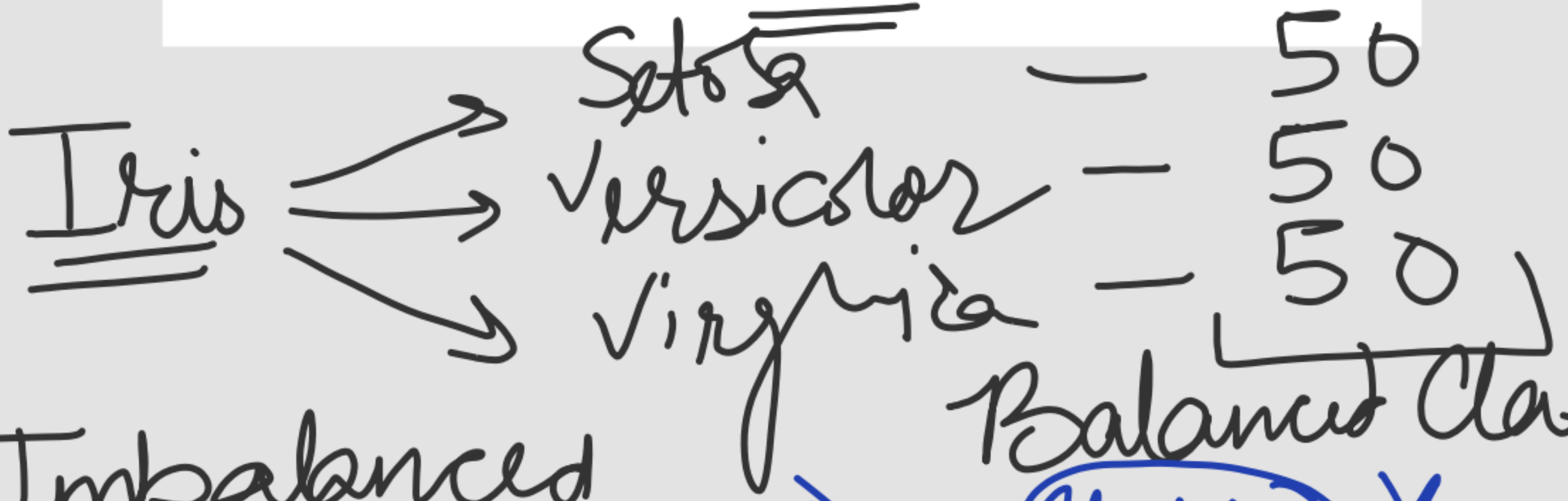
Imbalanced Class Problem



Imbalanced Class (Skewed)

① Credit Card Fraud Detection

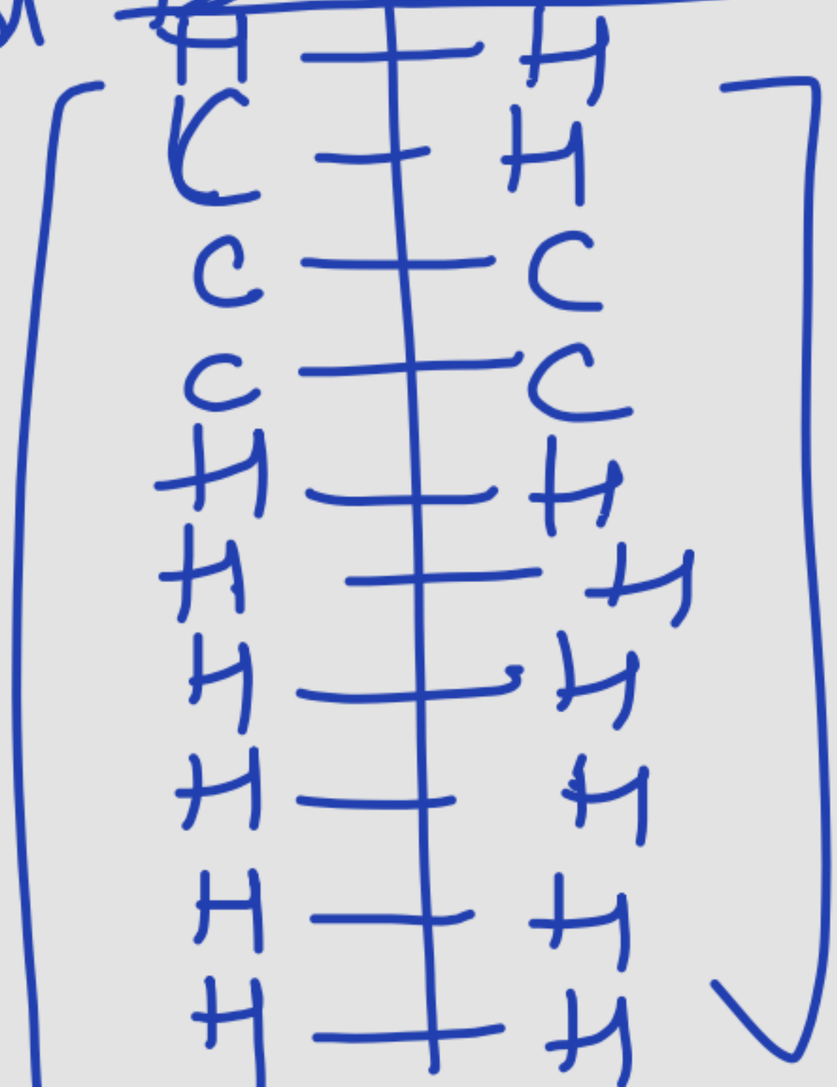
Class Y
- Transactions 100%
- Legitimate 95%
- Fraudulent 5%

Class Imbalance Problem

- Lots of classification problems where the classes are skewed (more records from one class than another)
  - Credit card fraud
  - Intrusion detection
  - Defective products in manufacturing assembly line
  - COVID-19 test results on a random sample

10 Samples

Test Dataset



Actual classes

Evaluate Performance

Predicted classes

Accuracy

90%

Challenges

- Evaluation measures such as accuracy are not well-suited for imbalanced class
- Detecting the rare class is like finding a needle in a haystack

Confusion Matrix

→ Summarizes the performance of classification algo

- Confusion Matrix:

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a	b
	Class=No	c	d

→ How many instances are correctly classified

a: TP (true positive)

b: FN (false negative)

c: FP (false positive)

d: TN (true negative)

Confusion Matrix

- Confusion Matrix:

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a	b
	Class=No	c	d

a: TP (true positive)

b: FN (false negative)

c: FP (false positive)

d: TN (true negative)

Positive Negative

Positive TP FN

Negative FP TN

Truely classified as +ve

Falsely classified as -ve

Falsely classified as +ve

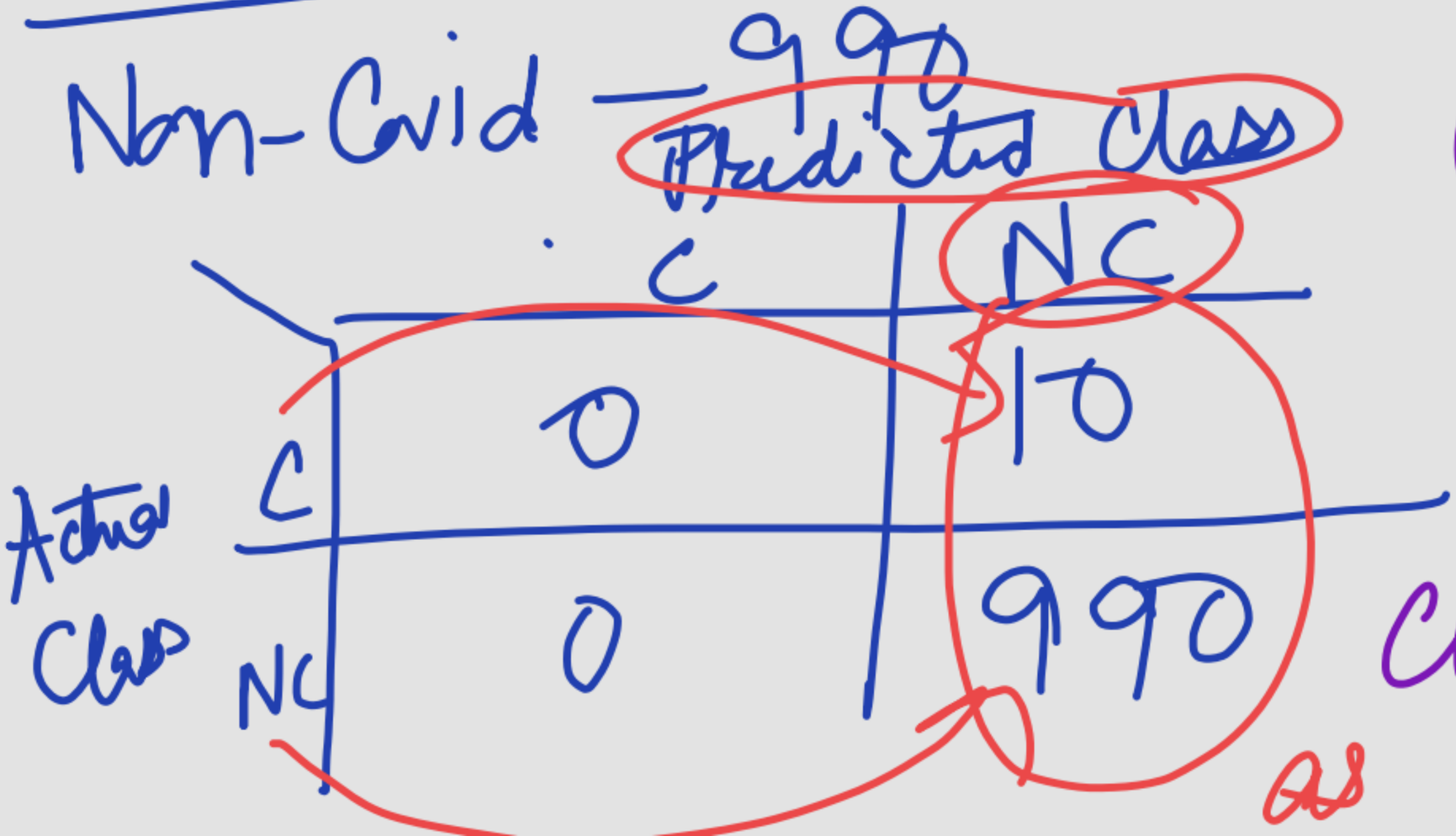
Truely classified as -ve


Predicted Class
Covid
Non-Covid
Yes / 1 / + / Positive
No / 0 / - / Negative
Actual Class
Yes
Covid
+
No
Non-Covid
-
1
3
1
3
Confusion Matrix
Accuracy = (1+3)/8 = 50%
Y
Y
Actual class
R1 C = C
R2 C = N
R3 C = N
R4 C = N
R5 N = C
R6 N = N
R7 N = N
R8 N = N

What is the problem with accuracy for imbalanced classes?

Covid — 10

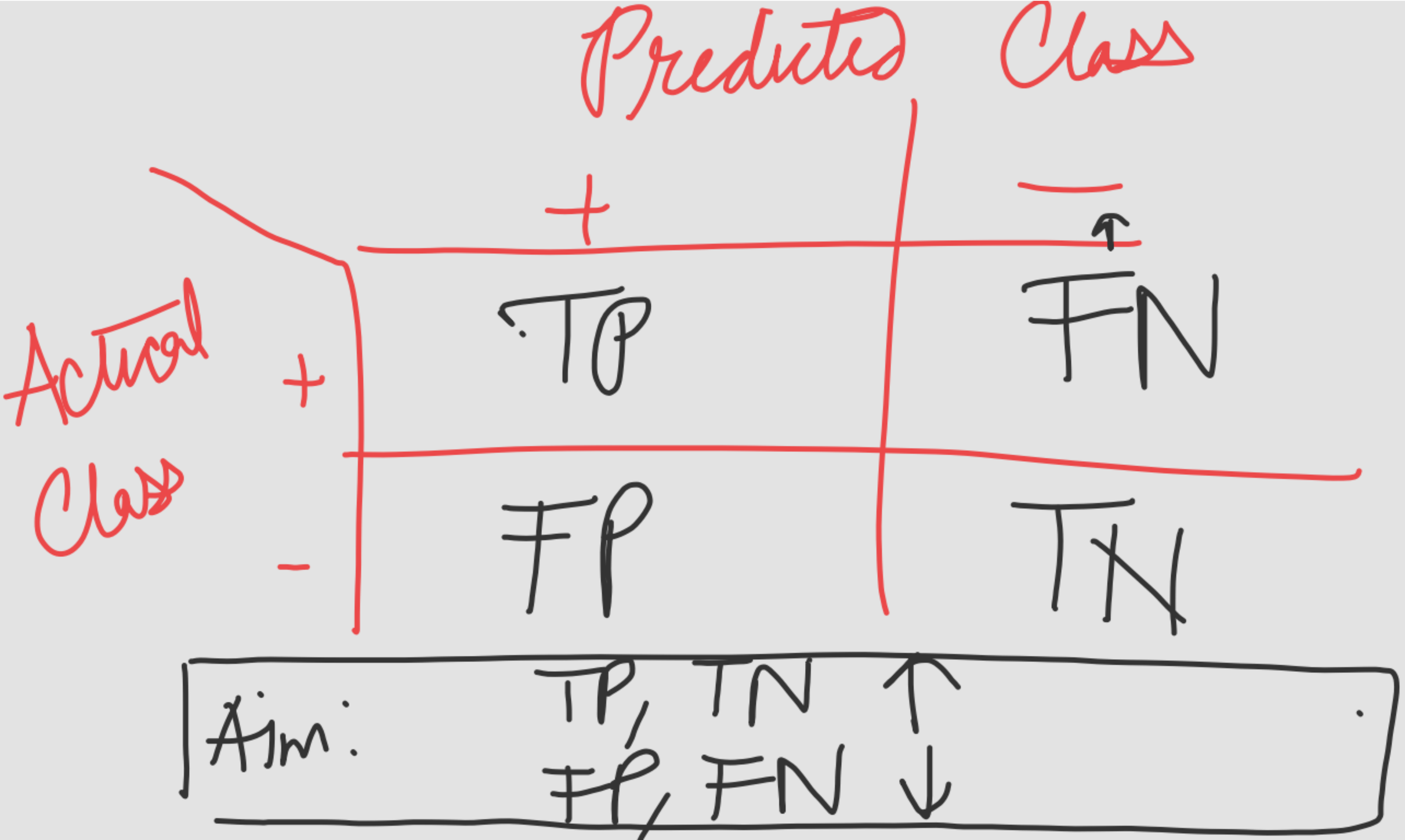
Non-Covid — 990



Accuracy =

990 / 1000 = 99%

Classify everything as NC.


Predicted Class
+
−
Actual Class
+
−
TP
FN
FP
TN
Aim:
TP, TN ↑
FP, FN ↓

Accuracy

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

- Most widely-used metric:

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

Problem with Accuracy

- Consider a 2-class problem
  - Number of Class NO examples = 990
  - Number of Class YES examples = 10

Problem with Accuracy

- Consider a 2-class problem
  - Number of Class NO examples = 990
  - Number of Class YES examples = 10

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	0 (TP)	10 (FN)
	Class=No	0 (FP)	990 (TN)

True Positive Rate (TPR)
Out of all +ve samples how many r correctly classified

$$TPR = \frac{TP}{TP+FN}$$

False Negative Rate (FNR)
Out of all +ve Samples how many r incorrectly classified

$$FNR = \frac{FN}{TP+FN}$$

$$FPR = \frac{FP}{FP+TN}$$

$$TNR = \frac{TN}{FP+TN}$$

Problem with Accuracy

- Consider a 2-class problem
  - Number of Class NO examples = 990
  - Number of Class YES examples = 10

- If a model predicts everything to be class NO, accuracy is 990/1000 = 99 %
  - This is misleading because the model does not detect any class YES example
  - Detecting the rare class is usually more interesting (e.g., frauds, intrusions, defects, etc)

Which model is better?

Accuracy

A

		PREDICTED	
		Class=Yes	Class=No
ACTUAL	Class=Yes	0	10
	Class=No	0	990

99%

B

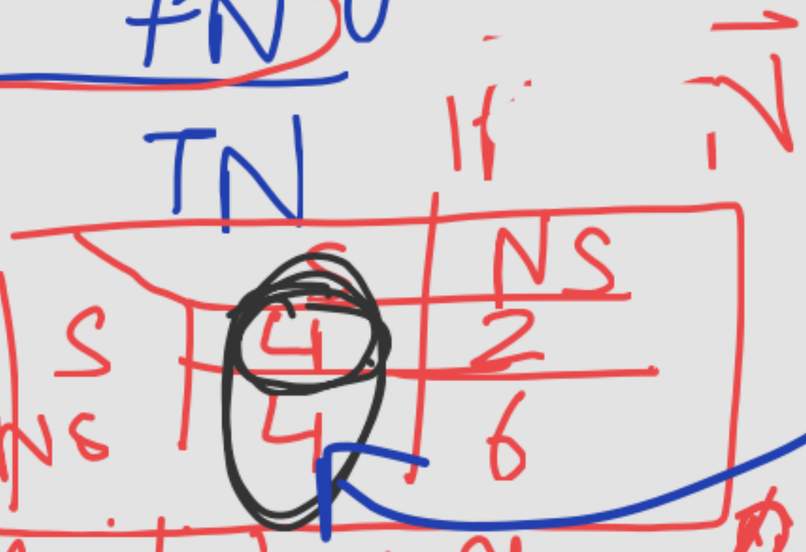
		PREDICTED	
		Class=Yes	Class=No
ACTUAL	Class=Yes	10	0
	Class=No	90	900

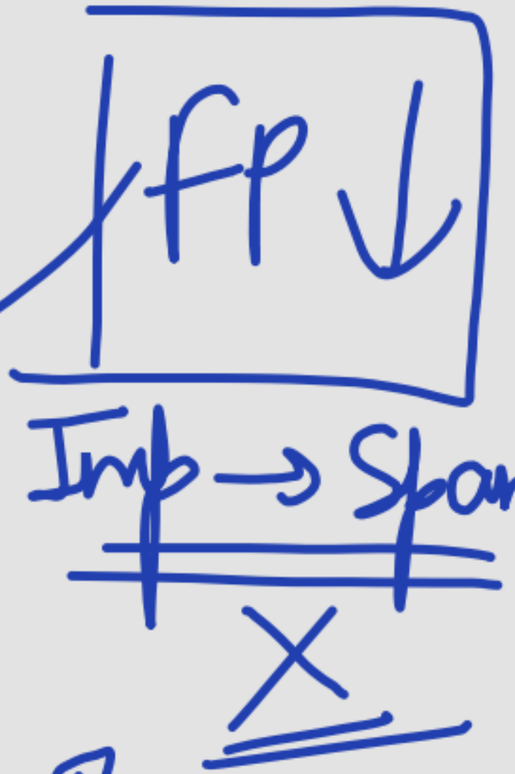
91%

Precision

$$\frac{TP}{TP+FP}$$

Fraction of records which are actually +ve out of total records which are classified or predicted as +ve.

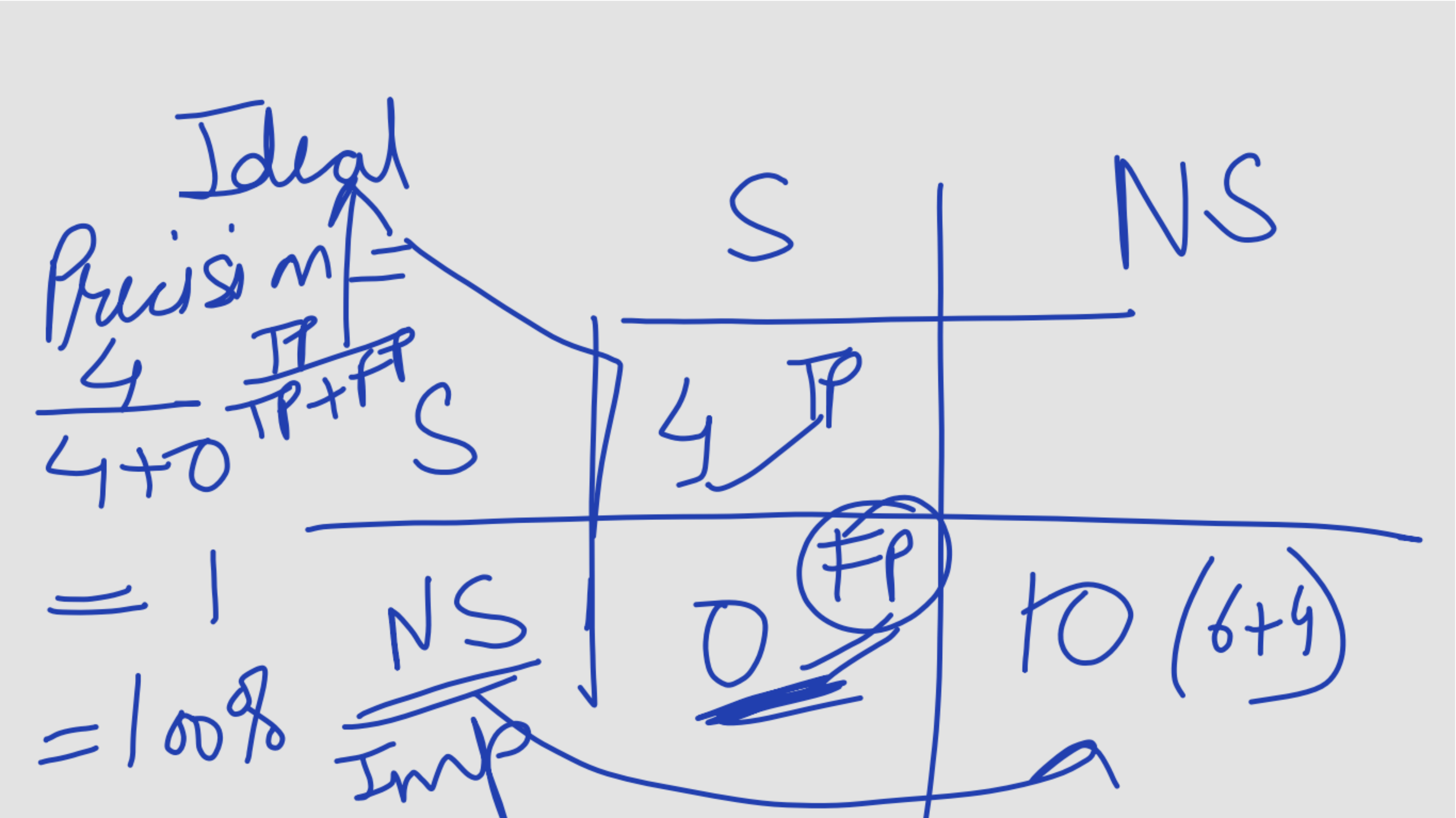




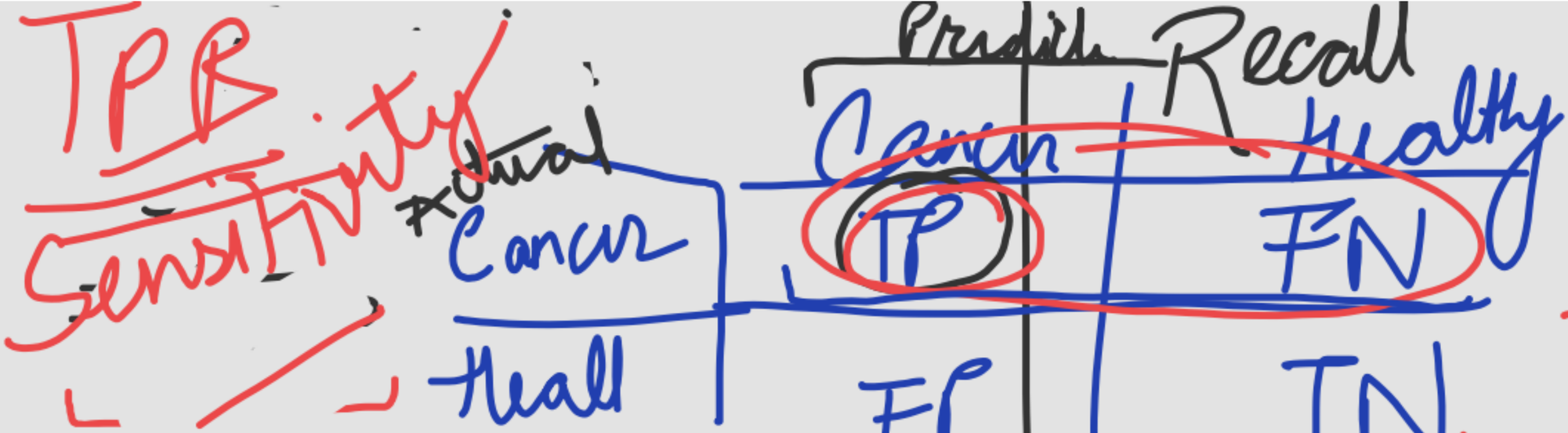
Don't want to miss Imp. mail

Precision $= \frac{TP}{TP+FP} = \frac{4}{4+4} = 50\%$

High Precision


Ideal
Precision =
4 / 4+0
TP / TP+FP
S
NS
S
4
TP
FP
NS
0
10 (6+4)
= 1
= 100%
Imp

TPR
Sensitivity



$$\frac{TP}{TP+FN}$$

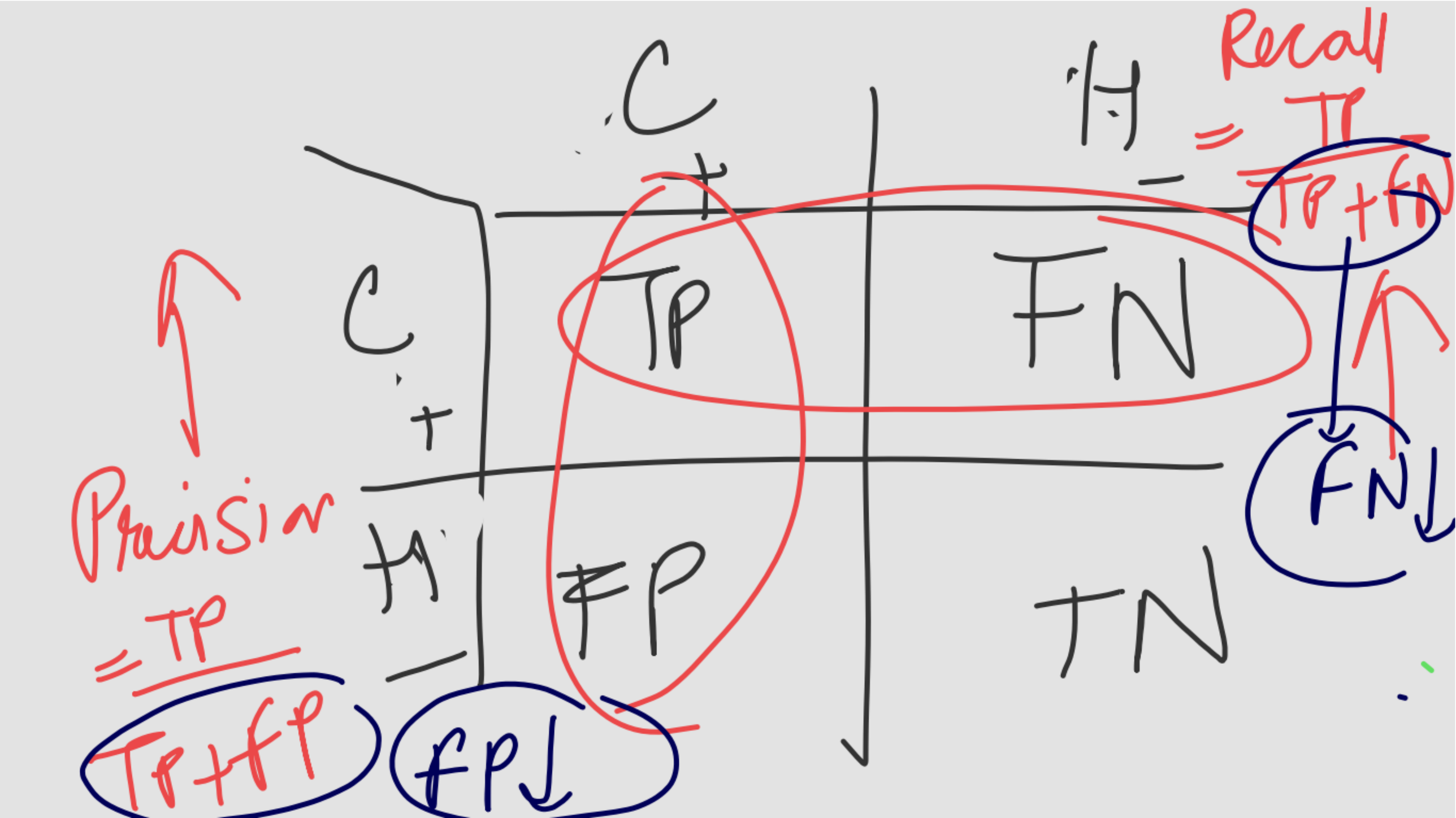
1 Out of all total +ve records, how many r correctly classified as +ve

Recall → 0

10 Cancers → classified as Healthy

If a person has cancer
→ & model says ∆t⁰
↳ there is a great risk

Recall ↑


C
+
H
−
Recall
=
TP
TP+FN
C
+
TP
FN
Precision
=
TP
TP+FP
H
−
FP
TN
FN
FP

For following confusion metrics find accuracy, Precision & recall of

(A) Model 1

	+	−	
+	1	9	= 10
−	0	990	= 990

(B) Model 2

	+	−	
+	10	0	10
−	90	900	990

	Acc	Pr	Re
A	99.1%	$\frac{1}{1+0}$ = 100%	$\frac{1}{1+9}$ = 10%
B	91%	$\frac{10}{10+90}$ = 10%	$\frac{10}{10+0}$ = 100%

High Pr & High Recall → 100%

	+	−	
+	10	0	
−	0	990	

100%

TP ↑ FP ↓

FN ↓

High precision & low recall

low precision & high recall

	Precisi	Recall	Measur
Model	1.00	1.0	$\frac{1.0+1.0}{2}$ = .55
2.	.20	1.00	.60
3	1.00	1.00	1.00

As per arithmetic mean go for model 3

Let's break till 11:56 pm

Ques How to compare precision & recall?

	Precision (P)	Recall (R)	Average $\left(\frac{P+R}{2}\right)$
Model 1	0.5	0.4	0.45 ✓
✗ Model 2	0.7	0.1	0.4
✓ Model 3	0.02	1.0	0.51 (Highest for model 3)

☹ Not a good classifier
Predict y=1 all the time

CM

	+	−
+	20	0
−	980	

Let's break till 11:56 pm

Ques How to Compare precision & recall?

	Precision (P)	Recall (R)	Average $\left(\frac{P+R}{2}\right)$
Model 1	0.5	0.4	0.45
Model 2	0.7	0.1	0.4
Model 3	0.02	1.0	0.51

(Highest for model 3)

☹ Not a good classifier
Predict y=1 all the time

0 ... 1 → 0.5+ → 0

F-Score

Geometric Mean

Harmonic Mean

AM: 0.44, 0.175, 0.03

$\frac{2}{\frac{1}{P}+\frac{1}{R}}$

$\frac{2PR}{P+R}$

Harmonic Mean (F_1 Score)

$$= \frac{2}{\frac{1}{p} + \frac{1}{r}} = \frac{2pr}{p + r}$$

$$p = \frac{TP}{TP + FP}$$

$$R = \frac{TP}{TP + FN}$$

$$= \frac{2TP}{2TP + FP + FN}$$

Alternative Measures

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a	b
	Class=No	c	d

$$\text{Precision (p)} = \frac{a}{a+c}$$

$$\text{Recall (r)} = \frac{a}{a+b}$$

$$\text{F - measure (F)} = \frac{2rp}{r+p} = \frac{2a}{2a+b+c}$$

Measures of Classification Performance

	PREDICTED CLASS		
ACTUAL CLASS		Yes	No
	Yes	TP	FN
	No	FP	TN

α is the probability that we reject the null hypothesis when it is true. This is a Type I error or a false positive (FP).

β is the probability that we accept the null hypothesis when it is false. This is a Type II error or a false negative (FN).

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN}$$

$$ErrorRate = 1 - accuracy$$

$$Precision = Positive\ Predictive\ Value = \frac{TP}{TP + FP}$$

$$Recall = Sensitivity = TP\ Rate = \frac{TP}{TP + FN}$$

$$Specificity = TN\ Rate = \frac{TN}{TN + FP}$$

$$FP\ Rate = \alpha = \frac{FP}{TN + FP} = 1 - specificity$$

$$FN\ Rate = \beta = \frac{FN}{FN + TP} = 1 - sensitivity$$

$$Power = sensitivity = 1 - \beta$$

Alternative Measures

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	10	40
	Class=No	10	40

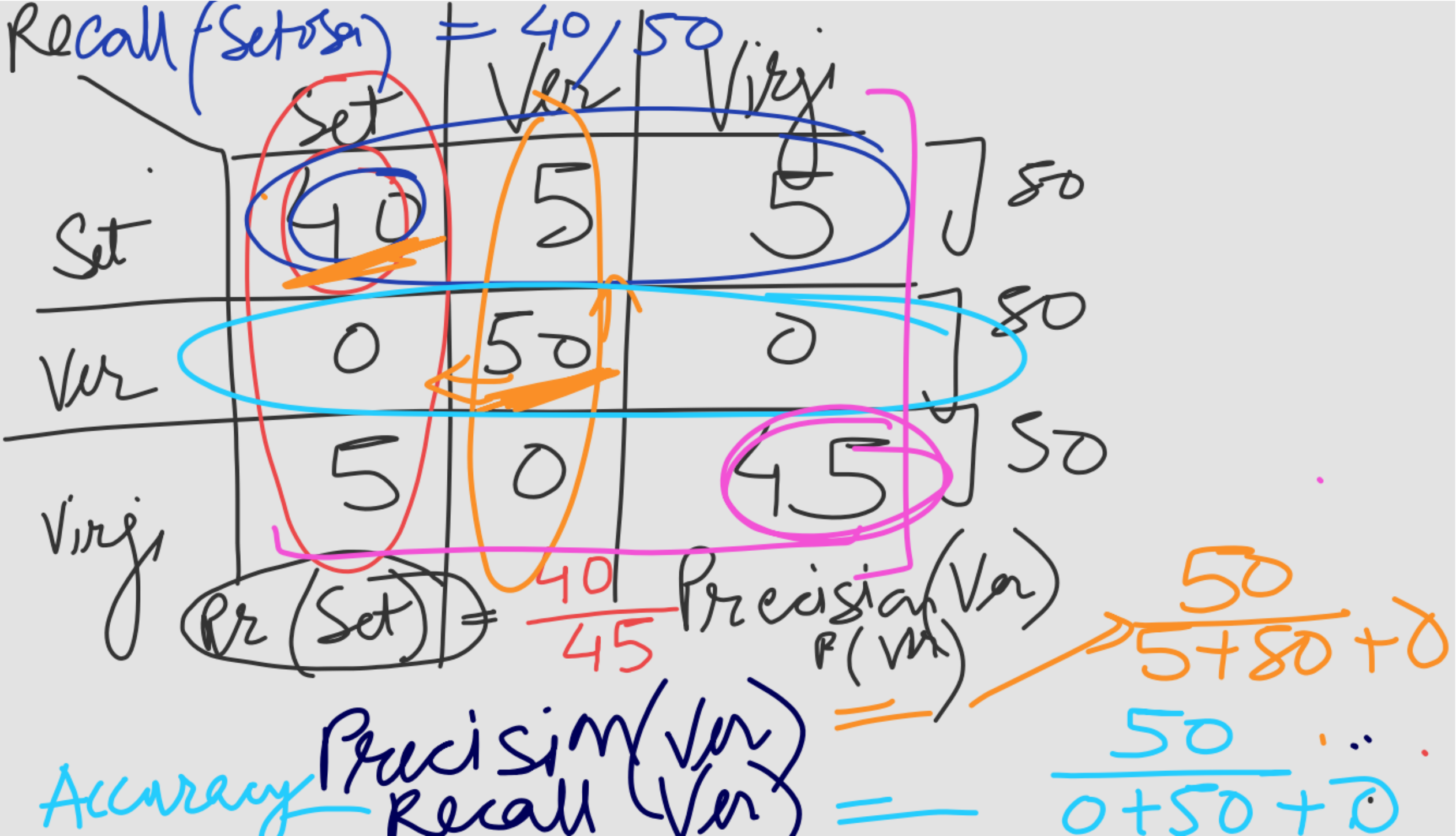
Precision (p) = 0.5
TPR = Recall (r) = 0.2
FPR = 0.2
F – measure = 0.28

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	25	25
	Class=No	25	25

Precision (p) = 0.5
TPR = Recall (r) = 0.5
FPR = 0.5
F – measure = 0.5

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	40	10
	Class=No	40	10

Precision (p) = 0.5
TPR = Recall (r) = 0.8
FPR = 0.8
F – measure = 0.61


Recall (Setosa) = 40/50
Set
Ver
Virg
Set
Ver
Virj
40
5
5
50
0
50
0
50
5
0
45
50
Pr (Set) = 40/45
Precision (Ver)
P(Ver)
50/5+80+0
Accuracy
Precision (Ver)
Recall (Ver)
50/0+50+0

ROC (Receiver Operating Characteristic)

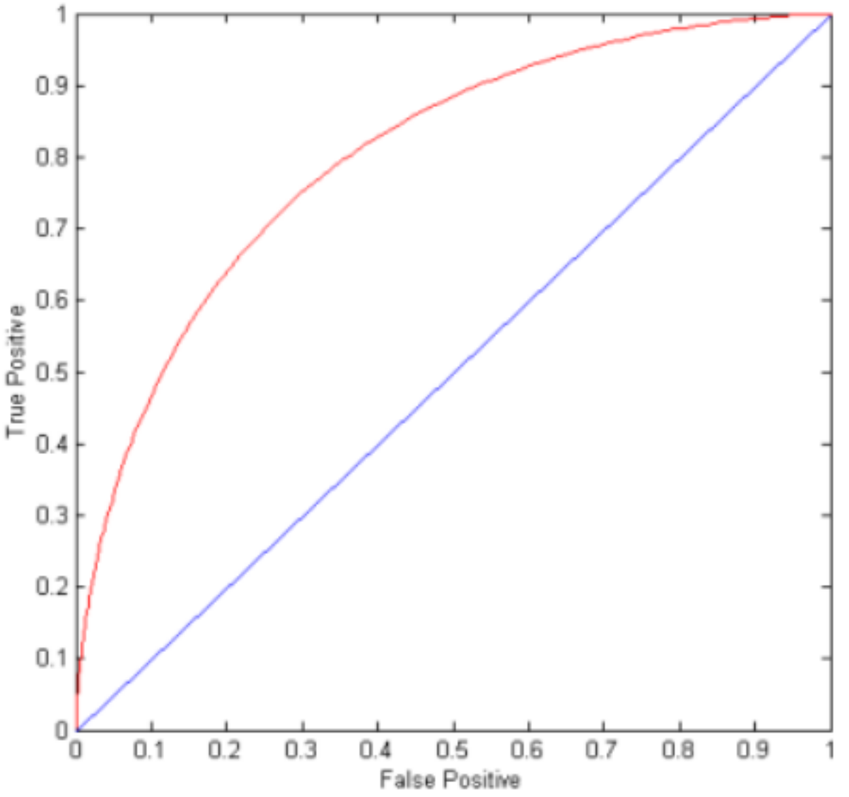
- A graphical approach for displaying trade-off between detection rate and false alarm rate
- Developed in 1950s for signal detection theory to analyze noisy signals
- ROC curve plots TPR against FPR
  - Performance of a model represented as a point in an ROC curve
  - Changing the threshold parameter of classifier changes the location of the point

ROC Curve

(TPR,FPR):

- (0,0): declare everything to be negative class
- (1,1): declare everything to be positive class
- (1,0): ideal

- Diagonal line:
  - Random guessing
  - Below diagonal line:
    - prediction is opposite of the true class



ROC (Receiver Operating Characteristic)

- To draw ROC curve, classifier must produce continuous-valued output
  - Outputs are used to rank test records, from the most likely positive class record to the least likely positive class record

- Many classifiers produce only discrete outputs (i.e., predicted class)
  - How to get continuous-valued outputs?
    - Decision trees, rule-based classifiers, neural networks, Bayesian classifiers, k-nearest neighbors, SVM

Evaluating the Performance of a Classifier

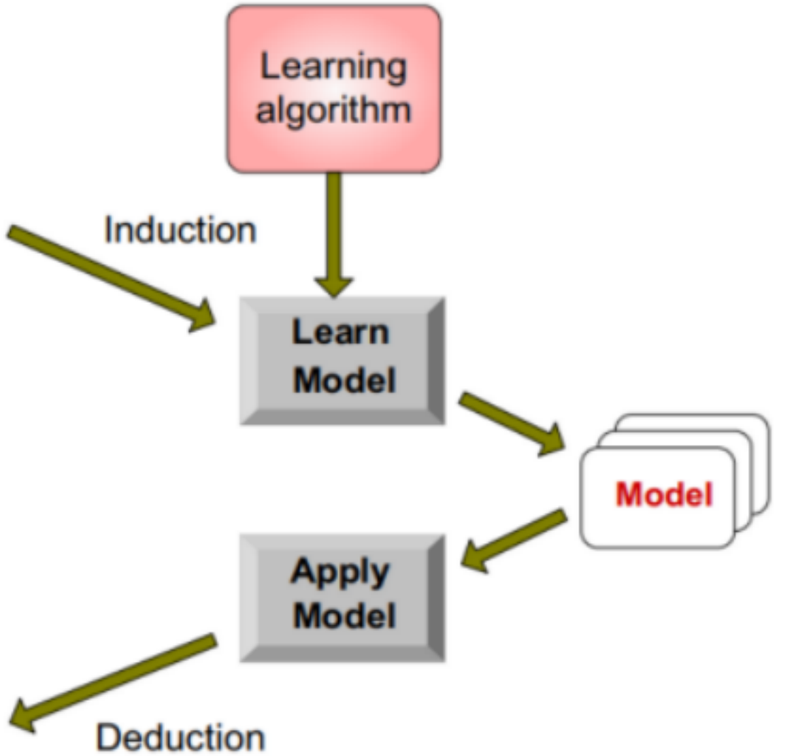
- Often useful to measure the performance of the model on the test set because such a measure provides an unbiased estimate of its generalization error.
- The accuracy or error rate computed from the test set can also be used to compare the relative performance of different classifiers on the same domain.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

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

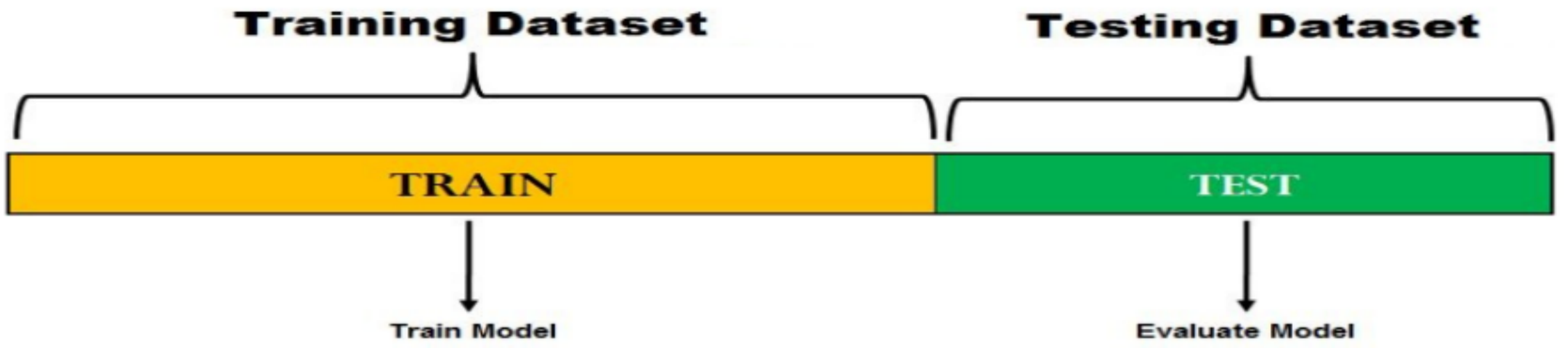
Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

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

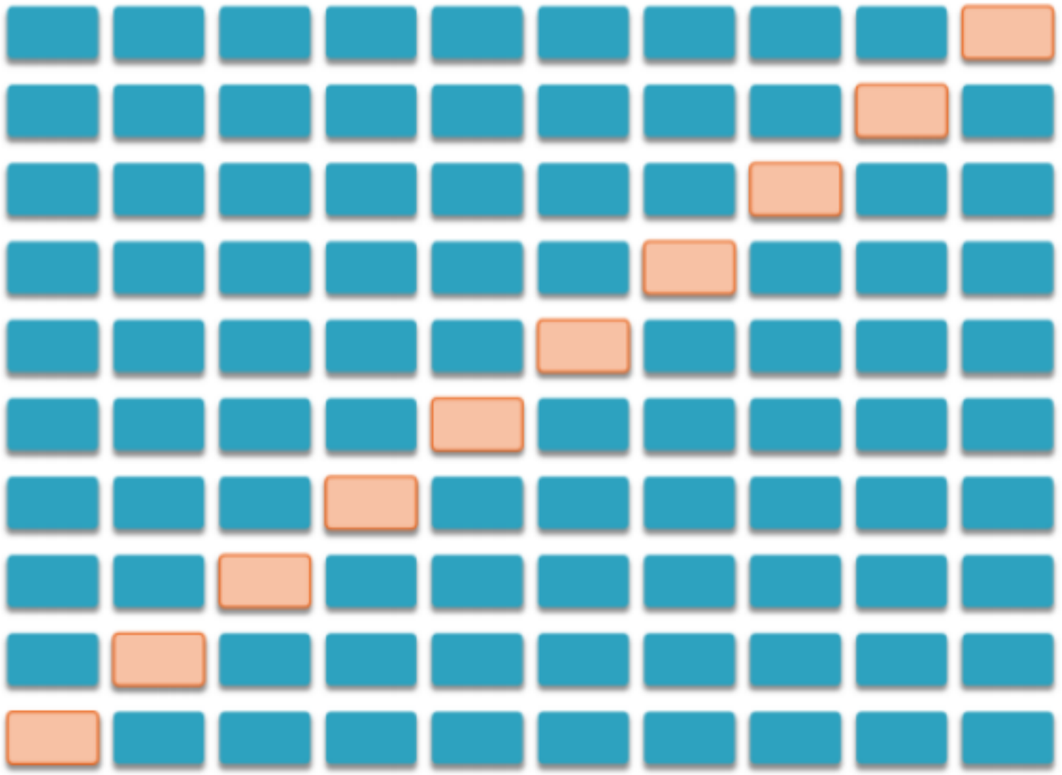
- ☹ No control over the number of times each record is used for training or testing.

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.

  - k-fold Cross Validation
    - *k = number of folds* (integer)
    - *k = 10* is common
  - More computationally expensive

10-fold Cross Validation

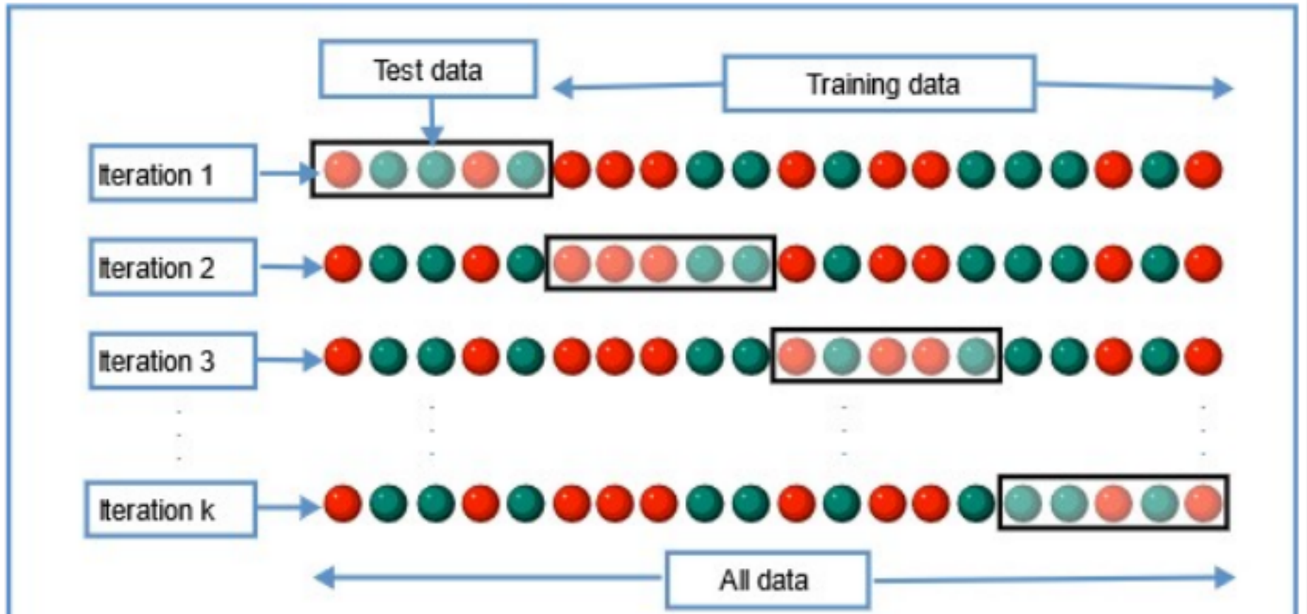


1/10th of training data

X 9 = 90% used for training

10% used for testing

- Repeat *k* times
- Average results
- Each instance will be used once in testing



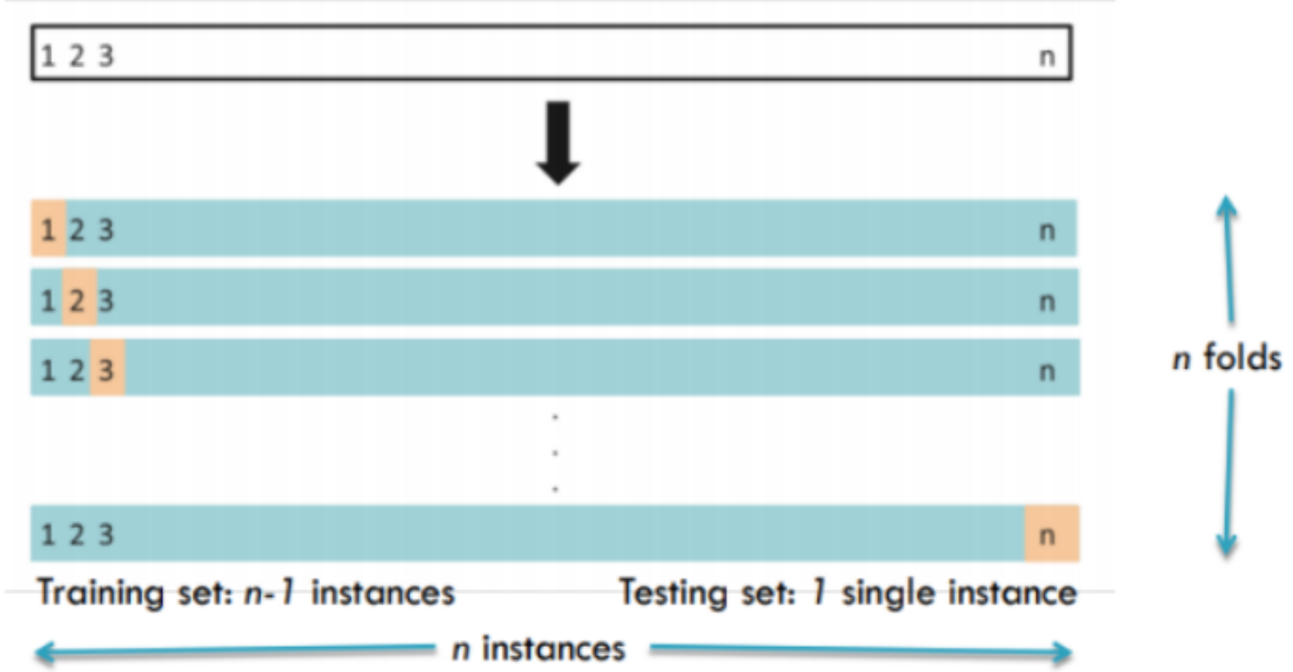
$$CV_k = \frac{1}{k}\sum_{i=1}^{k} ErrorRate_i$$

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



Stratified K-Fold Cross-Validation