹ No control over the number of times each record is used for training or testing.

3. Cross-Validation

- Each record is used the same number of times for training and exactly once for testing. To illustrate this method, suppose we partition the data into two equal-sized subsets. First, we choose one of the subsets for training and the other for testing.

- <u>k-fold Cross Validation</u>
  - k = *number of folds* (integer)
  - $k = 10$ is common
- More computationally expensive

10-fold Cross Validation

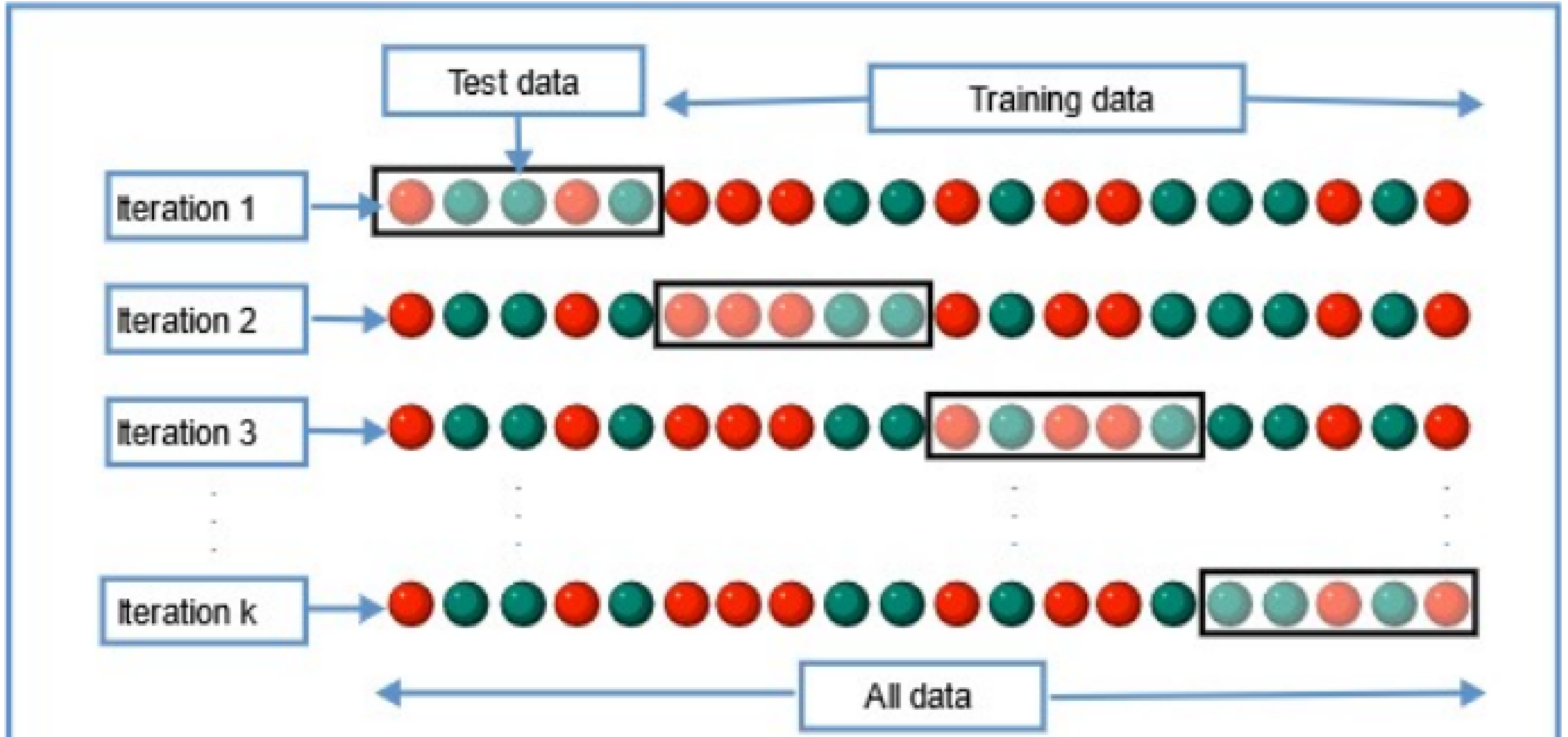


1/10th of training data

X 9 = 90% used for training

10% used for testing

- Repeat *k* times
- Average results
- Each instance will be used <u>once</u> in testing



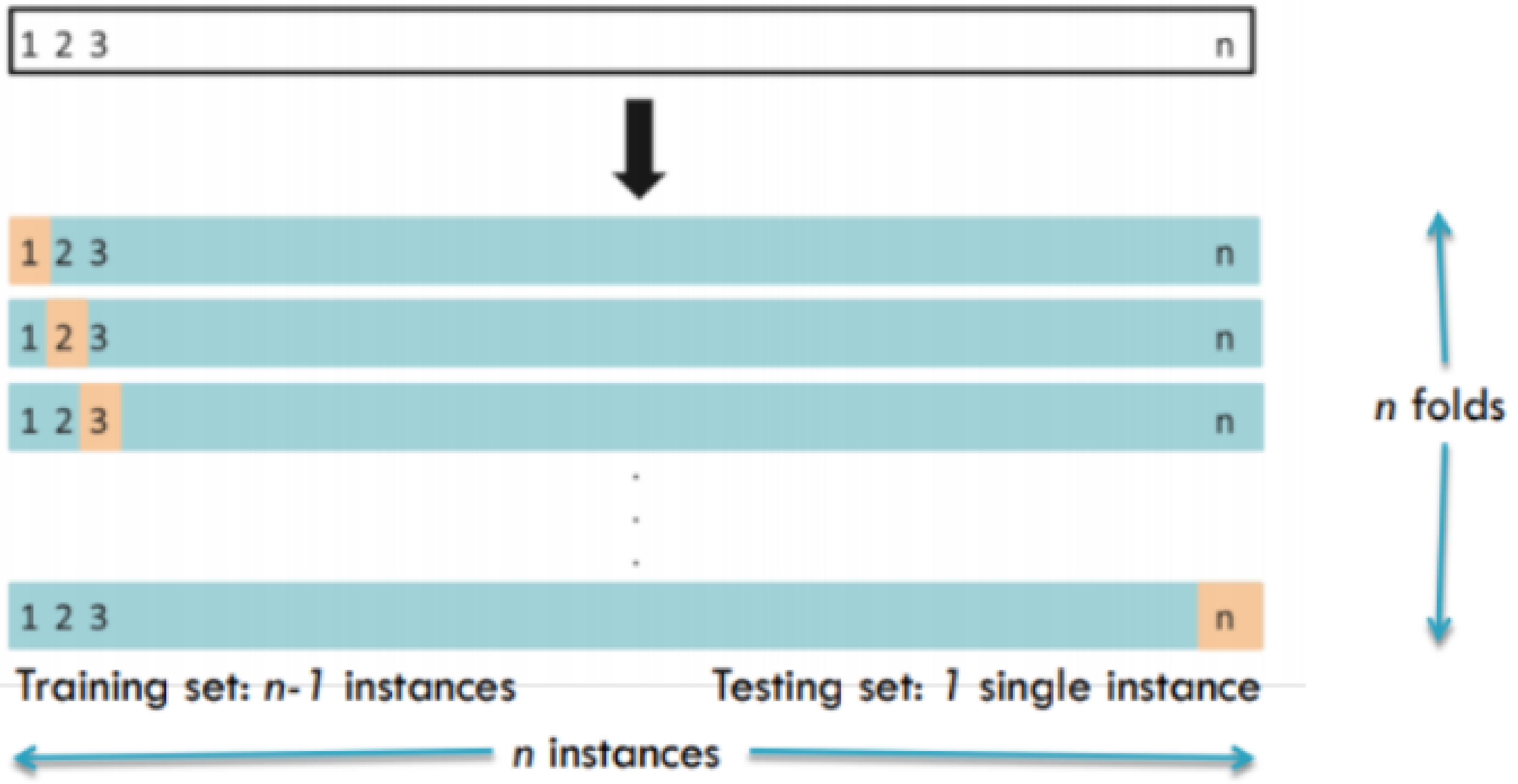
$$CV_k = \frac{1}{k}\sum_{i=1}^{k} ErrorRate_i$$

Average the error rate for each fold.

Leave-One-Out Cross-Validation

- k-fold Cross-Validation
  - k = *number of folds* (integer)
  - $k = 10$ is common
- Leave-One-Out Cross-Validation
  - *Extreme:* $k = n$, where there are n observations in training+validation set
  - Significantly more computationally expensive

Leave-One-Out Cross-Validation



5 Fold cross Validation

H	H	H	H	H	H	H	H	H	H	C	C	C	C	C
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

F1 F2 F3 F4 F5

Test Train acc_1

Train Test Train acc_2

⋮

F1 2 H 1 C F4

F2 3 4 1 2 F5

F3 5 6, 13

acc_5

Stratified K-Fold Cross-Validation

↳ Every fold should have equal fraction of all classes