Classification

Certificate Course on Artificial Intelligence & Deep Learning
by IIT Roorkee

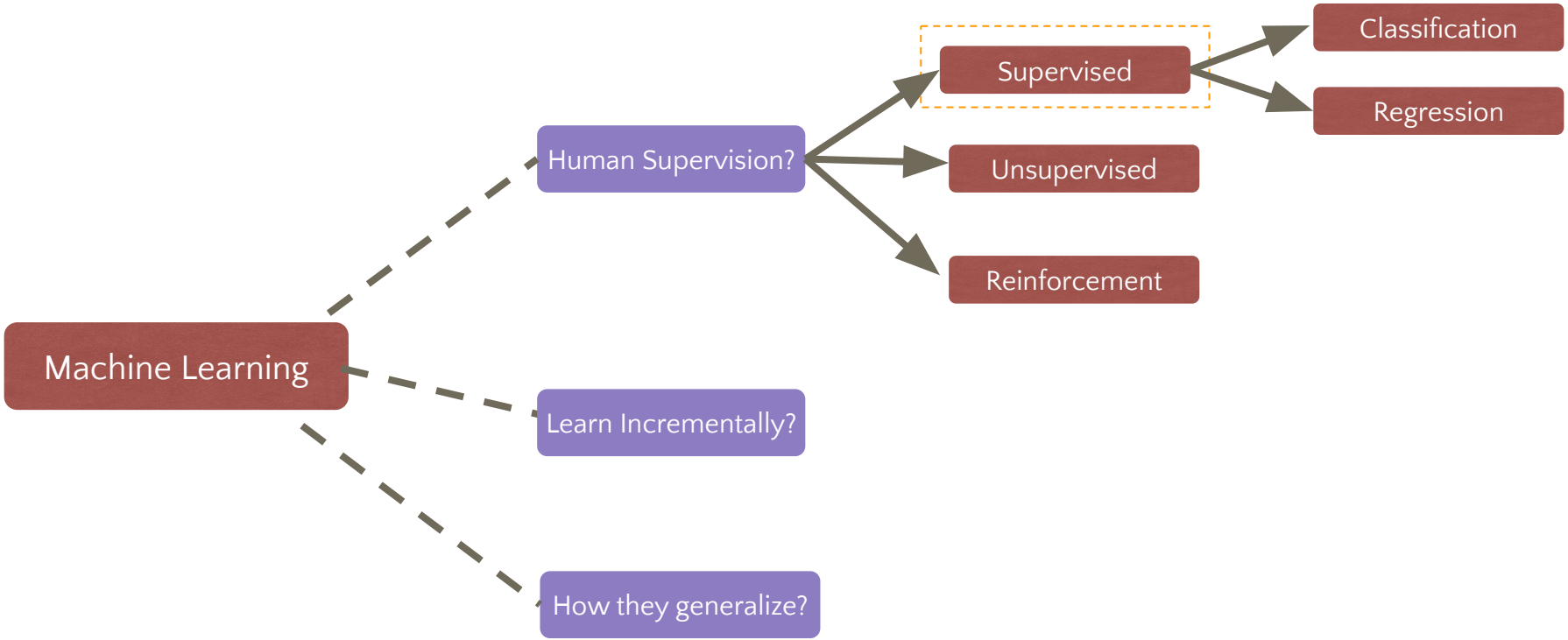
Agenda

- Introduction
- Build Model with MNIST Dataset
- Performance Measures
  - Cross Validation
  - Confusion Matrix
  - Precision & Recall
  - Precision/Recall Trade-off
  - ROC Curve
- Type of Classification
  - Multi Class
  - Multi Label
  - Multi Output

Agenda

- Ask questions
- Teaching Assistants will be there to help you out
- Post your questions on forum
  - discuss.cloudxlab.com

What is Classification ?



CLOUD x LAB

What is Classification?: Definition

Classification is the task of

- Identifying to which category a new observation belongs
- Since labeled data is available (value of output variable), it's a supervised classification.

Cat

Not Cat

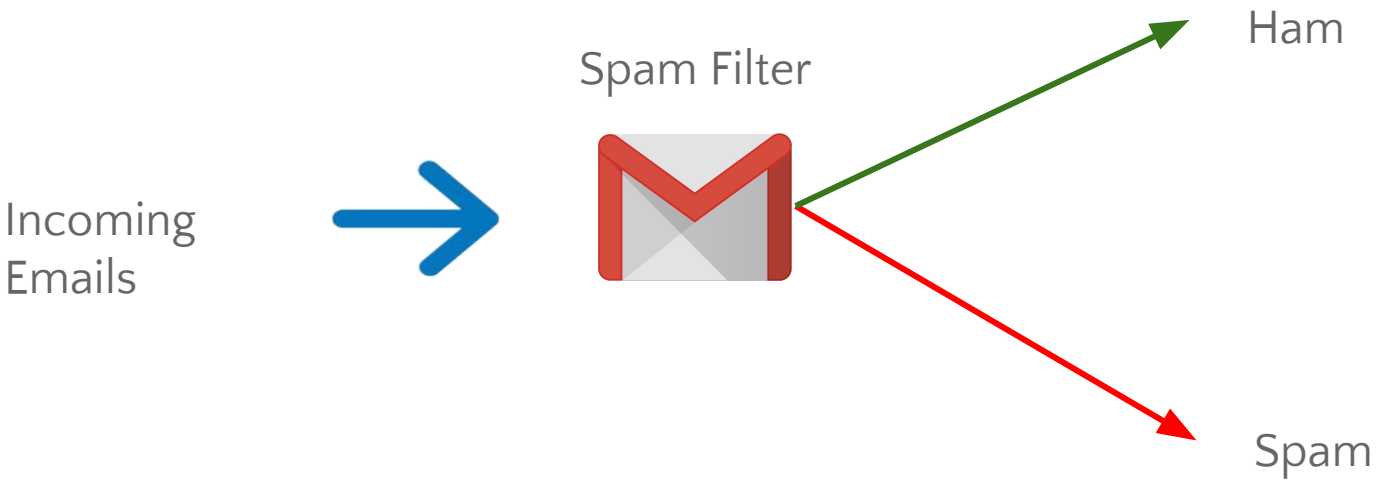
Another example:
C2 = {Cat, Dog}

Cat or Not Cat Classifier

C1 = {Cat, Not Cat}

CLOUD x LAB

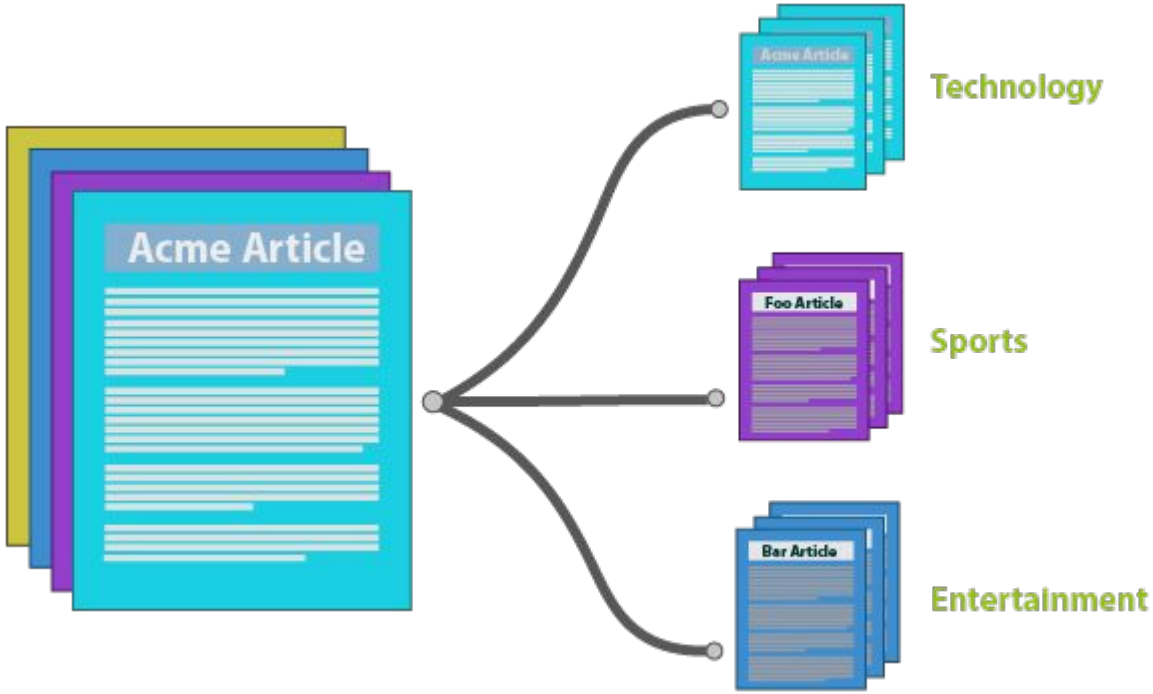
What is Classification?: Example-1



Spam Classifier

CLOUD x LAB

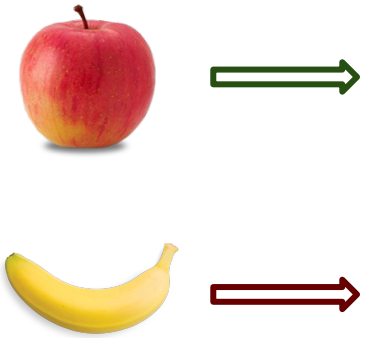
What is Classification ? Example-2



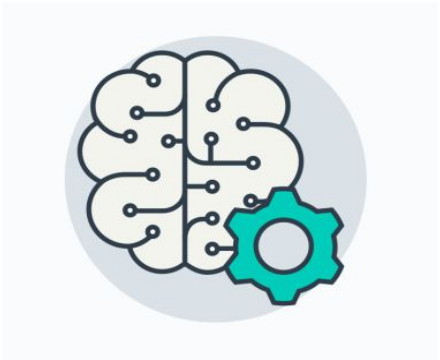
News Article Classifier

CLOUD x LAB

What is Classification?: Prediction



Images without Label



Model

Fruits Classifier

Apple

Banana

Model's Predictions

CLOUD x LAB

So how does classification training works?

CLOUD x LAB

Labeled Images for Building Model

Apple

Banana

Apple

Kiwi

Grapes

CLOUD x LAB

Labeled Images for Building Model

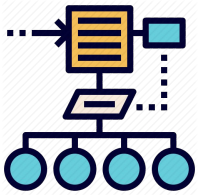
Apple

Banana

Apple

Kiwi

Grapes



Algorithm

Training Phase

CLOUD x LAB

Labeled Images for Building Model

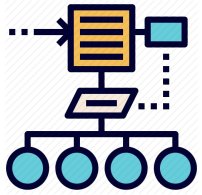
Apple

Banana

Apple

Kiwi

Grapes



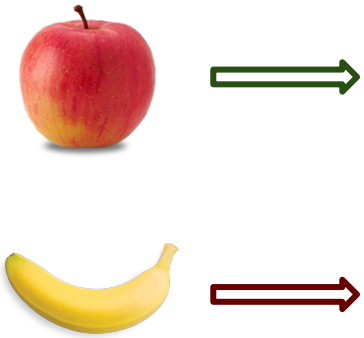
Algorithm

Training Phase

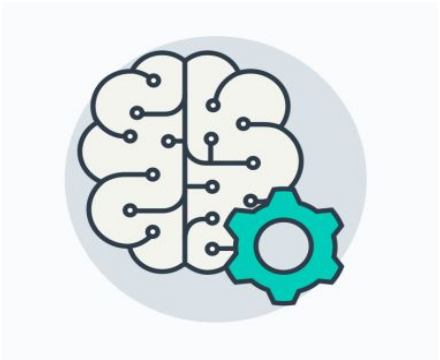
Model

CLOUD x LAB

Predictions



Images without Label



Model

Apple

Banana

Model's Predictions

Fruits Classifier

So how do we train the model in classification?

CLOUD x LAB

Labeled Images for Building Model

Apple

Banana

Apple

Kiwi

Grapes

Img-1	….	...	…….				Apple
Img-2							Banana
Img-3							Apple

Feature Matrix (X)

Label Vector (Y)

Train-test Split

Labeled Images for Building Model


Apple


Banana


Apple


Kiwi


Grapes

Split images into training and test set


Apple


Banana


Grapes


Kiwi

Training Set - 80% Data


Apple

Test Set - 20% Data

CLOUD x LAB

Labeled Images for Building Model


Apple


Banana


Apple


Kiwi


Grapes

Split images into training and test set


Apple


Banana

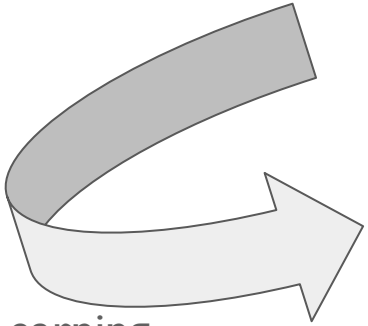

Grapes

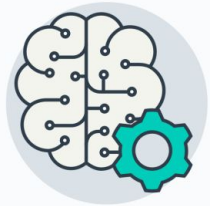

Kiwi

Training Set – 80% Data


Apple

Test Set – 20% Data





Using training set we build the model

AI & Deep Learning

CLOUD x LAB

Labeled Images for Building Model


Apple


Banana


Apple


Kiwi


Grapes

Split images into training and test set


Apple


Banana


Grapes


Kiwi

Training Set – 80% Data


Apple

Test Set – 20% Data

Using test set we evaluate performance of model

CLOUD × LAB

Classification Types

Cat

Not Cat

Binary Classification – Two Classes

Apple

Banana

Kiwi

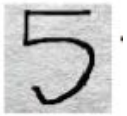
Multi Class Classification – Multiple Classes

CLOUD x LAB

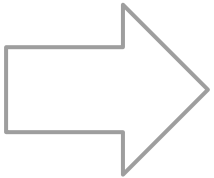
The goal of this session is to build a model to classify handwritten digits

CLOUD x LAB

Handwritten Digits Classifier

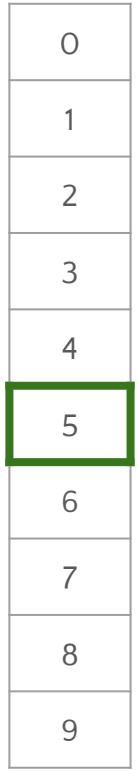


Input image



Model

Model classifies the image into one of the digits between 0 and 9



Model's prediction

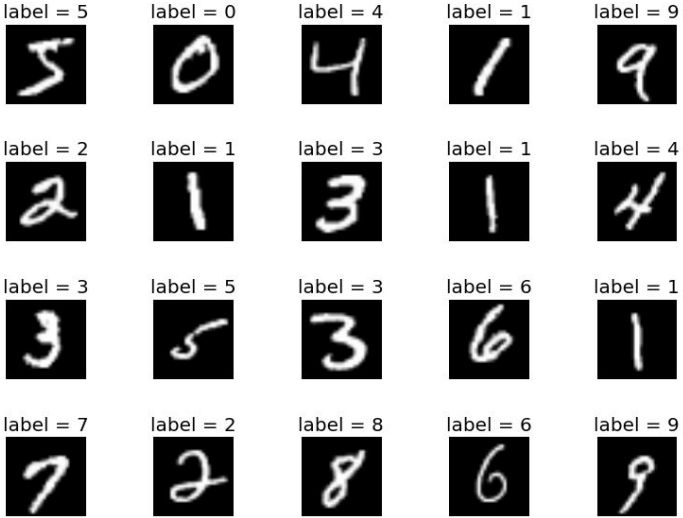
Multiclass Classification

CLOUD x LAB

Handwritten Digits Classifier - Dataset

Source: http://yann.lecun.com/exdb/mnist/

- In this problem, we will use MNIST dataset
  - Set of 70,000 small images
  - Each image is grayscale (black & white)
  - 28 by 28 pixels (total 784 pixels for an image)
  - The digits have been size-normalized and centered in a fixed-size image.

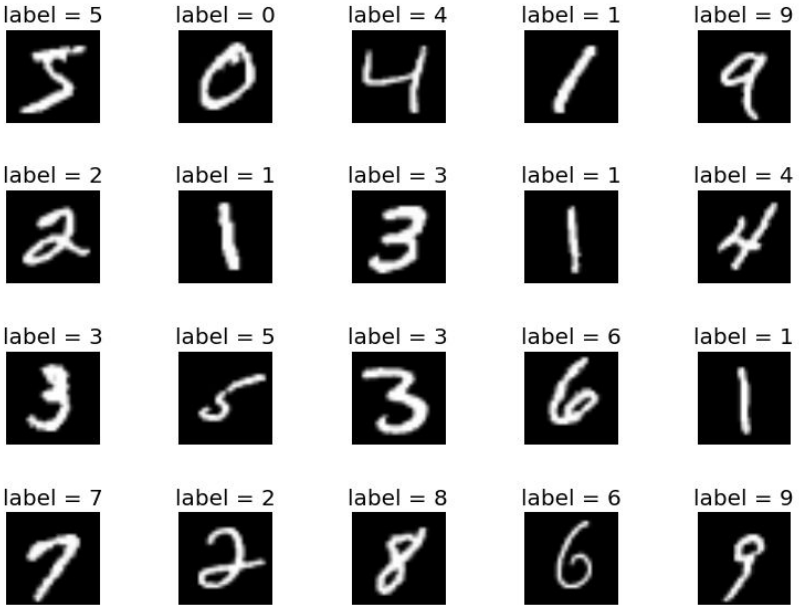


MNIST Dataset
(Modified National Institute of Standards and Technology)

CLOUD x LAB

Handwritten Digits Classifier – Dataset

- Handwritten by high school students and employees of the US Census Bureau.
- Each image is labeled with the digit it represents.

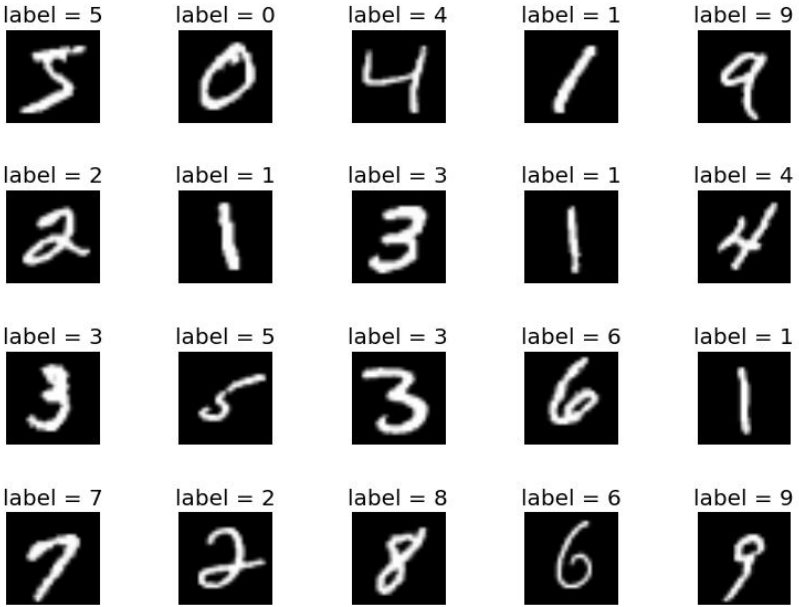


Every Image has Label Associated

CLOUD x LAB

Handwritten Digits Classifier - Dataset

- MNIST dataset is also called
  - "Hello World" of Machine Learning
- We have to build a model using this labelled dataset



Every Image has Label Associated

CLOUD x LAB

Handwritten Digits Classifier - Dataset

- Let's get the data and explore it
  - Go to Jupyter
  - Open New Terminal
  - Switch to the repository
    - `cd iitr-deep-learning-spl-tf2`
  - Get the latest code
    - `git pull origin master`

Handwritten Digits Classifier - Dataset

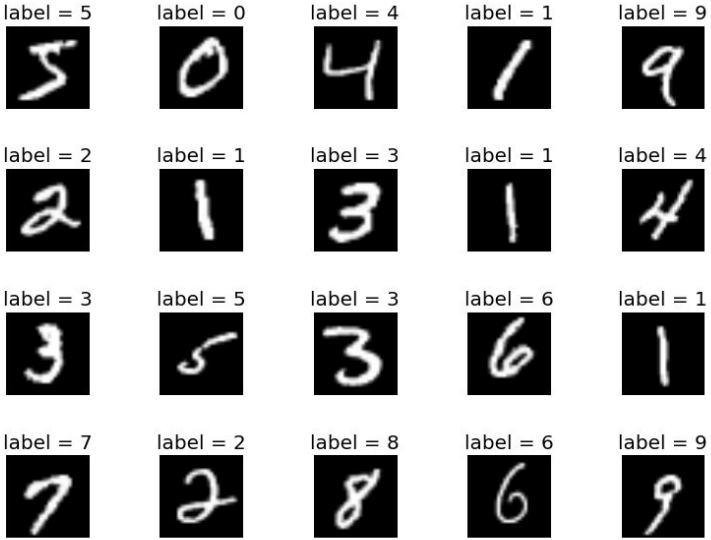
Fetch the Data

```
>>> from sklearn.datasets import fetch_openml
>>> mnist = fetch_openml('mnist_784', version=1)
>>> X, y = mnist["data"], mnist["target"]
```

Handwritten Digits Classifier - Dataset

So how do we start the training process?

Handwritten Digits Classifier – Training Process

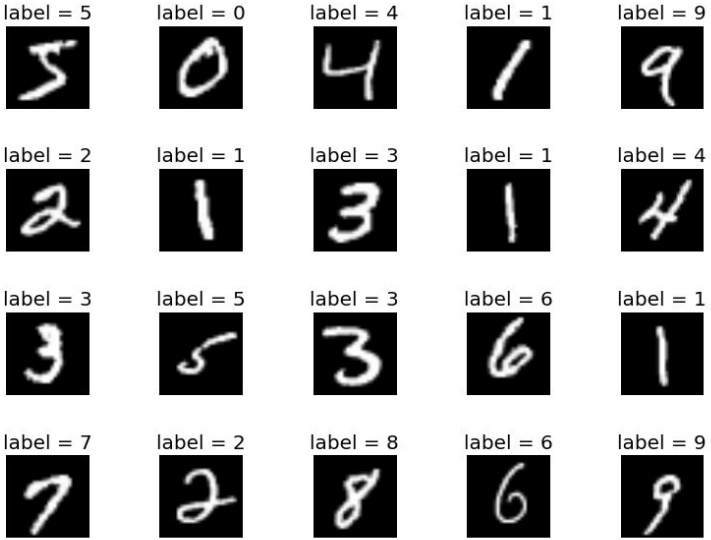


???

Tabular form –
Rows and Columns

CLOUD x LAB

Handwritten Digits Classifier – Training Process

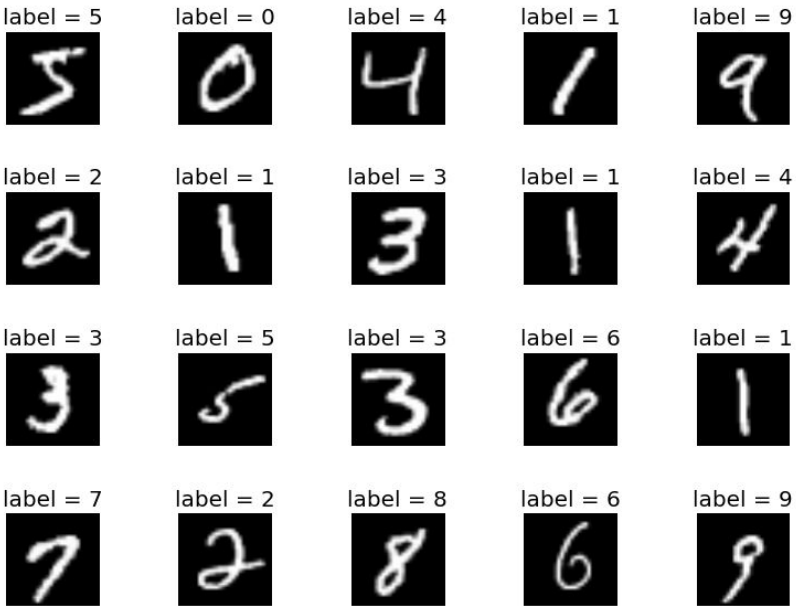


???

Instances, Features and Labels?

CLOUD x LAB

Handwritten Digits Classifier – Training Process

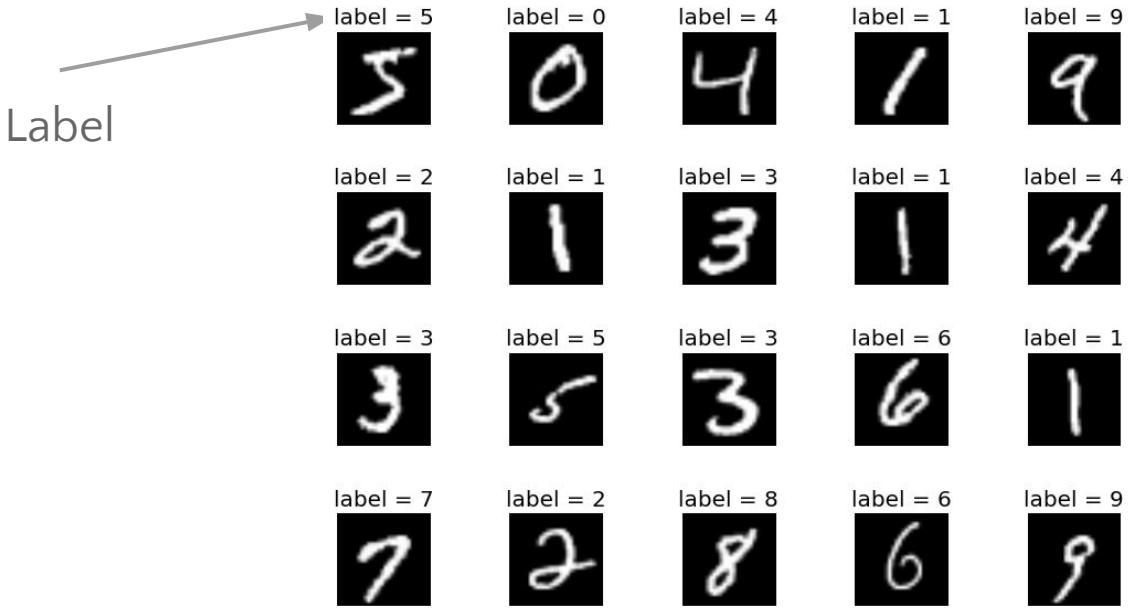


Total number of instances are seventy thousand

Each image is an instance

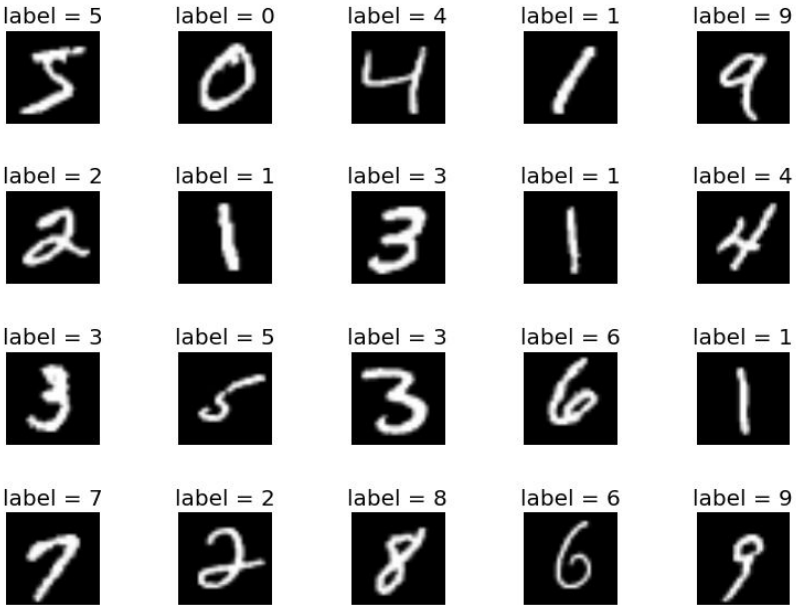
CLOUD x LAB

Handwritten Digits Classifier – Training Process



CLOUD x LAB

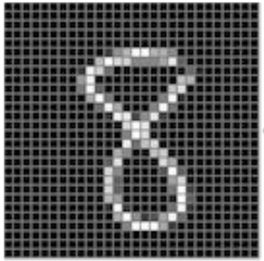
Handwritten Digits Classifier – Training Process



Features?

CLOUD x LAB

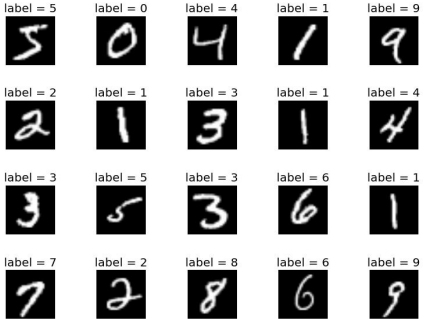
0 : Black Color
255 : White Color
1 - 254 : other shades



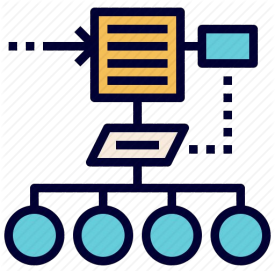
28 x 28
784 pixels

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	4	62	146	182	254	254	181	176	139	15	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	34	186	253	217	208	136	136	136	166	232	99	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	61	242	208	111	3	0	0	0	0	0	18	32	107	43	0	0	0	0	0	0
0	0	0	0	0	0	0	0	156	242	23	0	0	0	0	0	0	0	13	191	181	6	0	0	0	0	0	0
0	0	0	0	0	0	0	0	121	255	98	3	0	0	0	0	0	8	194	225	12	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	169	253	120	3	0	0	0	0	128	247	51	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	3	111	244	169	19	0	14	131	249	117	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	59	241	235	72	142	229	66	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	25	218	254	231	36	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	133	253	221	33	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	19	237	111	196	217	19	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	174	138	0	23	193	204	18	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	96	224	0	0	0	25	218	169	3	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	215	138	0	0	0	0	86	253	99	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	215	97	0	0	0	0	3	162	214	11	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	215	97	0	0	0	0	0	118	253	68	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	185	157	0	0	0	0	0	40	254	98	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	50	244	61	0	0	0	0	112	244	58	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	174	251	142	59	83	167	244	111	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	6	133	253	253	253	169	61	3	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Handwritten Digits Classifier – Training Process



Input data in tabular form – Includes both features and labels



Algorithm

Model

CLOUD x LAB

Fetching MNIST dataset in Scikit-Learn

How to visualize a tabular entry into image form

```
>>> %matplotlib inline
>>> import matplotlib as mpl
>>> import matplotlib.pyplot as plt

>>> some_digit = X[0]
>>> some_digit_image = some_digit.reshape(28, 28)
>>> plt.imshow(some_digit_image, cmap=mpl.cm.binary)
>>> plt.axis("off")
>>> plt.show()
```

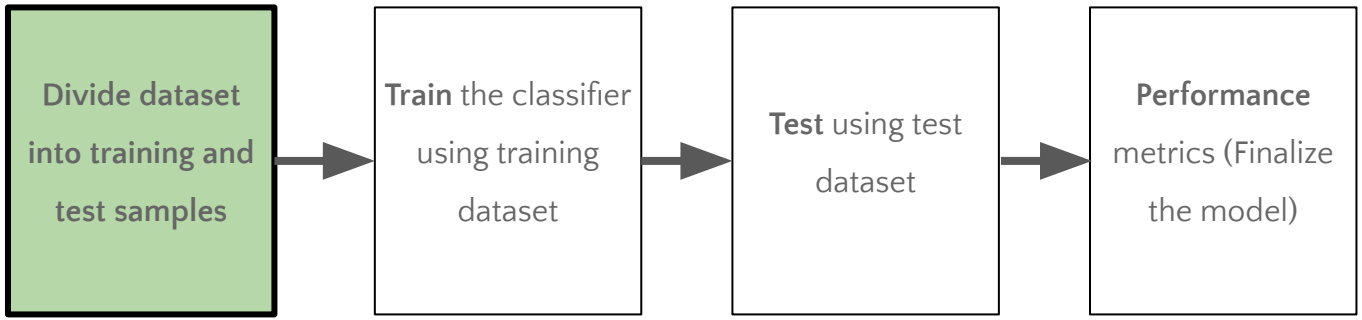
Switch to Notebook

Fetching MNIST dataset in Scikit-Learn

Looking at one of the data samples

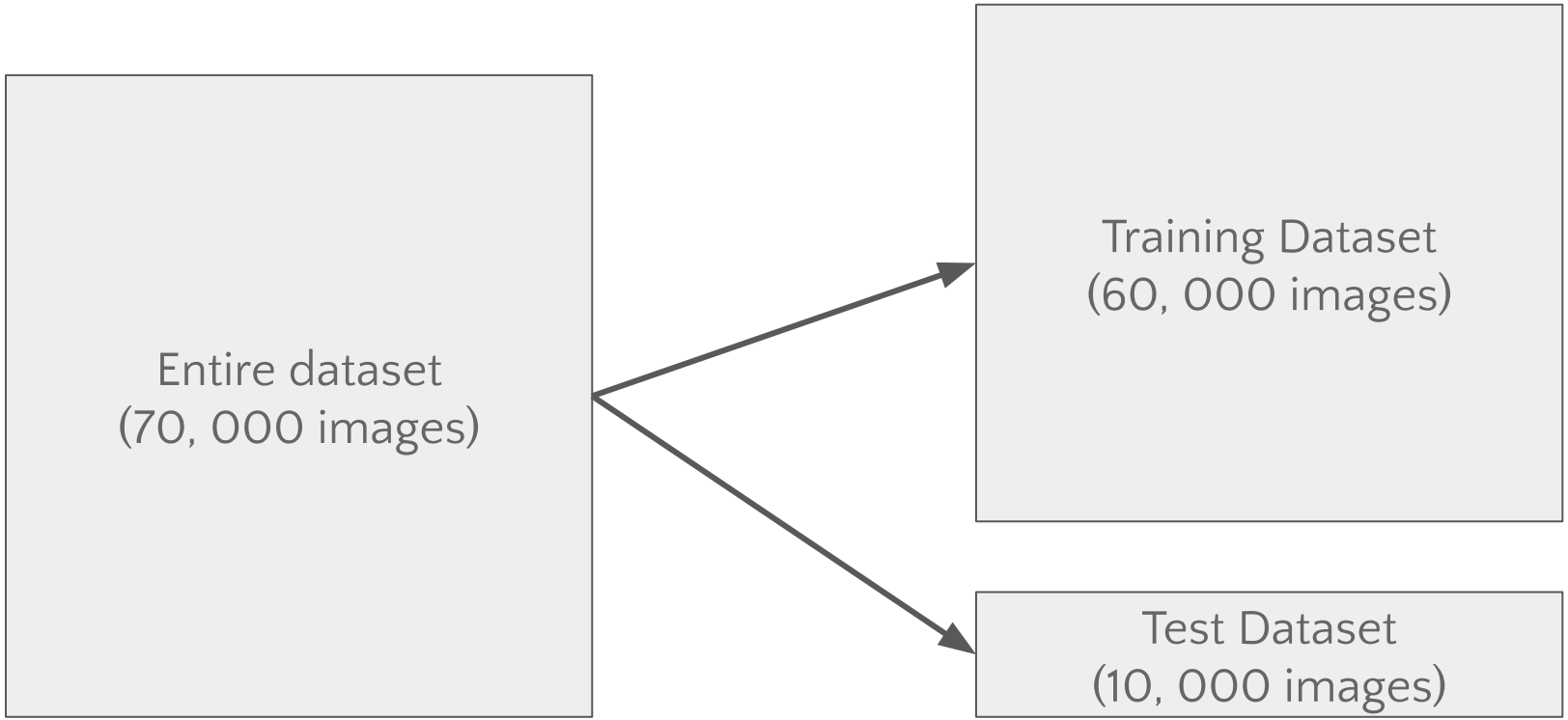
Switch to Notebook

Handwritten Digits Classifier - Training Process - Steps



CLOUD x LAB

Training and Test dataset



CLOUD x LAB

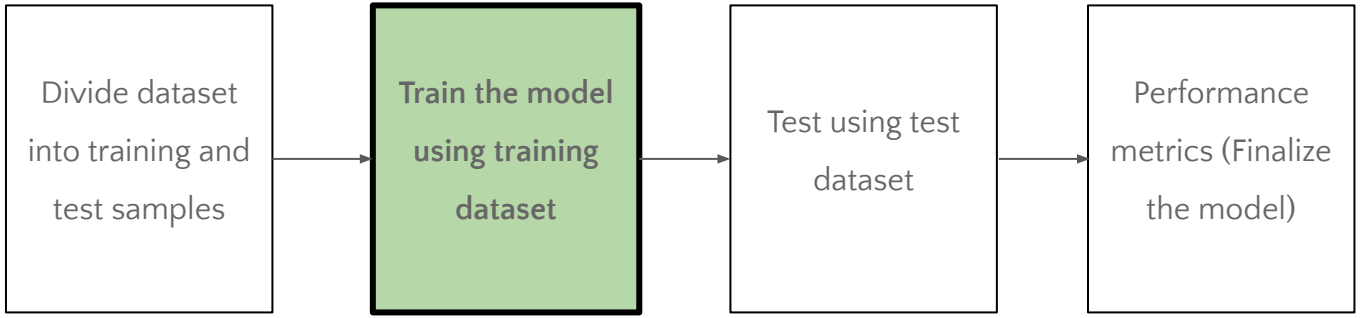
Dividing dataset into training and test samples

Dividing dataset into training and test in python:

```
>>> X_train, X_test, y_train, y_test = X[:60000], X[60000:], y[:60000], y[60000:]
```

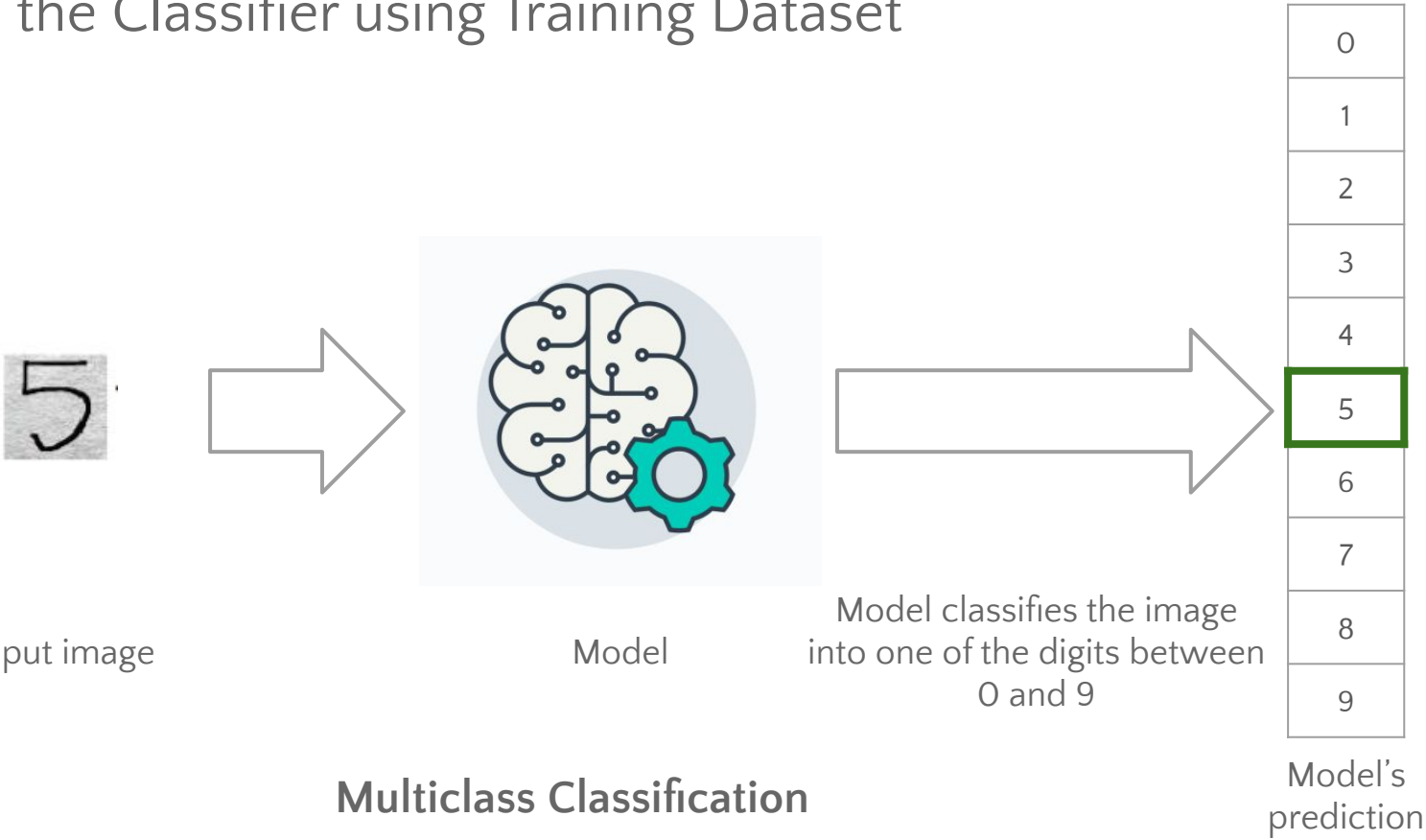
Switch to Notebook

Steps



CLOUD x LAB

Train the Classifier using Training Dataset



Multiclass Classification

CLOUD x LAB

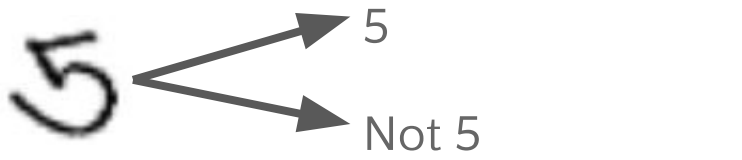
Train the Classifier using Training Dataset

For simplicity, let's first train a binary classifier instead of multiclass classifier

CLOUD x LAB

Train the Classifier using Training Dataset

We will train a binary classifier called 5-detector



Input: Image Output: Classification

5 - Detector

CLOUD x LAB

Train the Classifier using Training Dataset

Lets change our training and test set labels in MNIST to train 5 detector

```
>>> y_train_5 = (y_train == 5) # True for all 5s, False for all other digits
>>> y_test_5 = (y_test == 5) # True for all 5s, False for all other digits
```

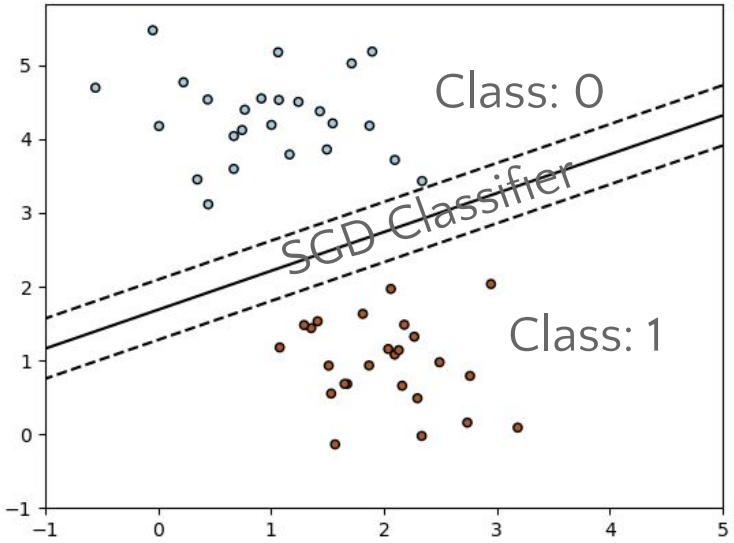
Train the Classifier using Training Dataset

We are going to use: Stochastic Gradient Descent (SGD) Classifier (a linear classifier)

SGD - Stochastic Gradient Descent Classifier

For the two-dimensional (2 features) training dataset,

- Trying to estimate the coefficients of the line (*WX+b*) which can be
- *W*-weights vector to feature matrix *X*, *b*- bias (from which point in *Y* the line should start at *x=0*)
- Best-fitted dividing the two categories



CLOUD x LAB

SGD – Stochastic Gradient Descent Classifier

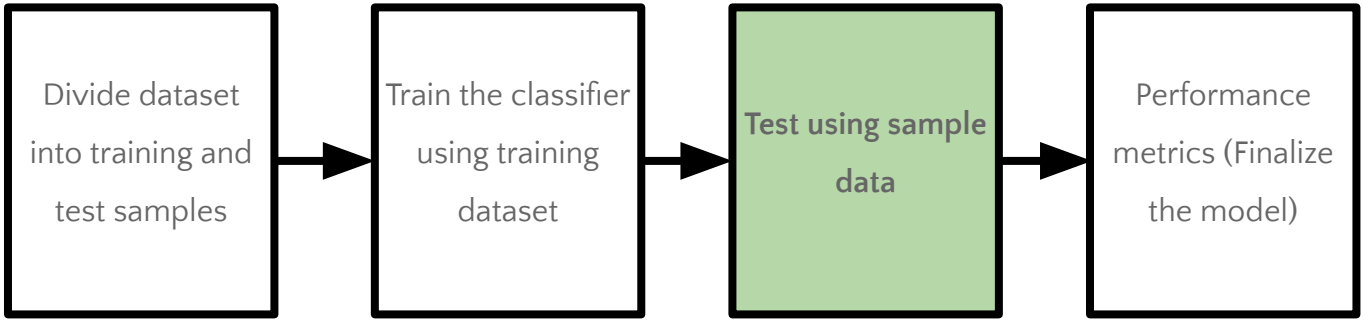
- Stochastic Gradient Descent (SGD) Classifier
  - Capable of handling large datasets
  - Well suited for online training

Training SGDClassifier

Original problem: '5' and 'Not 5' classifier

```
>>> from sklearn.linear_model import SGDClassifier

>>> sgd_clf = SGDClassifier(random_state=42) #initialize the model
>>> sgd_clf.fit(X_train, y_train_5) #fit the model
```

Steps



CLOUD x LAB

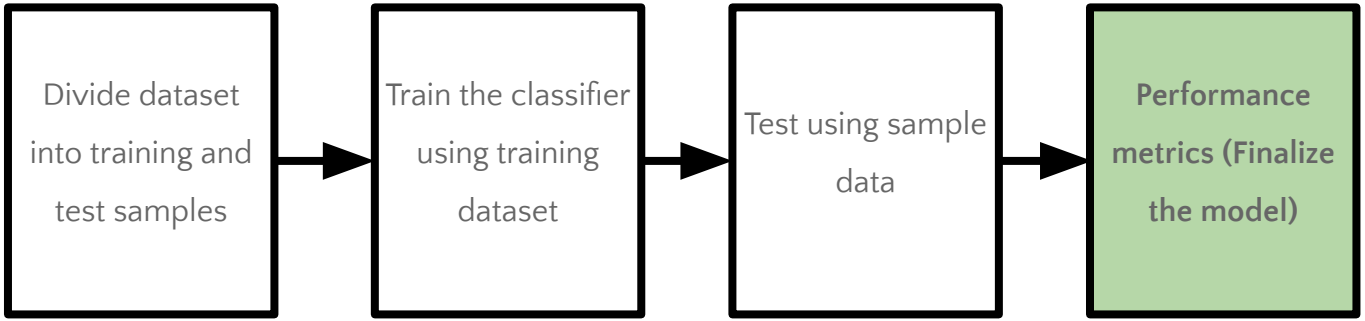
Testing SGDClassifier

- First image stores the digit 5 and the classifier correctly classifies it as ‘True’

```
>>> some_digit = X[0]  # Taking the First image
>>> sgd_clf.predict([some_digit])
array([True], dtype=bool)
```

Pause for Questions

Steps

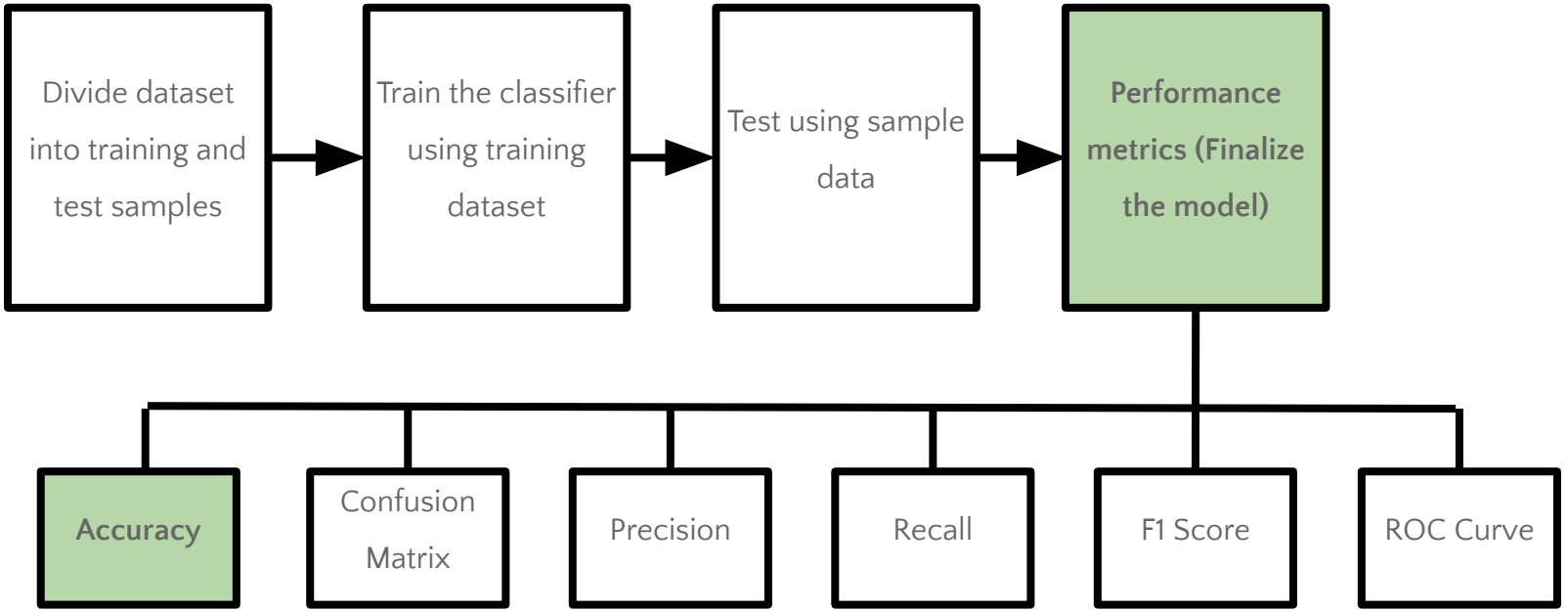


CLOUD x LAB

Measuring Performance of Classifier

- Accuracy
- Confusion Matrix
  - Precision
  - Recall
  - F1 score
- ROC Curve

Steps



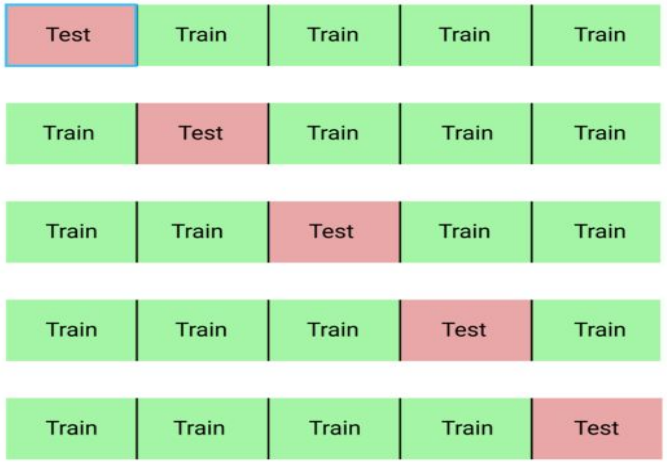
CLOUD x LAB

Performance Measures - Test Data Vs Cross-Validation

Case-1: Measure performance on test data: Report accuracy on test data only

Case-2: Measure performance on different splits (folds) of the entire data (Cross-validation): Report the combined accuracy

5-fold cross-validation →



CLOUD x LAB

Performance Measures – Test Data Vs Cross-Validation

cross_val_score

As discussed in end-to-end project session, cross_val_score() function in scikit-learn can be used to perform cross validation

Performance Measures – Cross Validation

Initiating cross validation method

```
>>> from sklearn.model_selection import cross_val_score

>>> cross_val_score(sgd_clf, X_train, y_train_5, cv=3, scoring="accuracy")
```

The scoring attribute can be “f1” for f-score or any other metric suitable to your model.
Refer: https://scikit-learn.org/stable/modules/model_evaluation.html

Switch to Notebook

Performance Measures – Cross Validation

```
>>> array([0.95035, 0.96035, 0.9604 ])
```

- The resulting accuracy is above 95 % for each of the folds

CLOUD x LAB

Performance Measures – Cross Validation

Is the accuracy of 95% good enough?

CLOUD x LAB

Performance Measures – Cross Validation

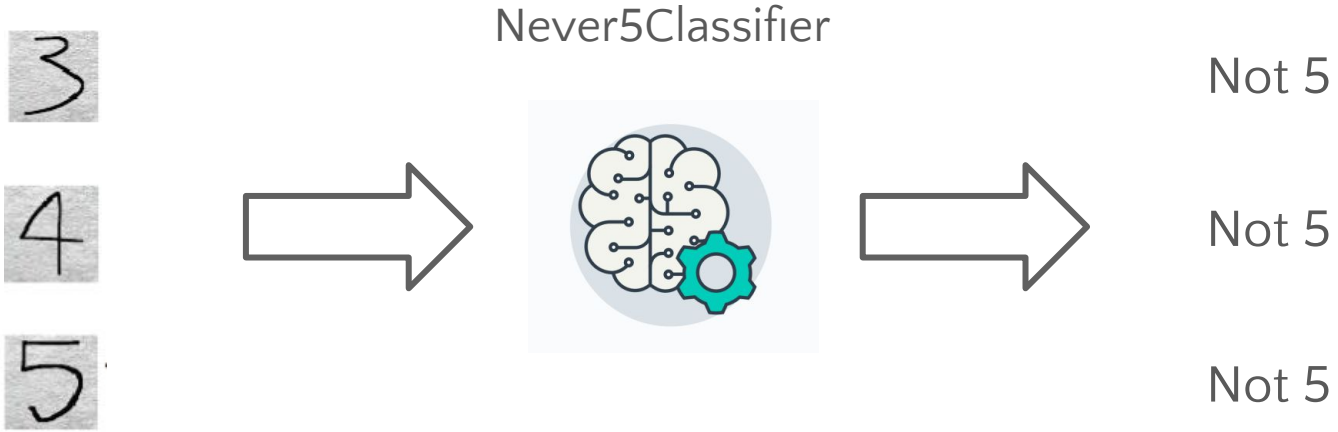
Is the accuracy of 95% good enough?

Let's see

CLOUD x LAB

Performance Measures – Cross Validation

For Now Let's Build Another Classifier – Never5Classifier – Which classifies every image as "Not 5"



CLOUD x LAB

Performance Measures – Cross Validation

Never5Classifier

```
>>> from sklearn.base import BaseEstimator
>>> class Never5Classifier(BaseEstimator):
      def fit(self, X, y=None):
          pass
      def predict(self, X):
          return np.zeros((len(X), 1), dtype=bool)
```

Switch to Notebook

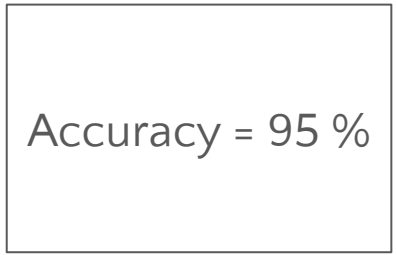
Performing Cross validation in Scikit learn

Calculating accuracy using cross_val_score

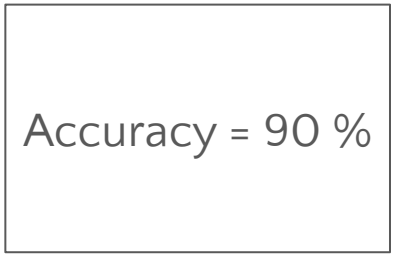
```
>>> never_5_clf = Never5Classifier()
>>> cross_val_score(never_5_clf, X_train, y_train_5, cv=3, scoring="accuracy")
```

Switch to Notebook

Performing Cross validation in Scikit learn

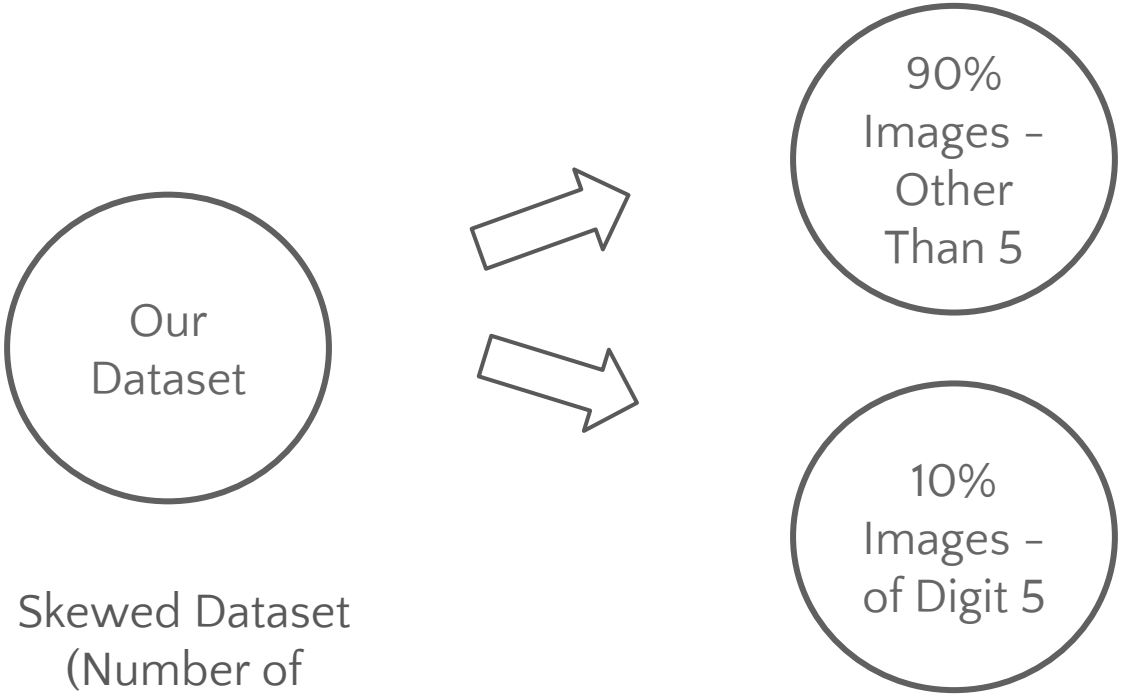


SGDClassifier



Dumb Classifier – Never5 Classifier

Skewed Dataset



Skewed Dataset (Number of instances are not equal)

Class Imbalance

CLOUD x LAB

Input

Model's Prediction

90% Images - Other Than 5

10% Images - of Digit 5

Dumb Classifier - Classifies every image as "Not 5"

Accuracy → 90%

Not 5

Not 5

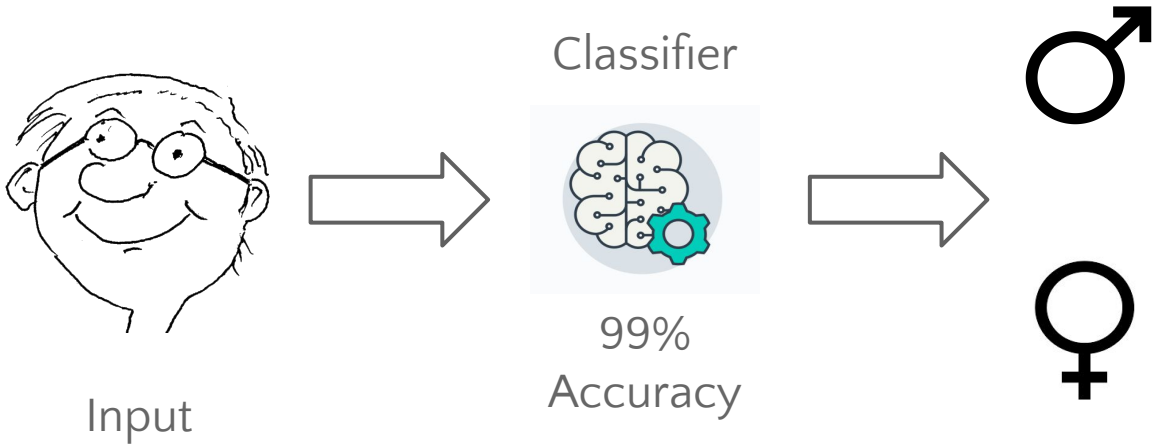
CLOUD x LAB

Learnings??

- Accuracy may not be a good performance measure when dealing with skewed datasets

CLOUD x LAB

Question – Let's say your friend has built a model which classify an image into a male or female with 99% accuracy. What questions you will ask your friend to make sure his model is fine?



Questions???

CLOUD x LAB

Labeled Images for Building Model

Male

Male

Male

Female

Male

% Males ???

% Females ???

Question – Percentage of males and females in training dataset

CLOUD x LAB

Labeled Images for Building Model


Male


Male


Male


Female


Male

80% Males

20% Females

Your friend answers 80% male and 20% females

CLOUD x LAB

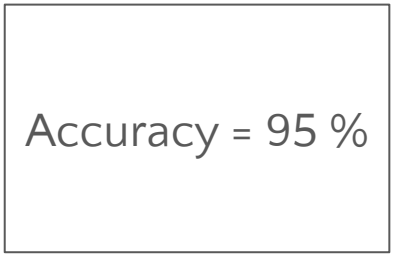
Input

Classifier

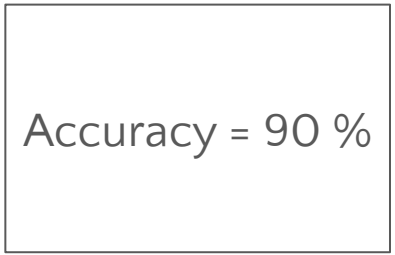
99%
Accuracy

Since dataset is skewed hence you will not trust his model

Performing Cross validation in Scikit learn



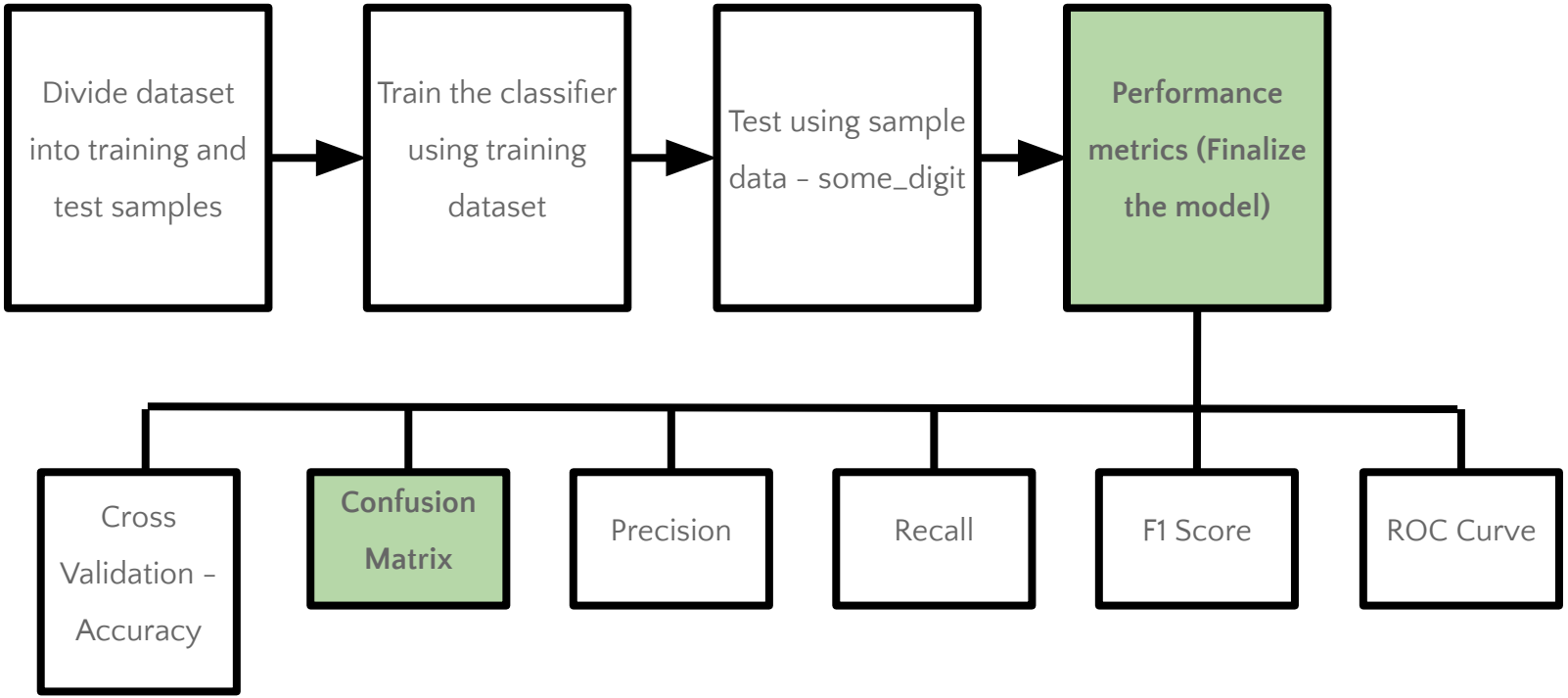
SGDClassifier



Dumb Classifier – Never5 Classifier

We need a better measure of performance for the classifier

Steps



CLOUD x LAB

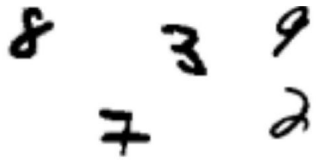
Performance Measures – Confusion Matrix

	Predicted YES	Predicted NO
Actual YES	TP 620	FN 380
Actual NO	FP 180	TN 8820

CLOUD x LAB

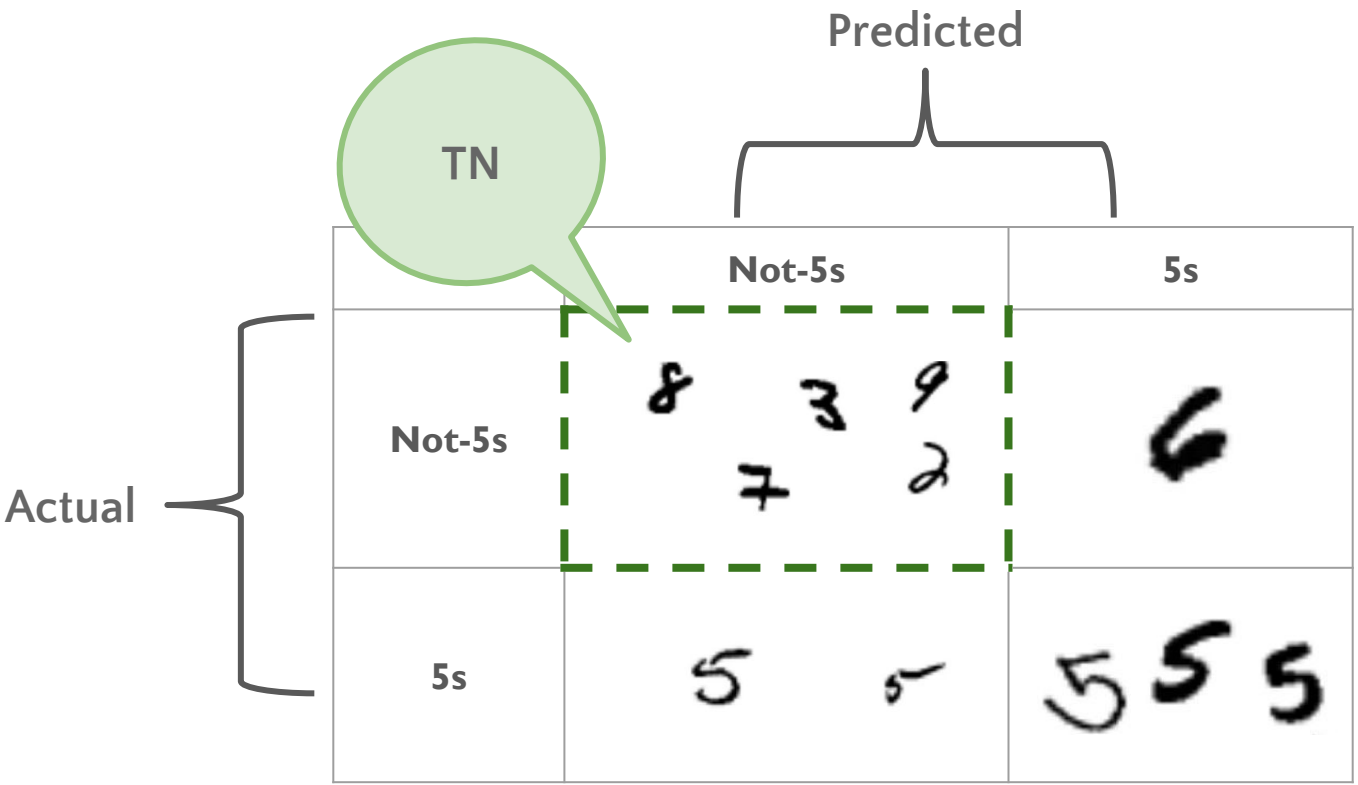
Predicted

Actual

	Not-5s	5s
Not-5s		
5s		

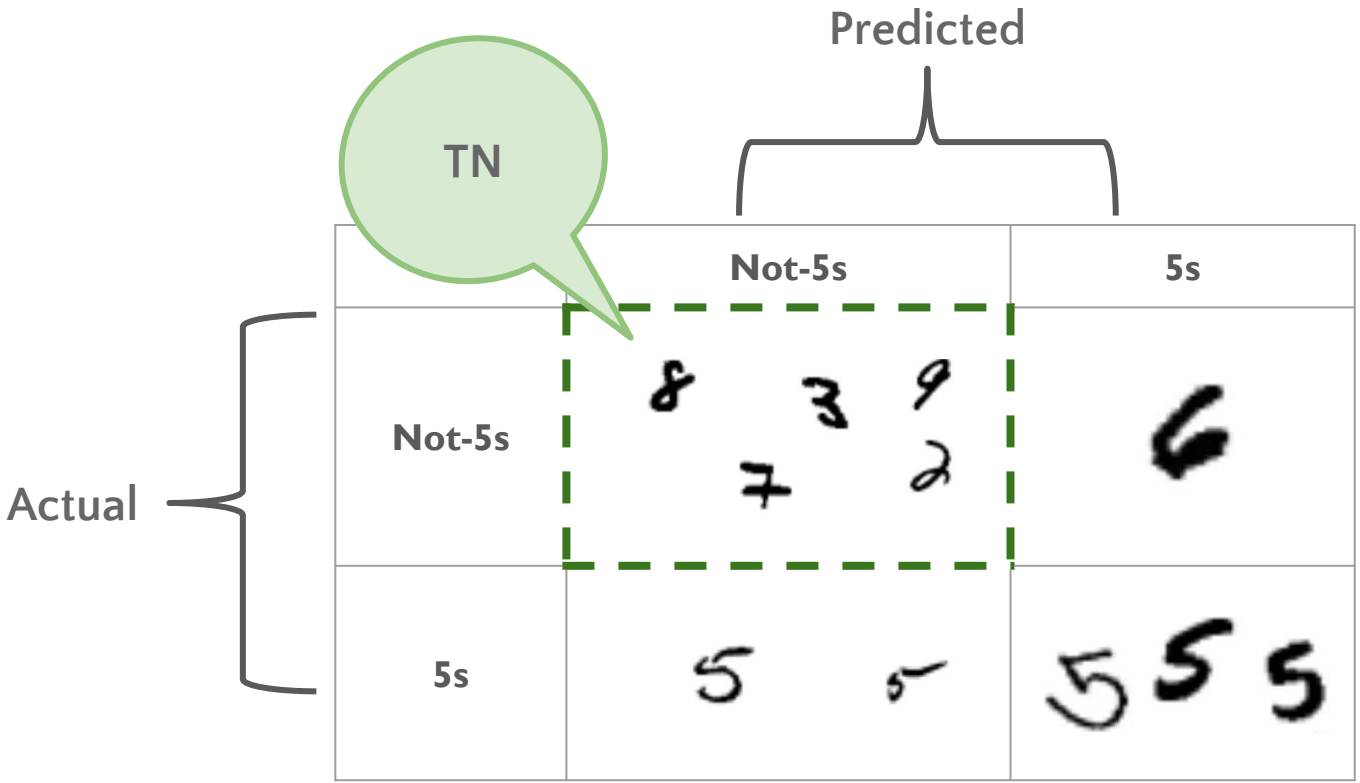
Sample Confusion Matrix of 5-Detector

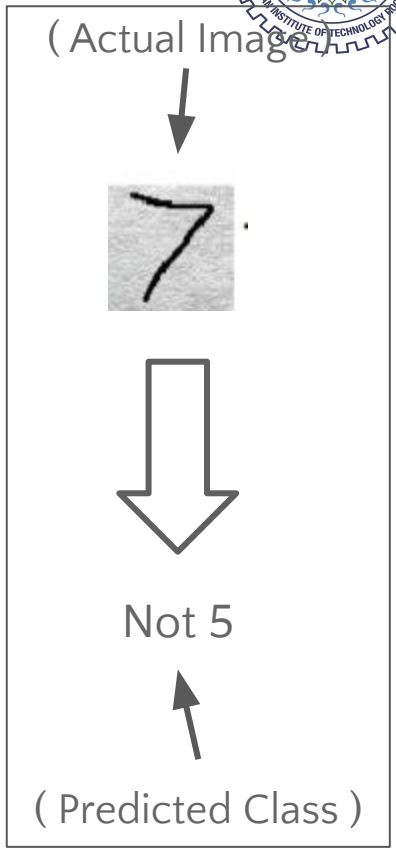
CLOUD x LAB



TN – True Negatives

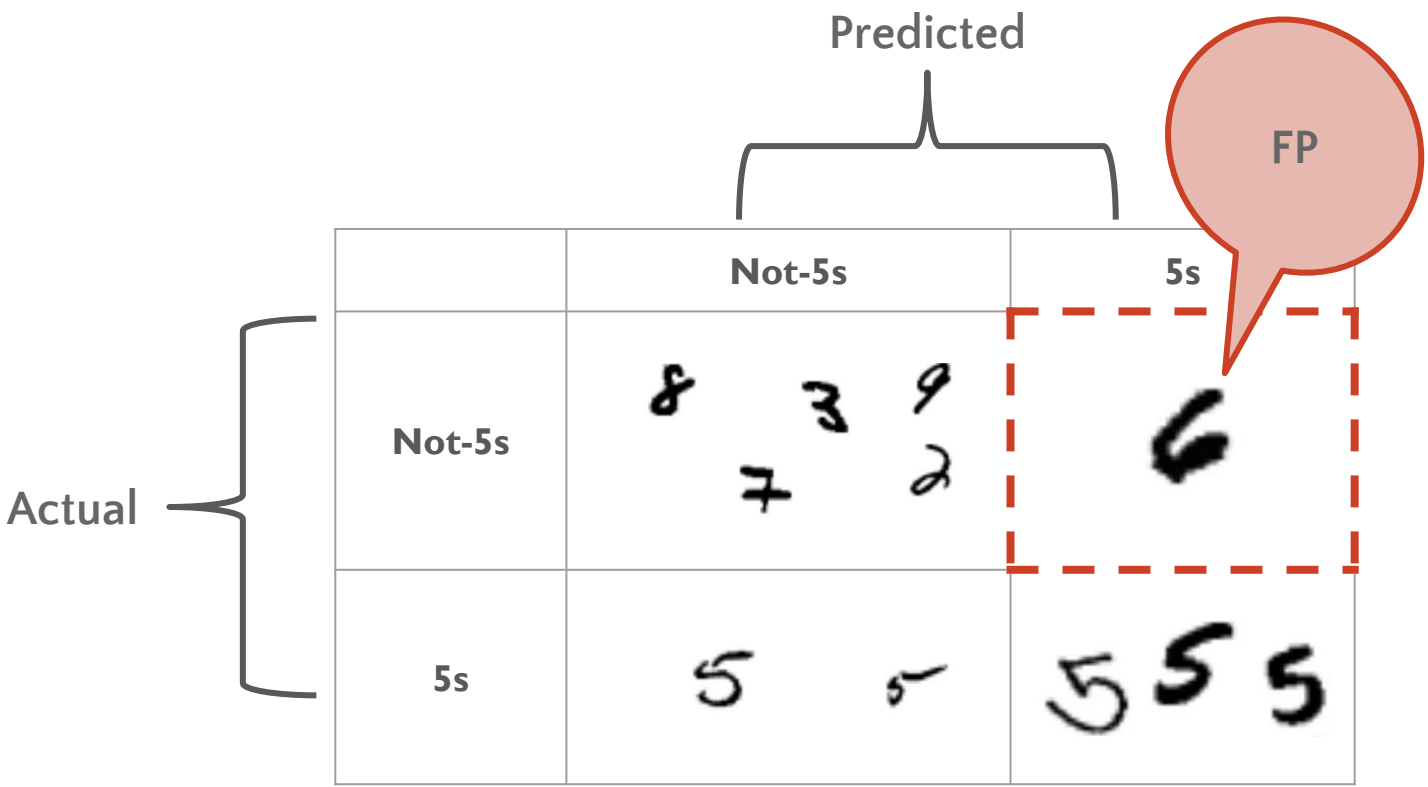
CLOUD x LAB





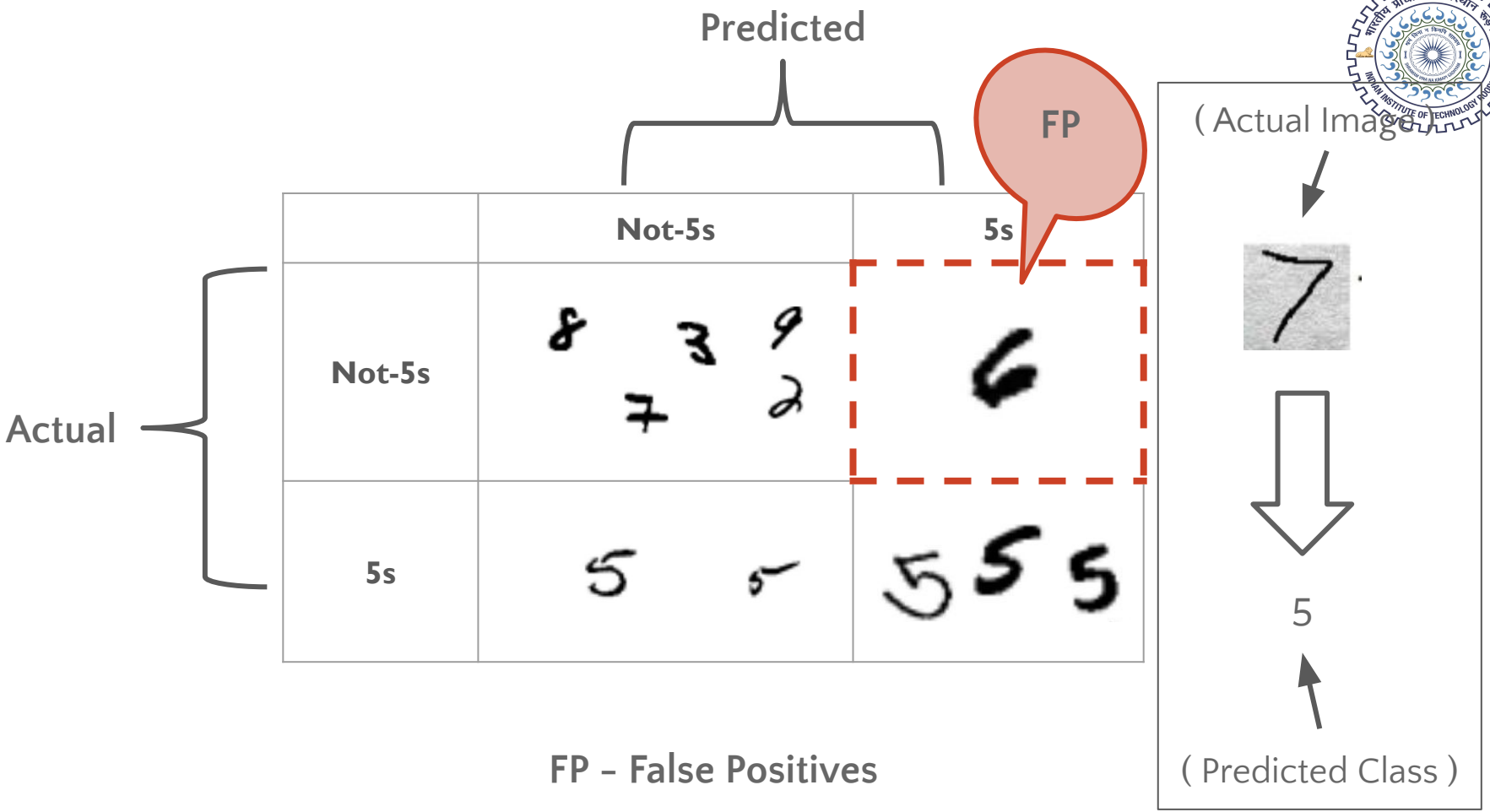
TN – True Negatives

CLOUD × LAB

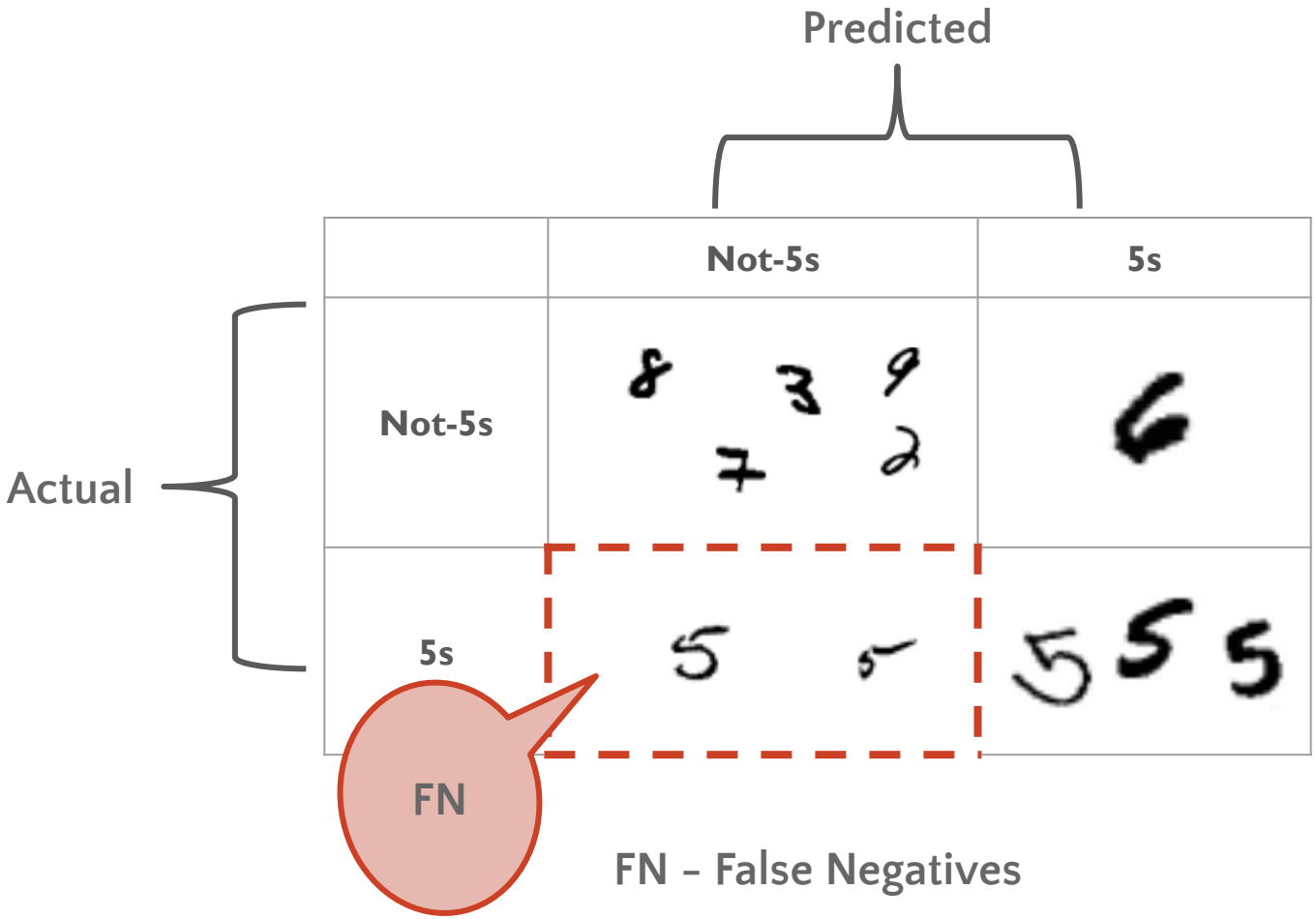


FP – False Positives

CLOUD x LAB

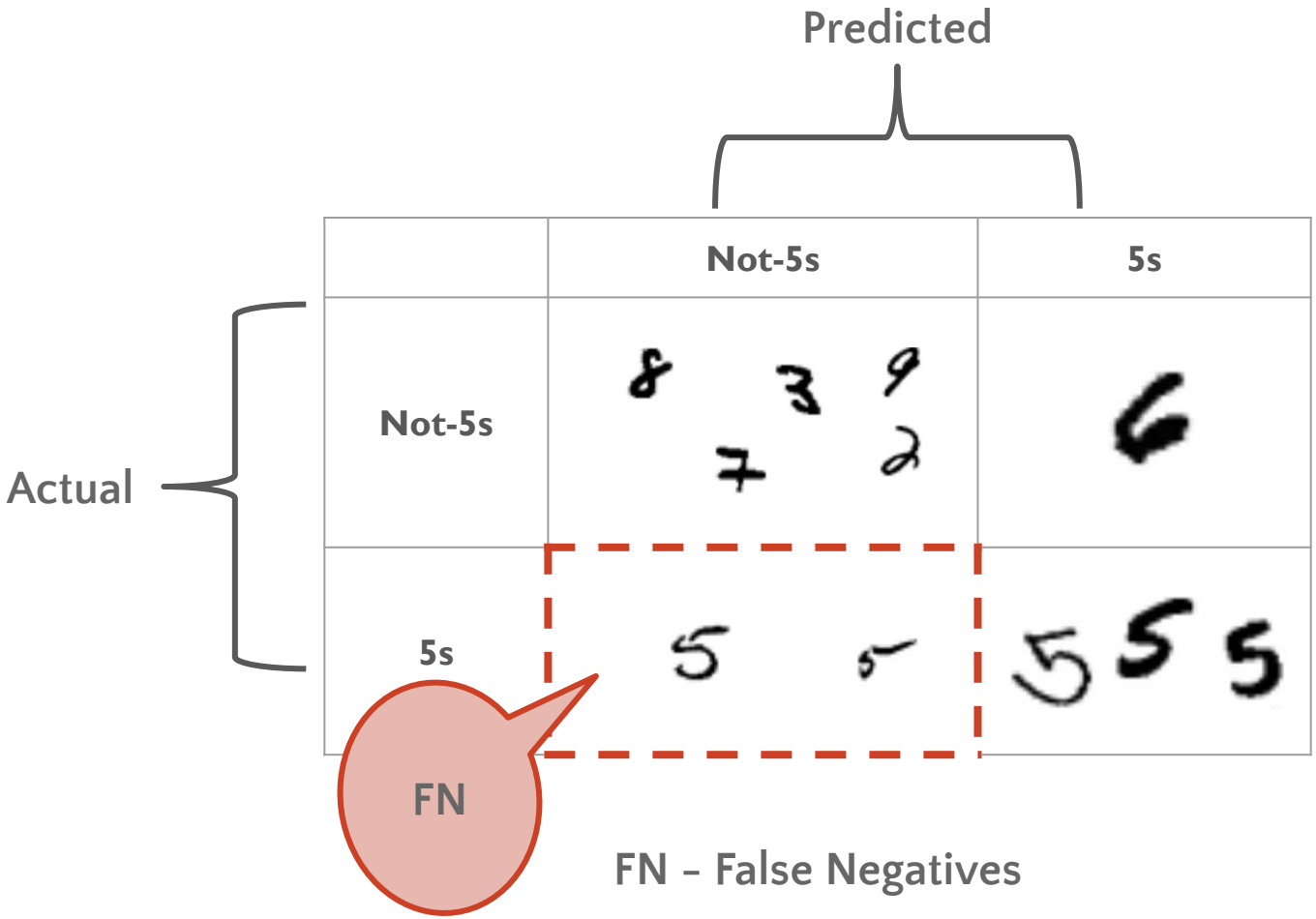

Predicted
FP
Not-5s
5s
Actual
Not-5s
5s
(Actual Image)
5
(Predicted Class)
FP – False Positives

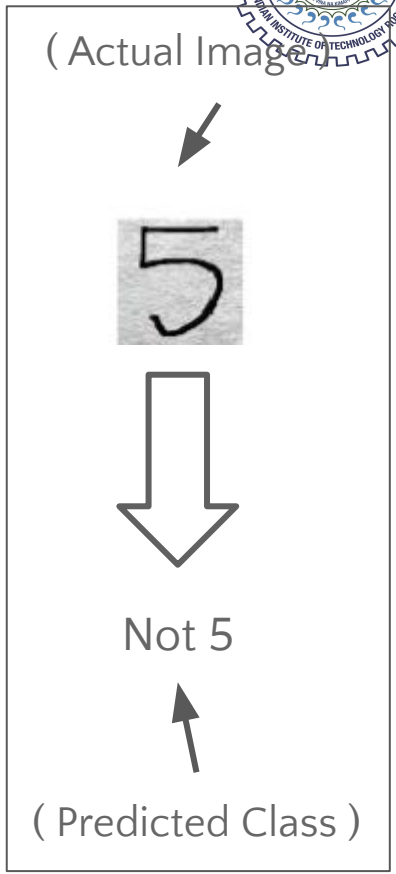
CLOUD × LAB



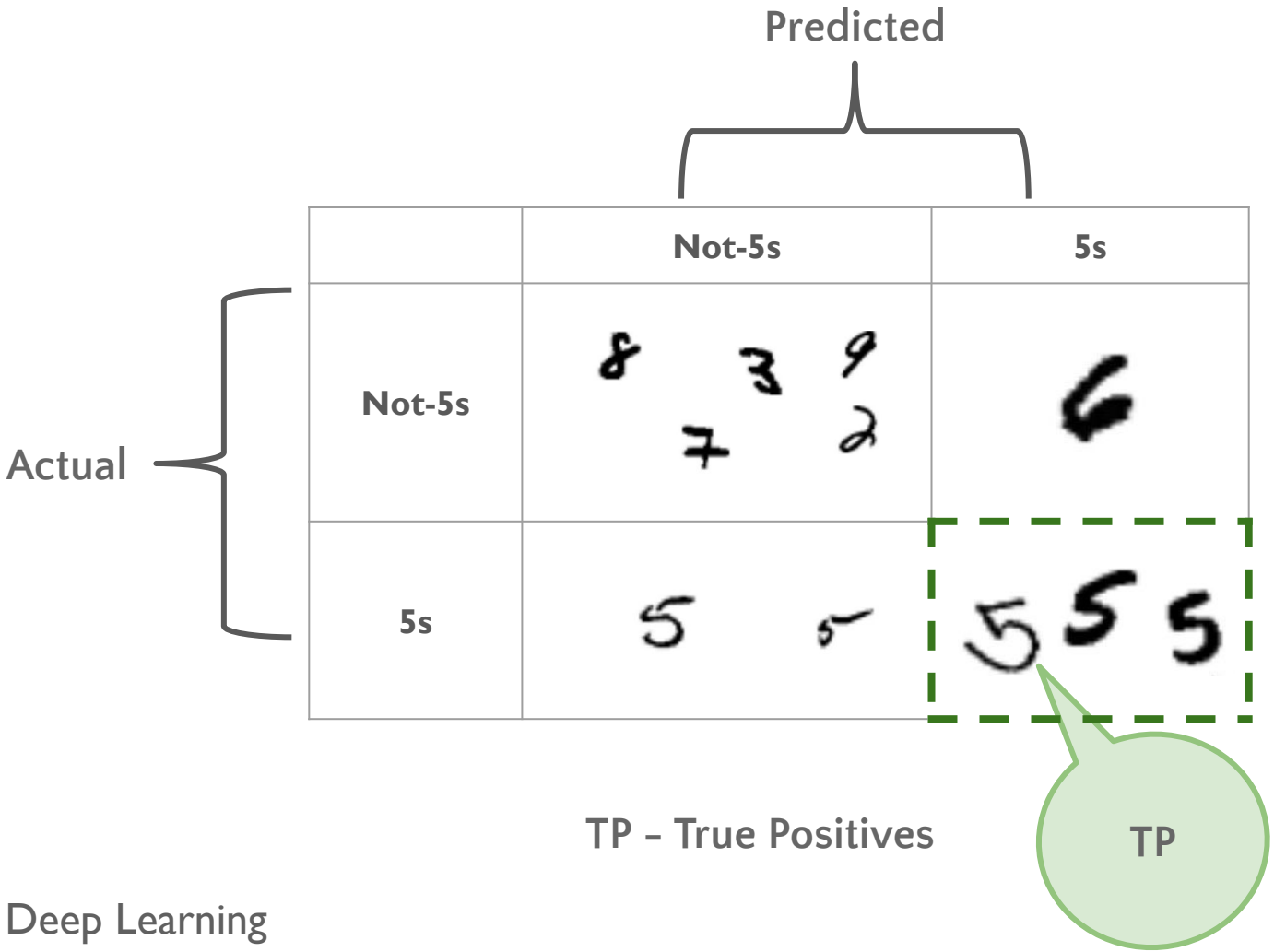
FN – False Negatives

CLOUD x LAB


Predicted
Not-5s
5s
Actual
Not-5s
5s
FN
FN – False Negatives

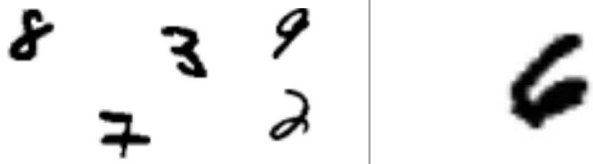

(Actual Image)
Not 5
(Predicted Class)

CLOUD × LAB



CLOUD × LAB

Predicted

Actual	Not-5s	5s
Not-5s		
5s		

TP – True Positives

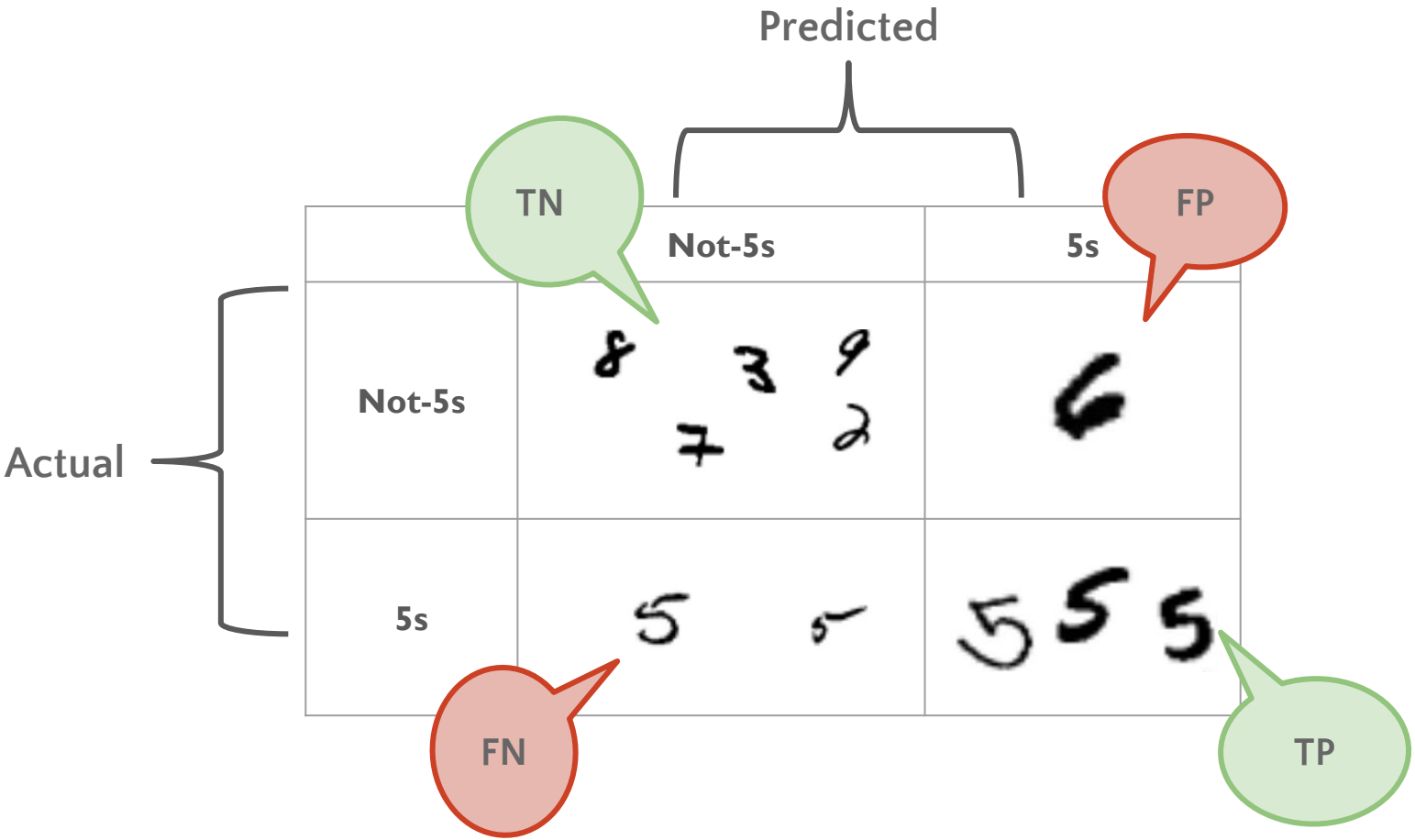
TP

(Actual Image)

5

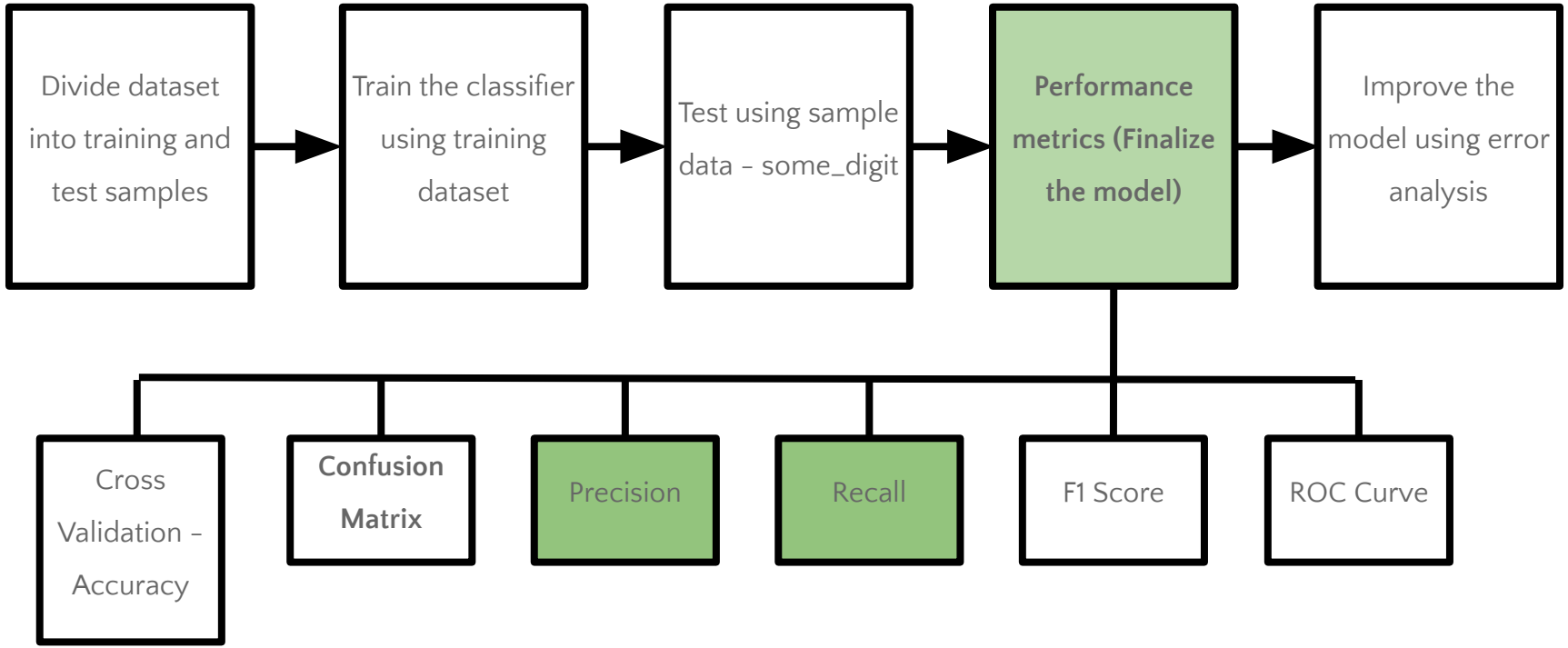
(Predicted Class)

CLOUD × LAB



CLOUD x LAB

Steps



CLOUD x LAB

Pause for Questions

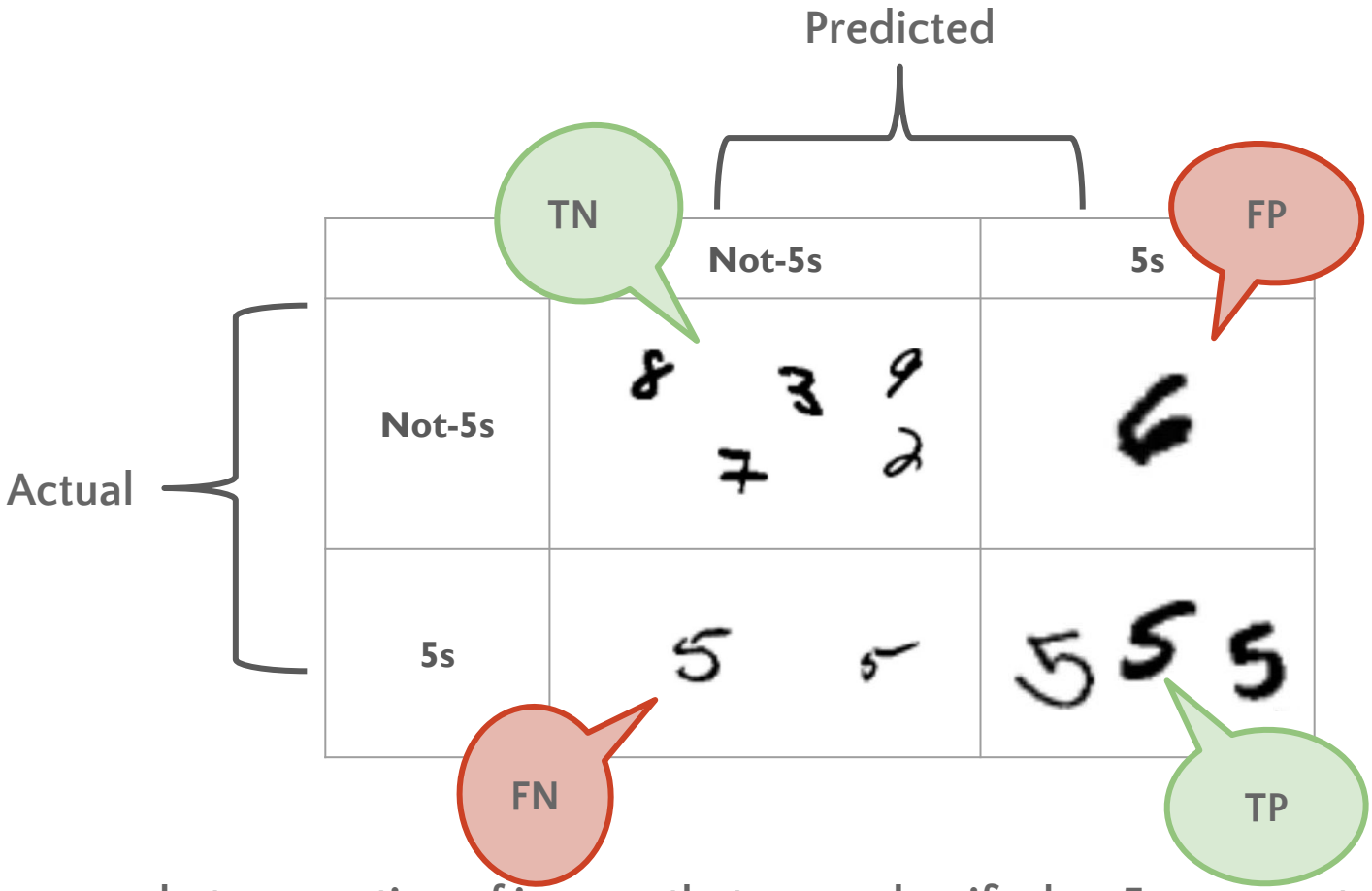
Precision and Recall

CLOUD x LAB

Performance Measures – Precision

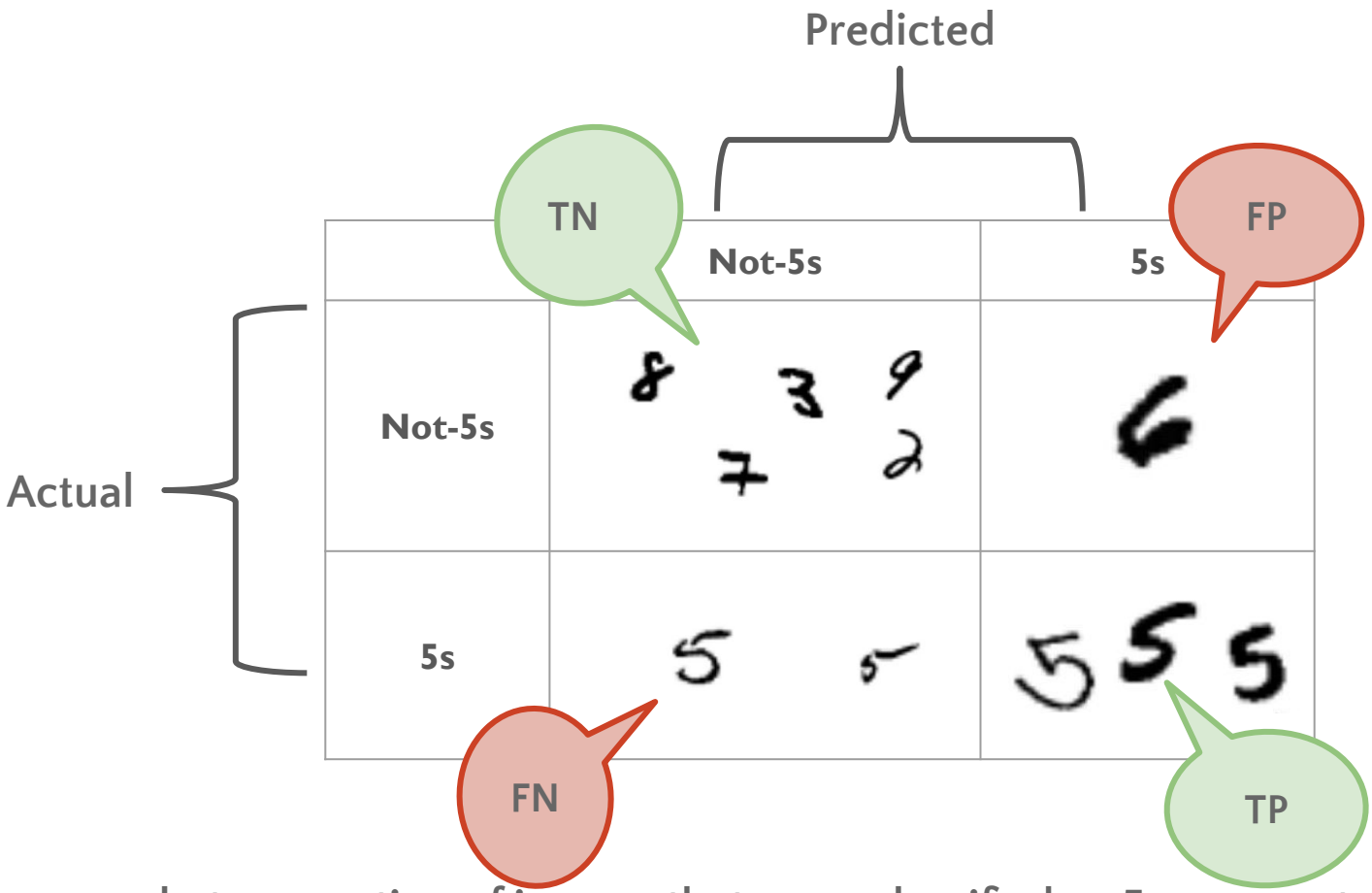
Precision means lack of mistakes

CLOUD x LAB



what proportion of images that were classified as 5s were actually 5

CLOUD x LAB

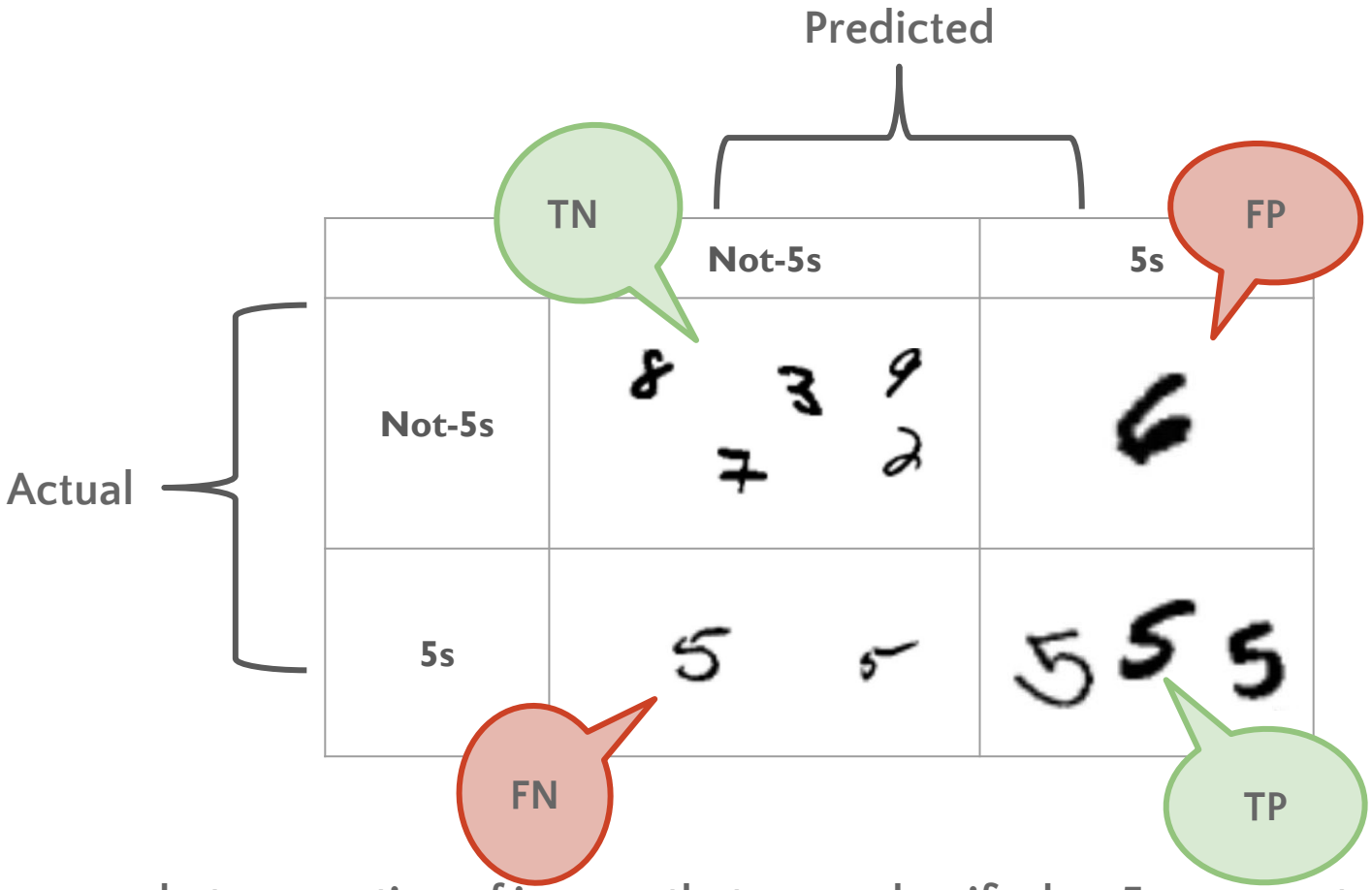


Precision

$$\text{precision} = \frac{TP}{TP + FP}$$

what proportion of images that were classified as 5s were actually 5

CLOUD × LAB



Precision –
3 out of 4

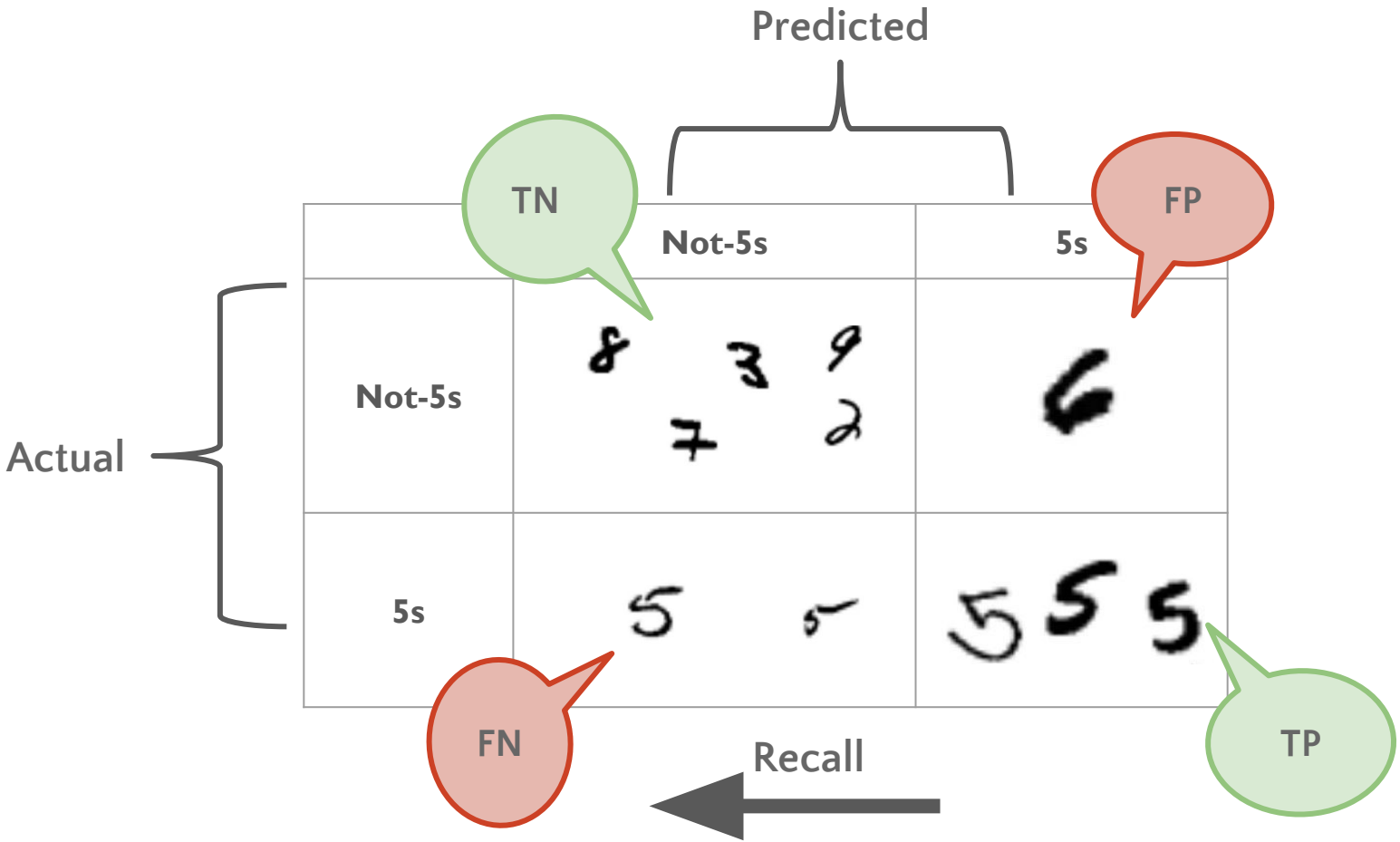
what proportion of images that were classified as 5s were actually 5

CLOUD x LAB

Performance Measures – Recall

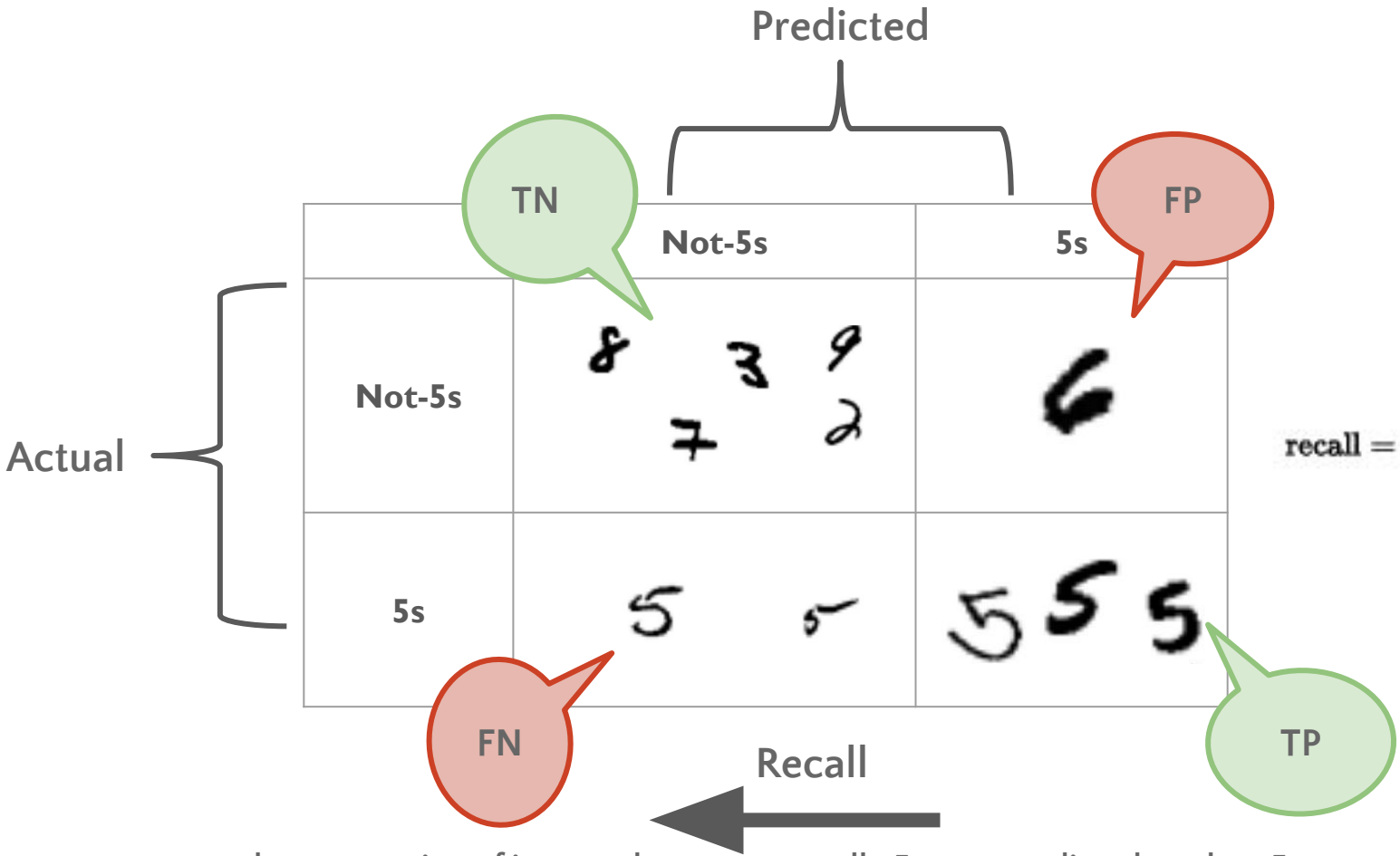
Recall means remember something learnt in past

CLOUD x LAB



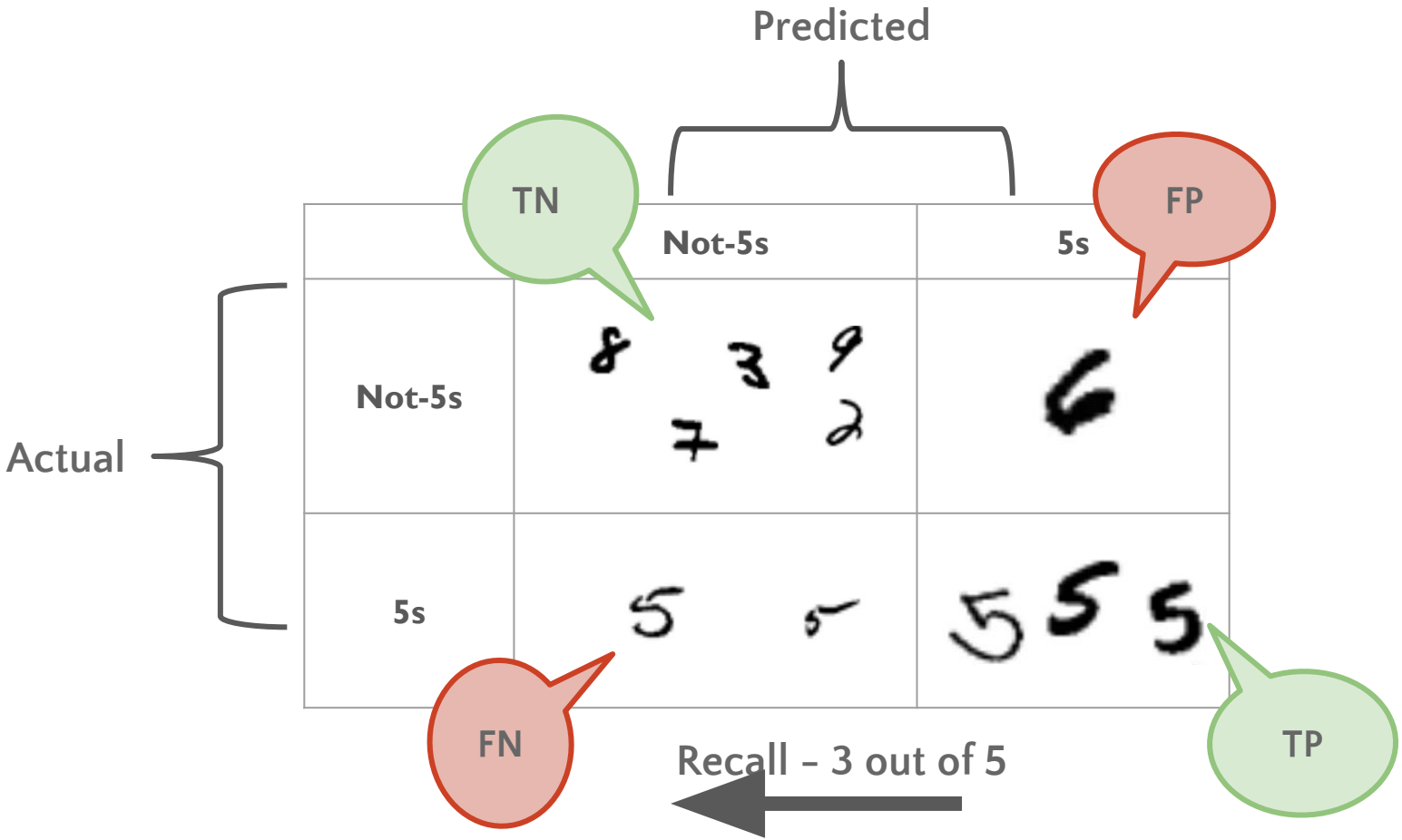
what proportion of images that were actually 5 were predicted as class 5

CLOUD x LAB



$$\text{recall} = \frac{TP}{TP + FN}$$

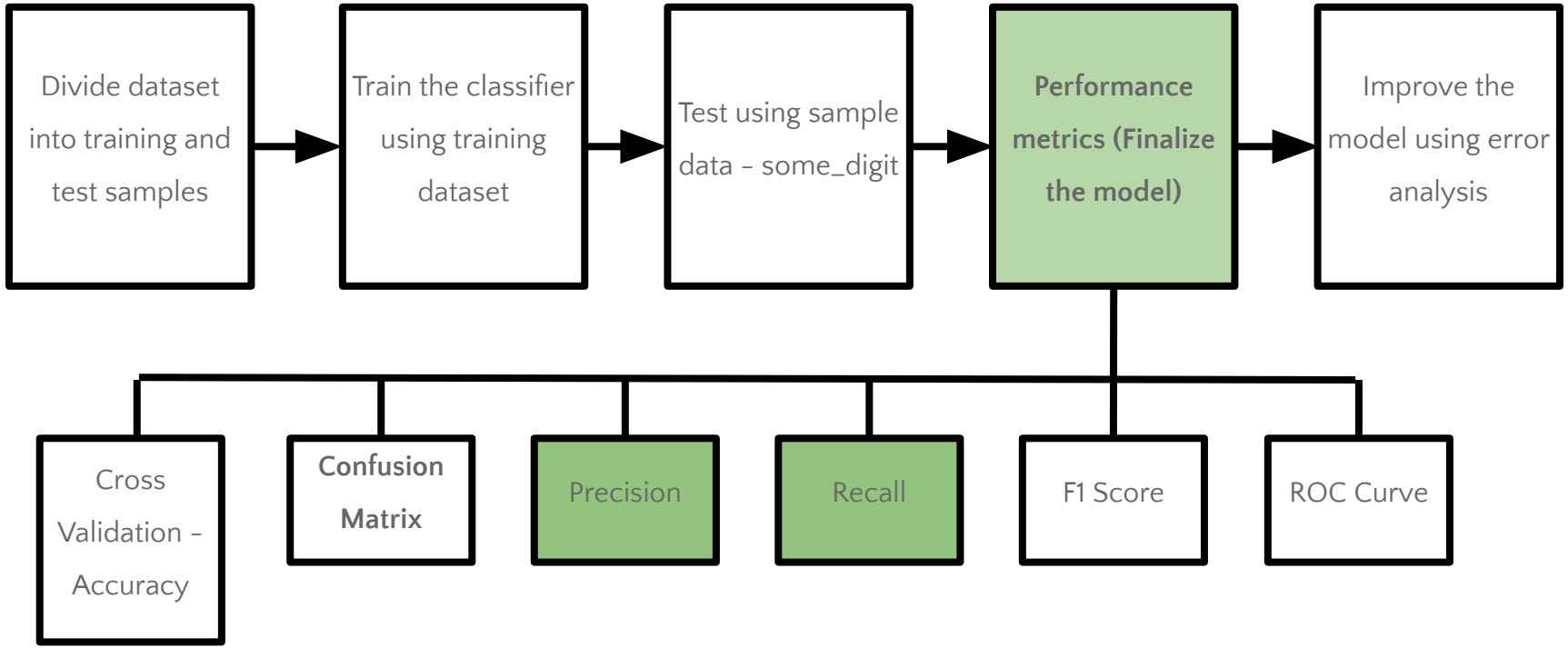
CLOUD x LAB



what proportion of images that were actually 5 were predicted as class 5

CLOUD × LAB

Steps

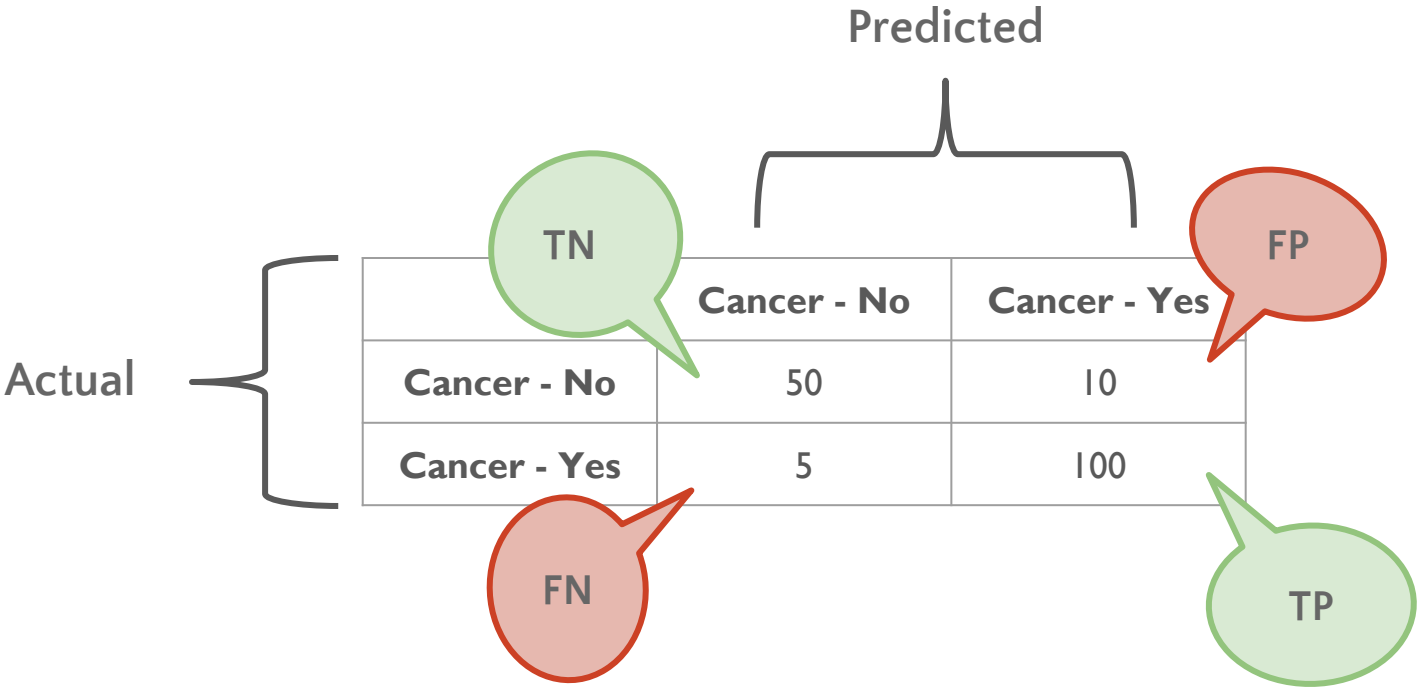


CLOUD x LAB

Let’s see one more example of confusion matrix of model predicting if someone has cancer or not

CLOUD x LAB

Performance Measures – Confusion Matrix



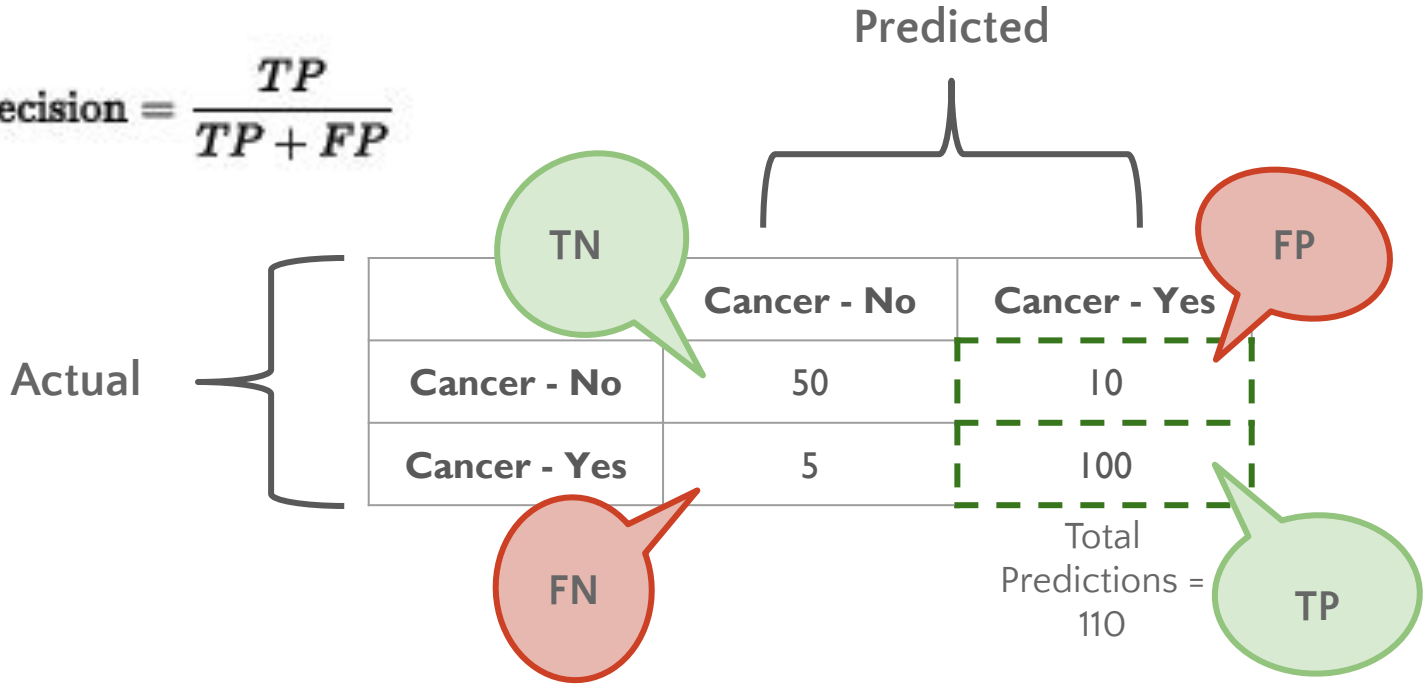
	Cancer - No	Cancer - Yes
Cancer - No	50	10
Cancer - Yes	5	100

What is Precision?

CLOUD x LAB

Performance Measures – Confusion Matrix

$$\text{precision} = \frac{TP}{TP + FP}$$

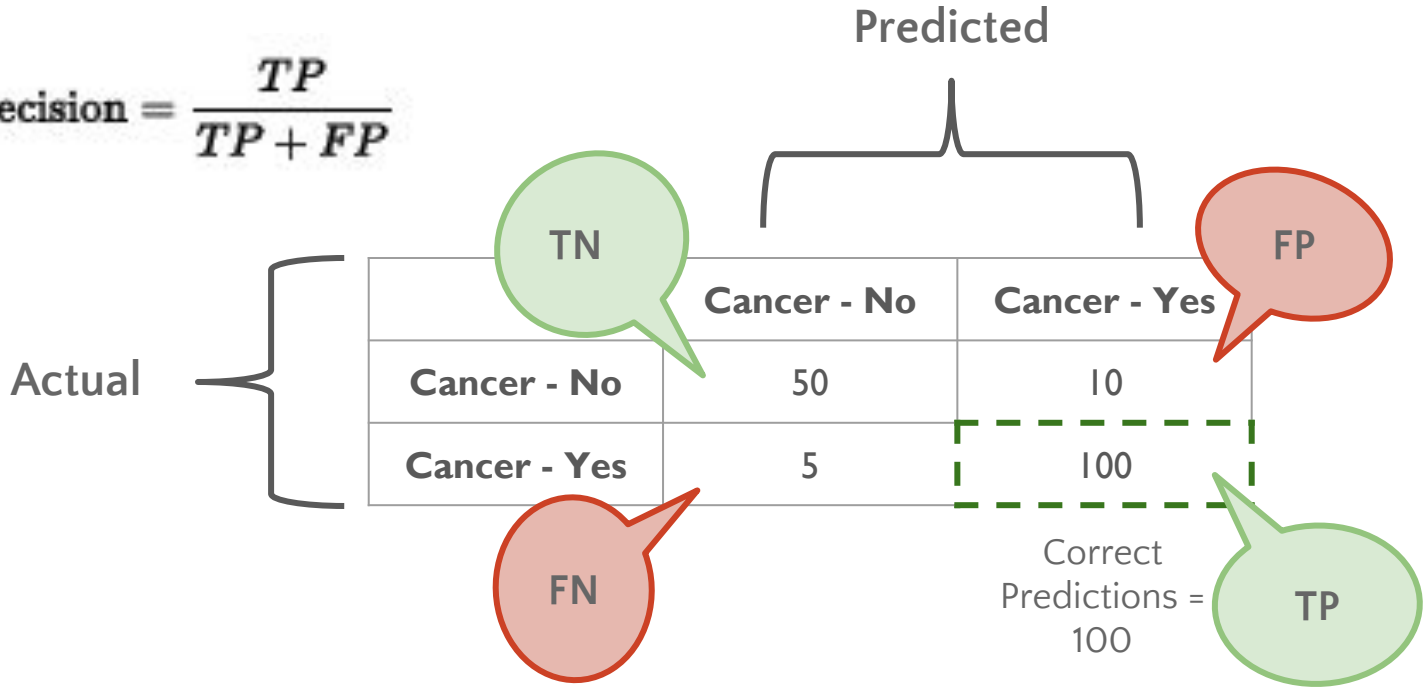


What is Precision?

CLOUD x LAB

Performance Measures – Confusion Matrix

$$\text{precision} = \frac{TP}{TP + FP}$$

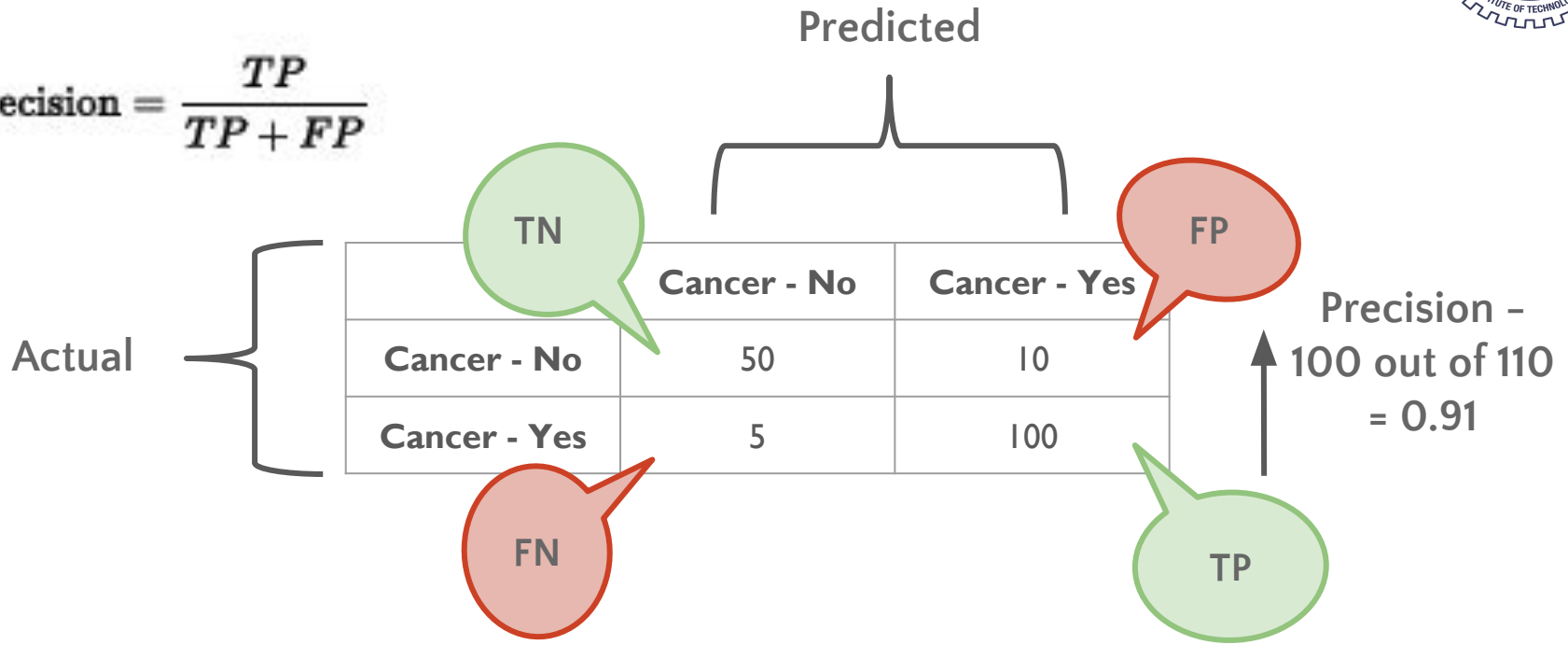


What is Precision?

CLOUD x LAB

Performance Measures – Confusion Matrix

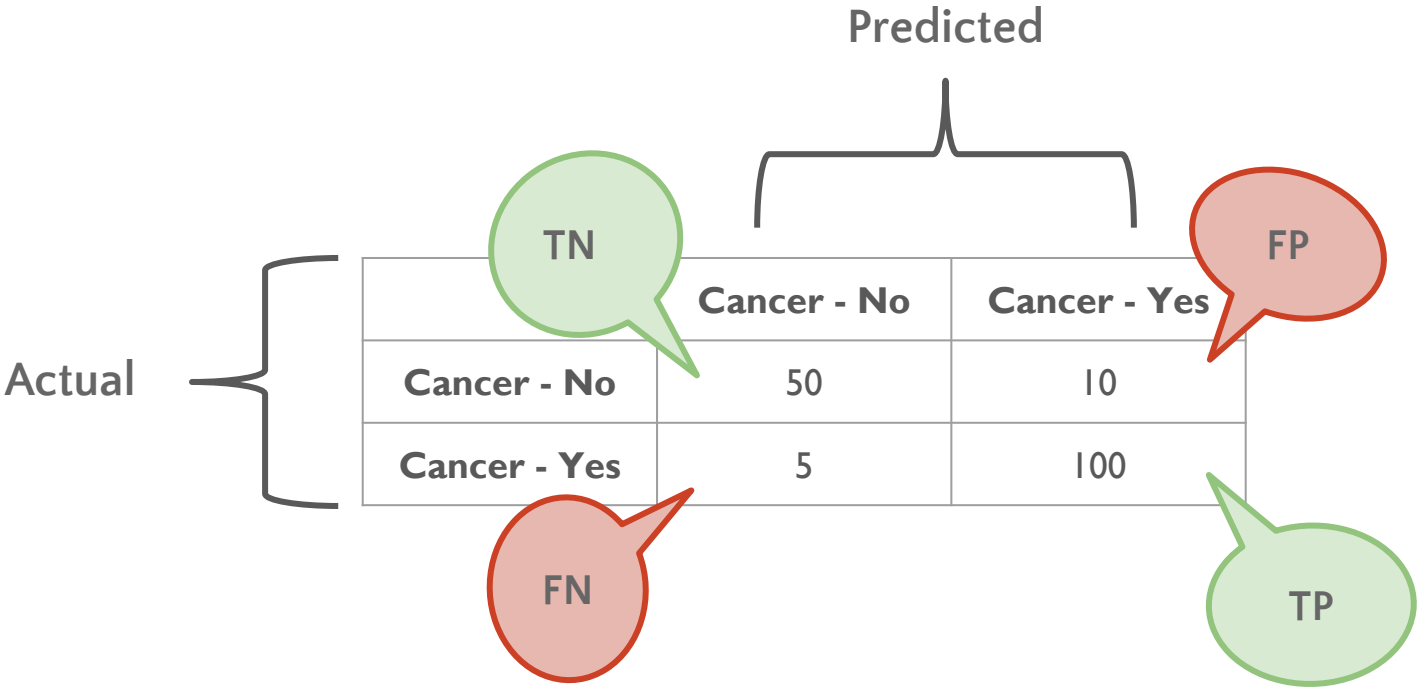
$$\text{precision} = \frac{TP}{TP + FP}$$



What is Precision?

CLOUD x LAB

Performance Measures – Confusion Matrix



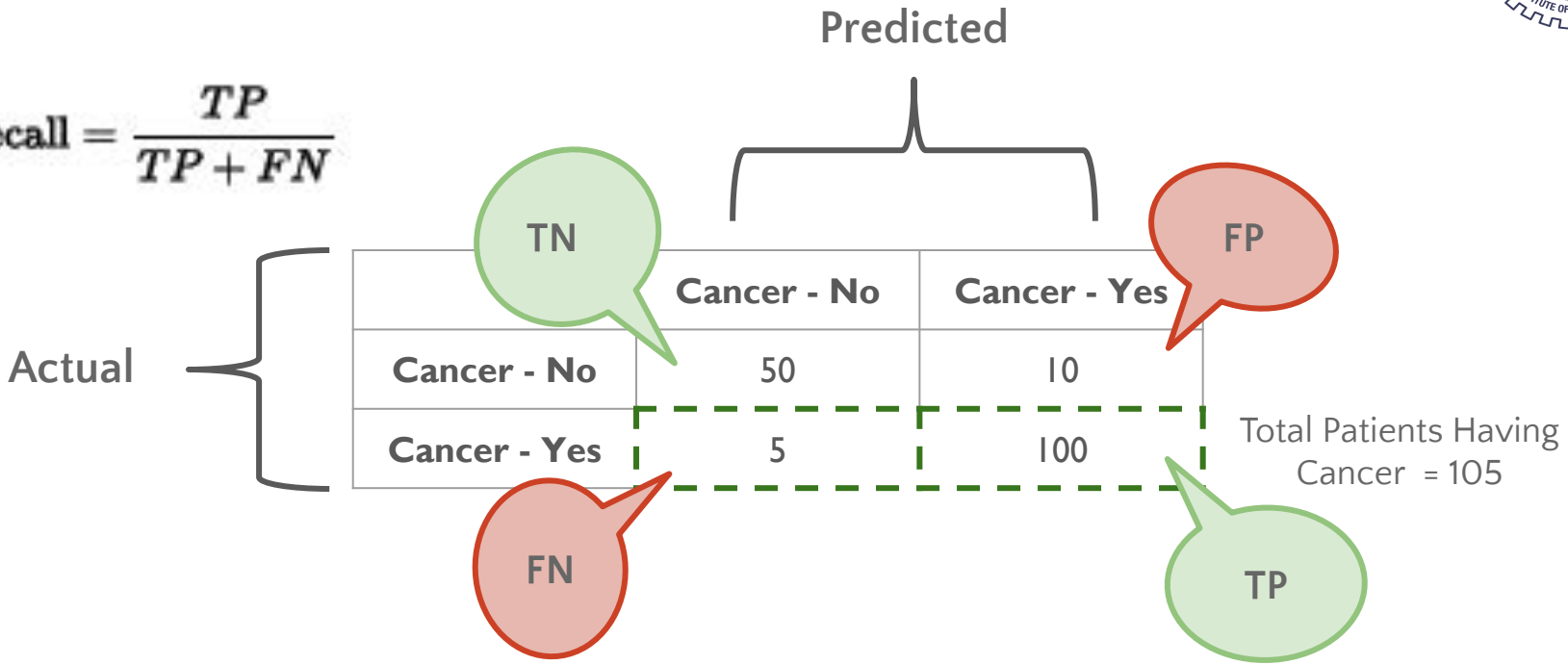
	Cancer - No	Cancer - Yes
Cancer - No	50	10
Cancer - Yes	5	100

What is Recall?

CLOUD x LAB

Performance Measures – Confusion Matrix

$$\text{recall} = \frac{TP}{TP + FN}$$

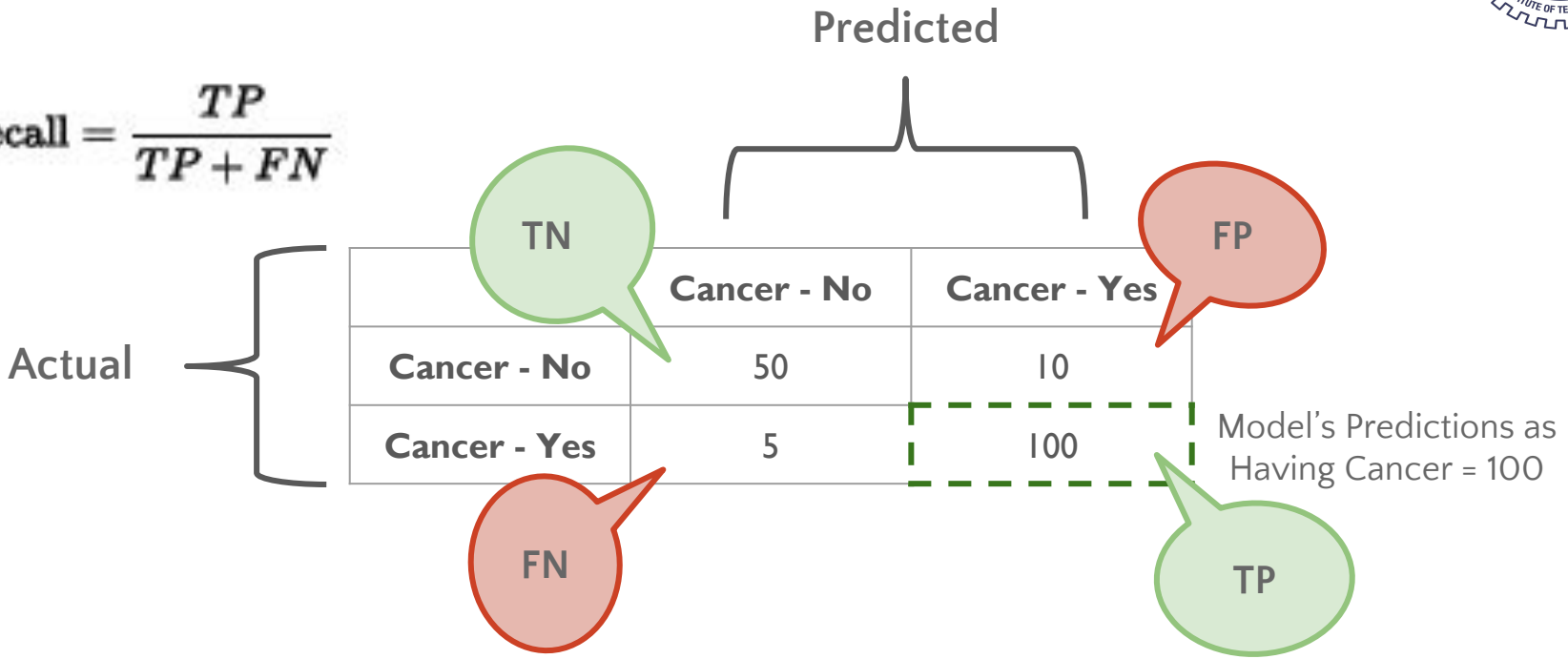


What is Recall?

CLOUD x LAB

Performance Measures – Confusion Matrix

$$\text{recall} = \frac{TP}{TP + FN}$$

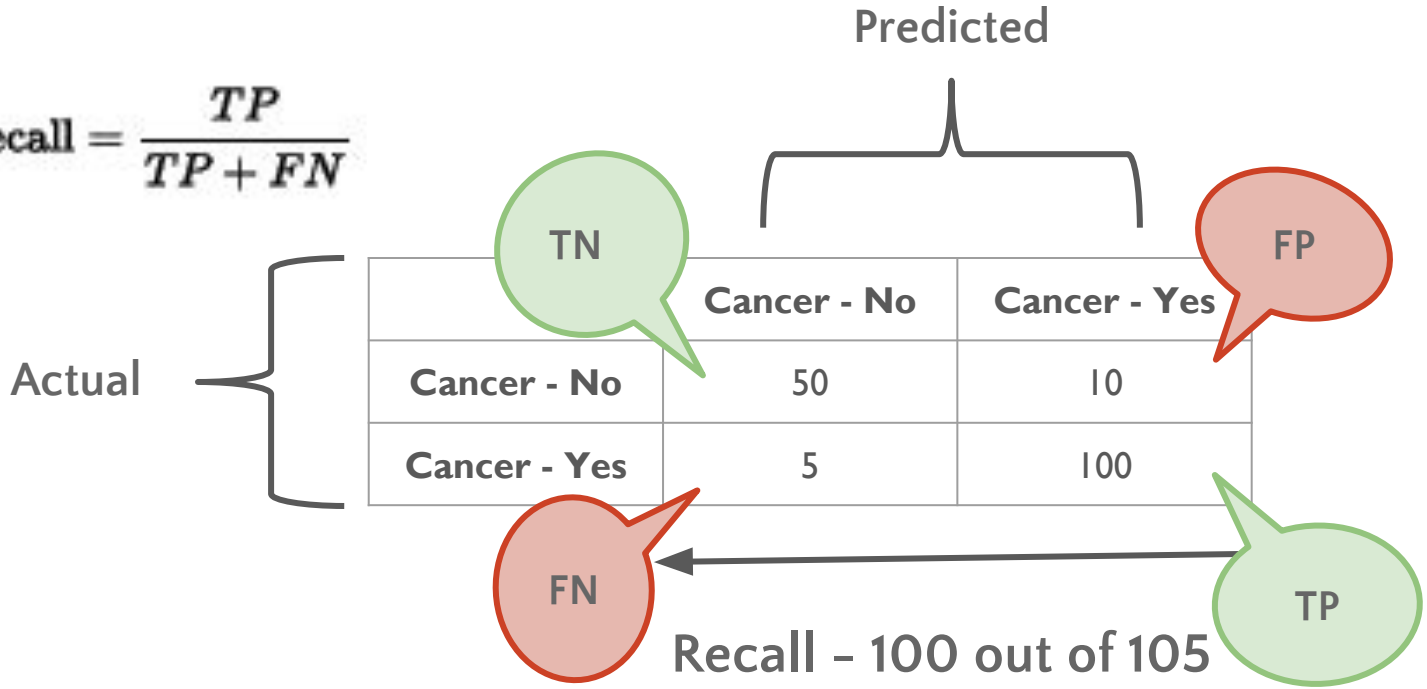


What is Recall?

CLOUD x LAB

Performance Measures – Confusion Matrix

$$\text{recall} = \frac{TP}{TP + FN}$$



What is Recall?

CLOUD x LAB

Performance Measures – Confusion Matrix

```
# Get the predicted values

>>> from sklearn.model_selection import cross_val_predict
>>> y_train_pred = cross_val_predict(sgd_clf, X_train, y_train_5, cv=3)


# Confusion matrix of 5-detector

>>> from sklearn.metrics import confusion_matrix
>>> confusion_matrix(y_train_5, y_train_pred)
```

```
array([[53892,   687],
       [ 1891,  3530]])
```

Confusion Matrix of 5-detector

Performance Measures – Confusion Matrix

```
# Get Precision Score of 5-detector

>>> from sklearn.metrics import precision_score, recall_score
>>> precision_score(y_train_5, y_train_pred)
```

Precision Score of 5-detector

Performance Measures – Confusion Matrix

```
# Get Recall Score of 5-detector

>>> recall_score(y_train_5, y_train_pred)
```

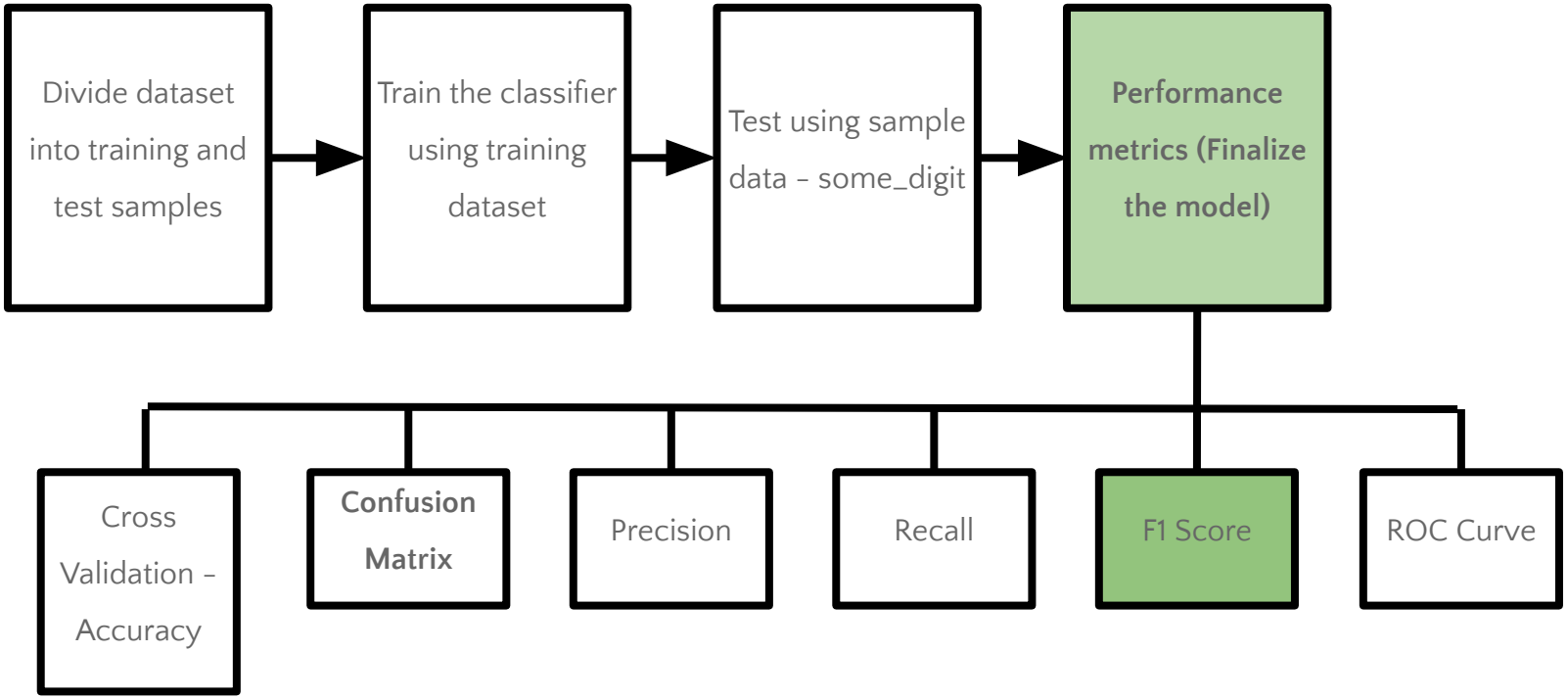
Recall Score of 5-detector

Performance Measures – Confusion Matrix

Precision = 83 %	Recall = 65 %
Correct only 83% of time	Detects only 65% of 5s

Precision / Recall Score of 5-detector

Steps



CLOUD x LAB

Performance Measures – F1 Score

- Instead of computing precision and recall every time
- We prefer a single metric which combines both precision and recall
- This single metric is f1 score

CLOUD x LAB

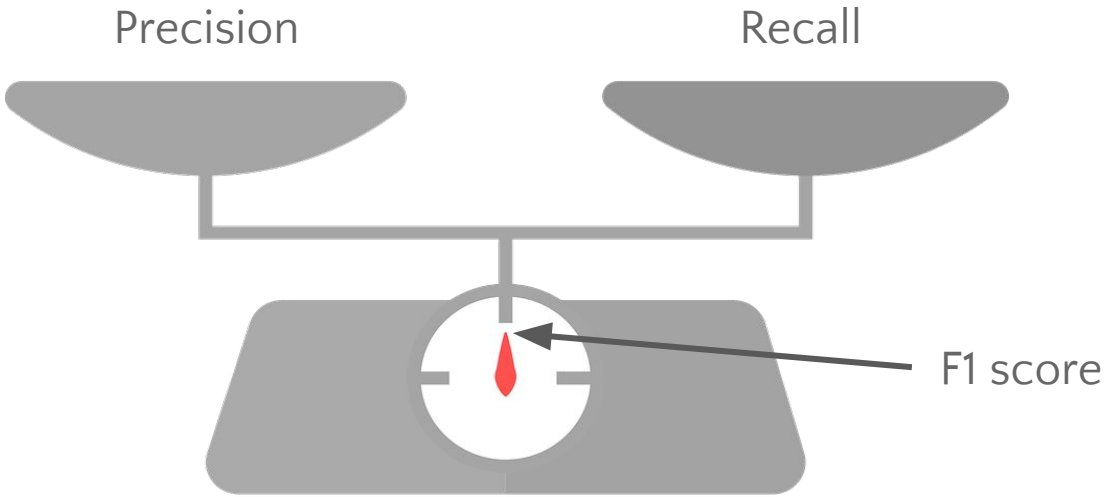
Performance Measures – F1 Score

- F1 score is harmonic mean of precision and recall

$$F_1 = \frac{2}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}}$$

CLOUD x LAB

Performance Measures – F1 Score



F1 score is high when both **Precision** and **Recall** are almost similar

CLOUD x LAB

Performance Measures – F1 Score

```
# Get f1 score of 5-detector

>>> from sklearn.metrics import f1_score
>>> f1_score(y_train_5, y_train_pred)
```

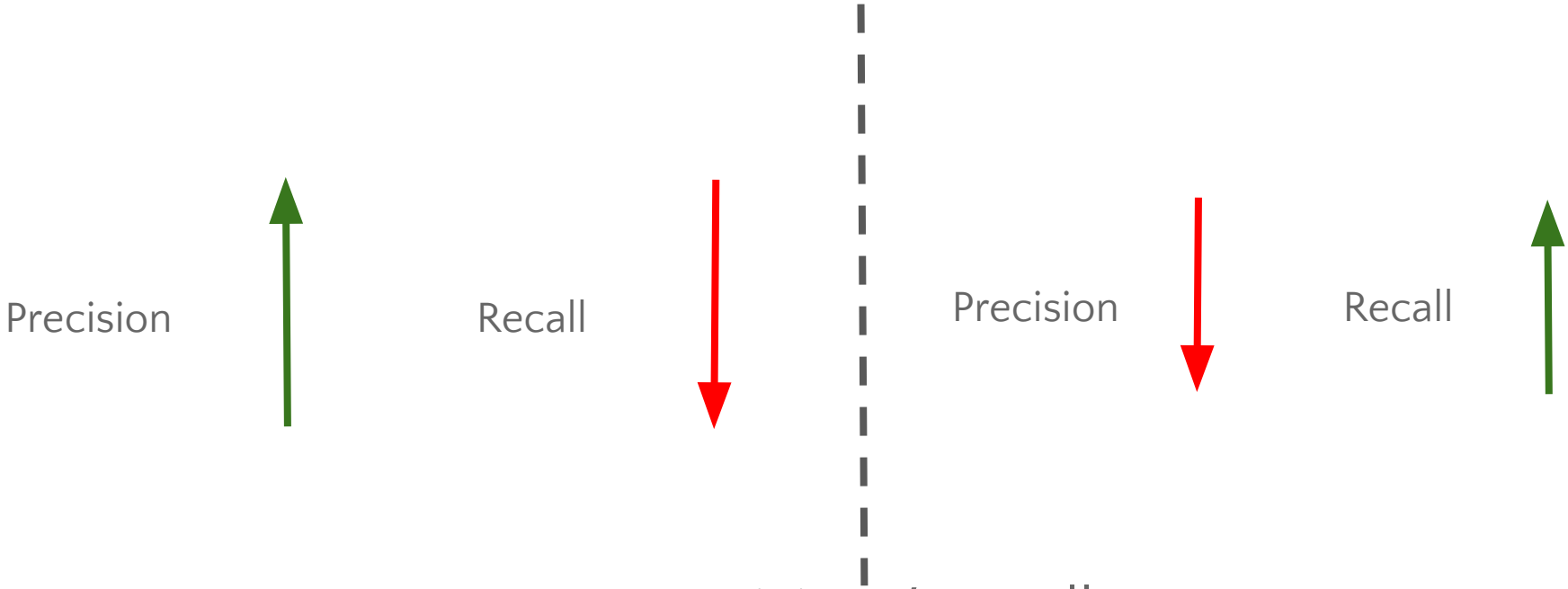
F1 Score of 5-detector

Performance Measures – Precision / Recall Tradeoff

Now you may think that we can have both high precision and high recall in a model. But unfortunately, we can't have both high precision and high recall at the same time.

CLOUD x LAB

Performance Measures – Precision / Recall Tradeoff



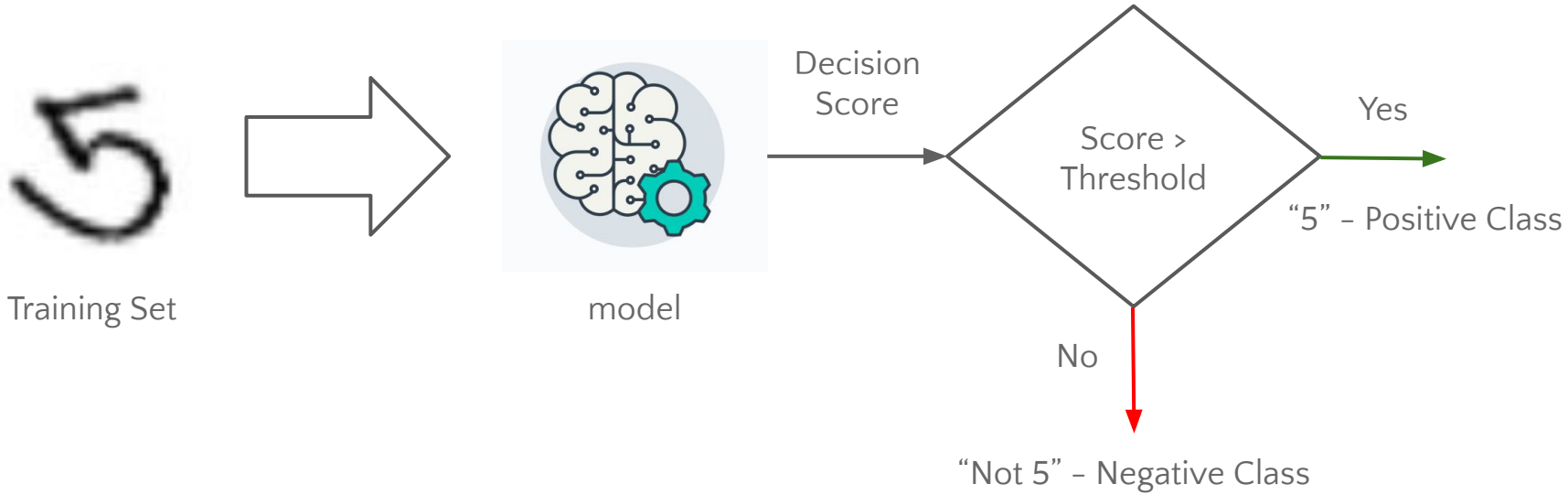
CLOUD x LAB

Performance Measures – Precision / Recall Tradeoff

To understand this tradeoff, lets see how SGDClassifier works

CLOUD x LAB

Performance Measures – Precision / Recall Tradeoff



Threshold – decided by the classification algorithm

CLOUD x LAB

Precision / Recall Tradeoff – Thresholds

- Scikit enables the user to get the scores from the classifier

Calculating the decision function score for the SGDClassifier

```
>>> y_scores = sgd_clf.decision_function([some_digit])
>>> y_scores
array([2164.22030239])
```

Precision / Recall Tradeoff

Increasing precision reduces recall and vice-versa
Raising the threshold, decreases recall

- For '5' and 'Not 5' classifier
  - For threshold = 0, classifier correctly classifies 5 as 5

```
>>> threshold = 0
>>> y_some_digit_pred = (y_scores > threshold)
>>> y_some_digit_pred
array([ True])
```

Run it on Notebook

CLOUD x LAB

Precision / Recall Tradeoff

Increasing precision reduces recall and vice-versa
Raising the threshold, decreases recall

- For '5' and 'Not 5' classifier
  - For threshold = 0, classifier correctly classifies 5 as 5
  - For threshold = 8000, classifier incorrectly classifies digit 5 as not 5

```
# Setting the threshold to 8000
>>> threshold = 8000
>>> y_some_digit_pred = (y_scores > threshold)
>>> y_some_digit_pred
array([False])
```

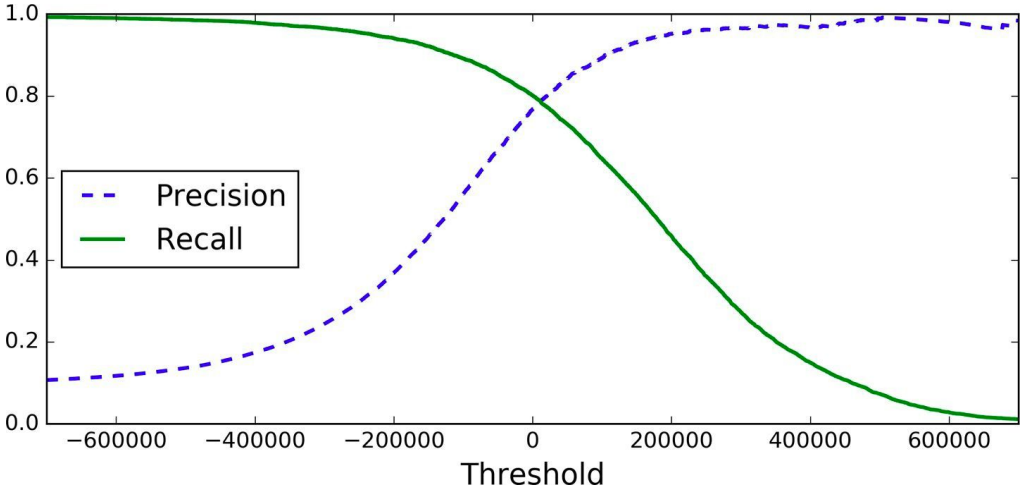
Run it on Notebook

CLOUD x LAB

Review of Precision/ Recall Tradeoff

- Hence, by selecting an appropriate threshold, the user can obtain the desired precision. However, the best precision may not have the best recall.

Precision / Recall Curve

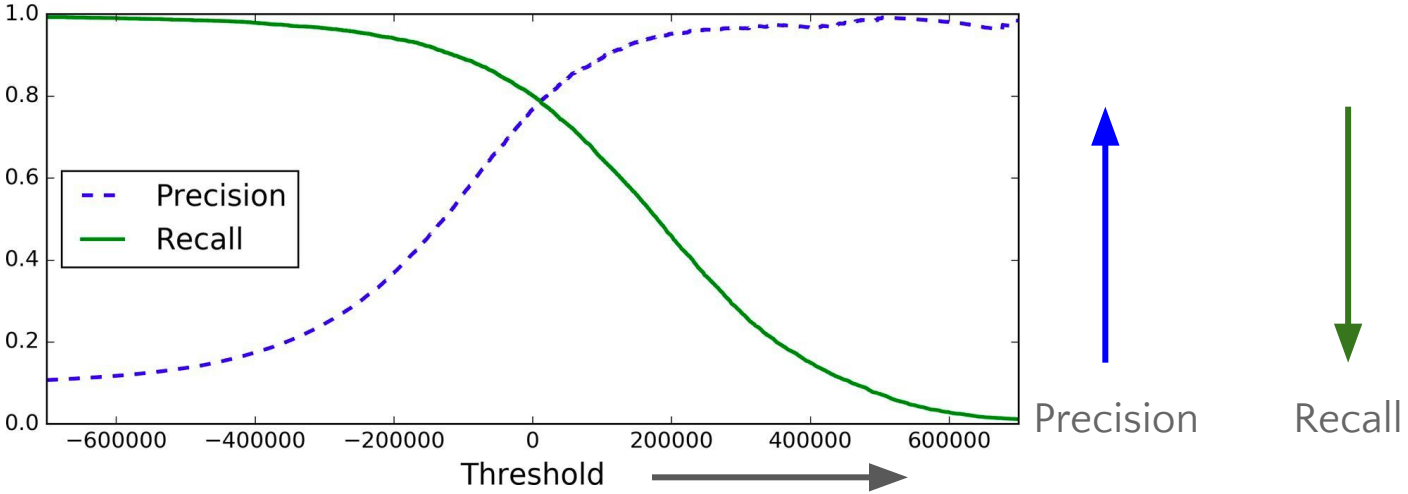


Precision and recall versus the decision threshold

Precision Recall Curve of 5-detector

CLOUD x LAB

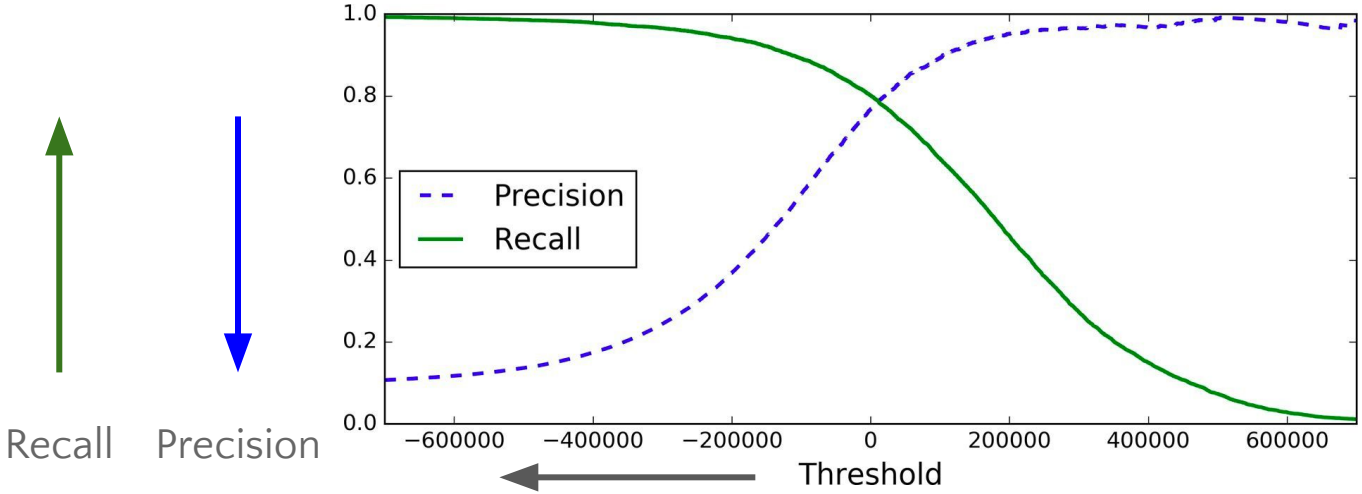
Precision / Recall Curve



Precision and recall versus the decision threshold

CLOUD x LAB

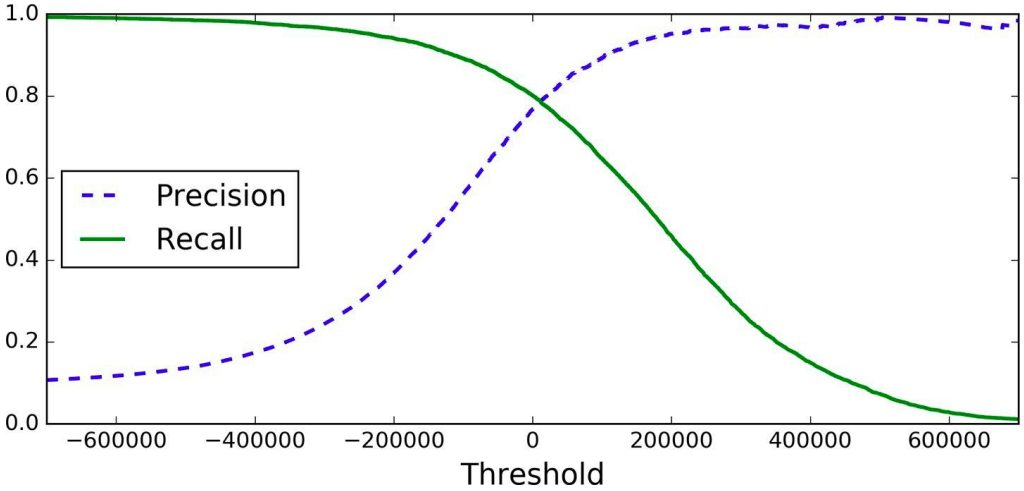
Precision / Recall Curve



Precision and recall versus the decision threshold

CLOUD x LAB

Precision / Recall Curve

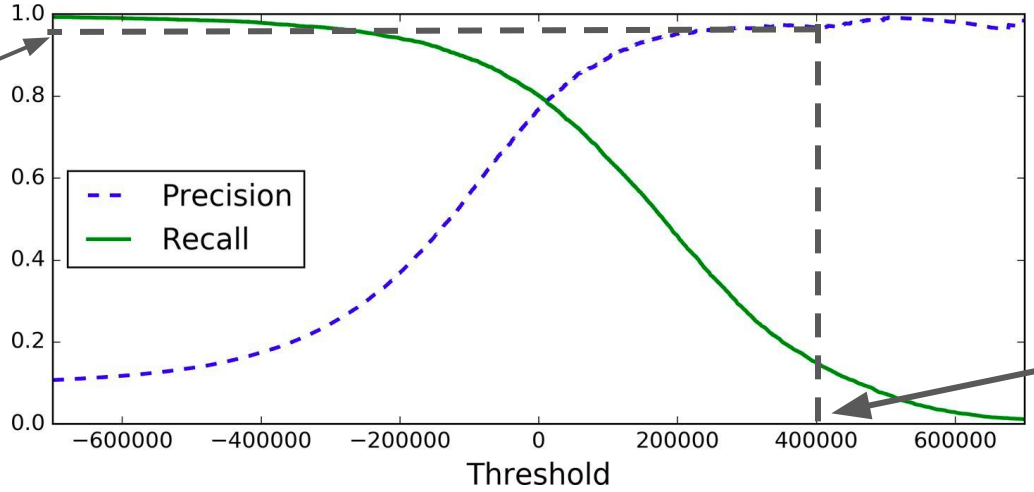


Which threshold value to use?

CLOUD x LAB

Precision / Recall Curve

90% Precision



Set threshold to 400000

Select the threshold to achieve 90% precision

Precision / Recall Curve Using Scikit-learn

Plot Precision / Recall Curve

- Get the scores of all the training dataset using `cross_val_predict` with `decision_function` as function
- Compute the precision and recall for all possible thresholds using `precision_recall_curve()`
- Plot both precision and recall for the thresholds using matplotlib
- Select the threshold value that gives the best precision/ recall tradeoff to the task at hand.

Precision / Recall Curve Using Scikit-learn

- Calculating precision/ recall curve using Scikit-Learn

```
>>> y_scores = cross_val_predict(sgd_clf, X_train, y_train_5, cv=3,
method="decision_function")

>>> from sklearn.metrics import precision_recall_curve
precisions, recalls, thresholds = precision_recall_curve(y_train_5,
y_scores)
```

Run it on Notebook

Precision / Recall Curve Using Scikit-learn

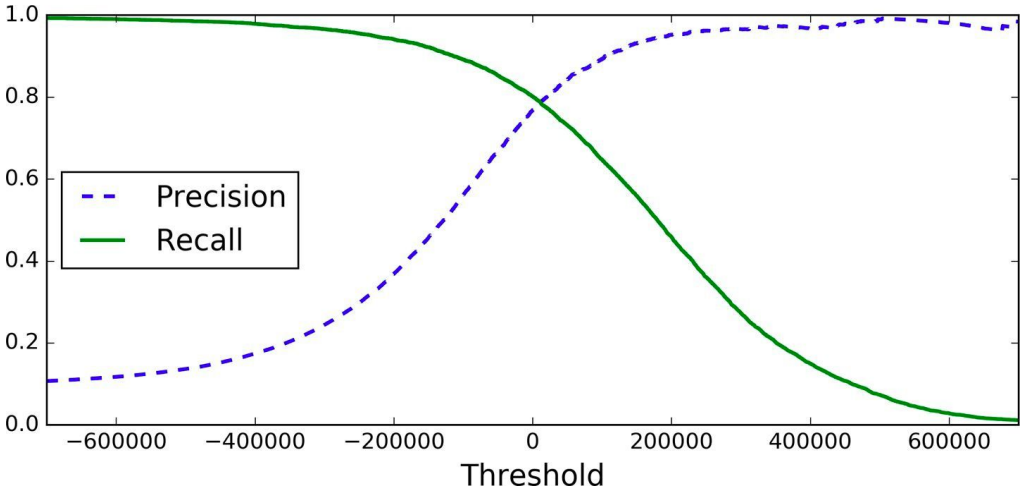
- Plotting precision/ recall curve using Scikit-Learn

```
>>> def plot_precision_recall_vs_threshold(precisions, recalls, thresholds):
    plt.plot(thresholds, precisions[:-1], "b--", label="Precision", linewidth=2)
    plt.plot(thresholds, recalls[:-1], "g-", label="Recall", linewidth=2)
    plt.legend(loc="center right", fontsize=16)
    plt.xlabel("Threshold", fontsize=16)
    plt.grid(True)
    plt.axis([-50000, 50000, 0, 1])

plt.figure(figsize=(8, 4))

>>> plot_precision_recall_vs_threshold(precisions, recalls, thresholds)
>>> plt.show()
```

Run it on Notebook

Precision / Recall Curve

- Plotting precision/ recall curve using Scikit-Learn

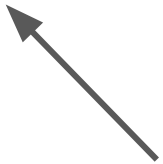


CLOUD x LAB

Building model with desired precision

- So let's say you want to build a classifier with 90% precision
  - Then first select the threshold value which gives you 90% precision
  - Then build classifier using this threshold

```
>>> threshold_90_precision = thresholds[np.argmax(precisions >= 0.90)]
>>> y_train_pred_90 = (y_scores >= threshold_90_precision)
```



This classifier will give 90% precision

Run it on Notebook

Building model with desired precision

- Verify it

```
>>> precision_score(y_train_5, y_train_pred_90)
0.9000380083618396

>>> recall_score(y_train_5, y_train_pred_90)
0.4368197749492714
```

Run it on Notebook

Pause for Questions

Boss

I want a model with
99% precision

ML Engineer

CLOUD x LAB

Boss

At what recall? :)

ML Engineer

CLOUD x LAB

Say we have to build a model which detects if a video is safe for kids or not.

Question – High Precision or High Recall?

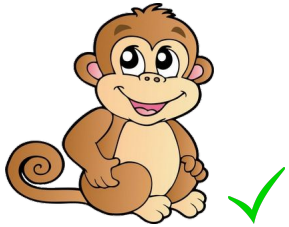
High Precision or High Recall?

High precision means if the model classifies video 4 and video 6 as safe for kids, they are actually safe for kids.

1 – Not Safe

2 – Safe

3 – Not Safe



4 – Safe

5 – Not Safe

6 – Safe

High **Precision**

CLOUD x LAB

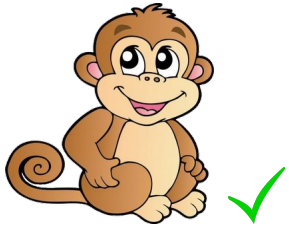
High Precision or High Recall?

In high precision, we are okay if the model is not able to classify video 2 as safe for kids but whichever videos it classifies as safe for kids they are actually safe.

1 – Not Safe

2 – Safe

3 – Not Safe



4 – Safe

5 – Not Safe

6 – Safe

High **Precision**

CLOUD x LAB

High Precision or High Recall?

We would prefer a model which has high precision and low recall. It is okay if the model rejects many good videos but keeps only really safe ones

1 – Not Safe

2 – Safe

3 – Not Safe

4 – Safe ✓

5 – Not Safe

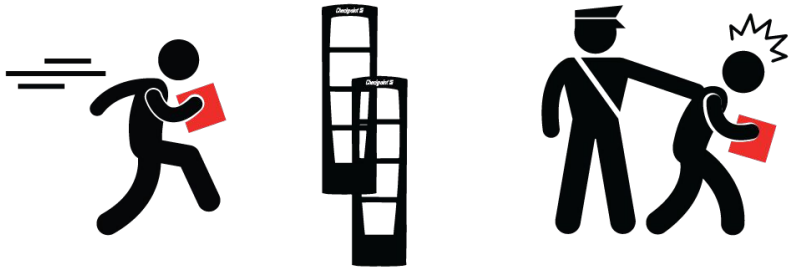
6 – Safe ✓

High **Precision**, Low **Recall** ✓

CLOUD x LAB

Say we have to build a model which detects shoplifters on the basis of surveillance image. In case, someone is marked as shoplifter, we manually examine.

Question – High Precision or High Recall?



High Precision or High Recall?

We would prefer the model to have high recall even if the precision is low because our goal is to catch almost all the shoplifters.

High **Recall**

CLOUD x LAB

High Precision or High Recall?

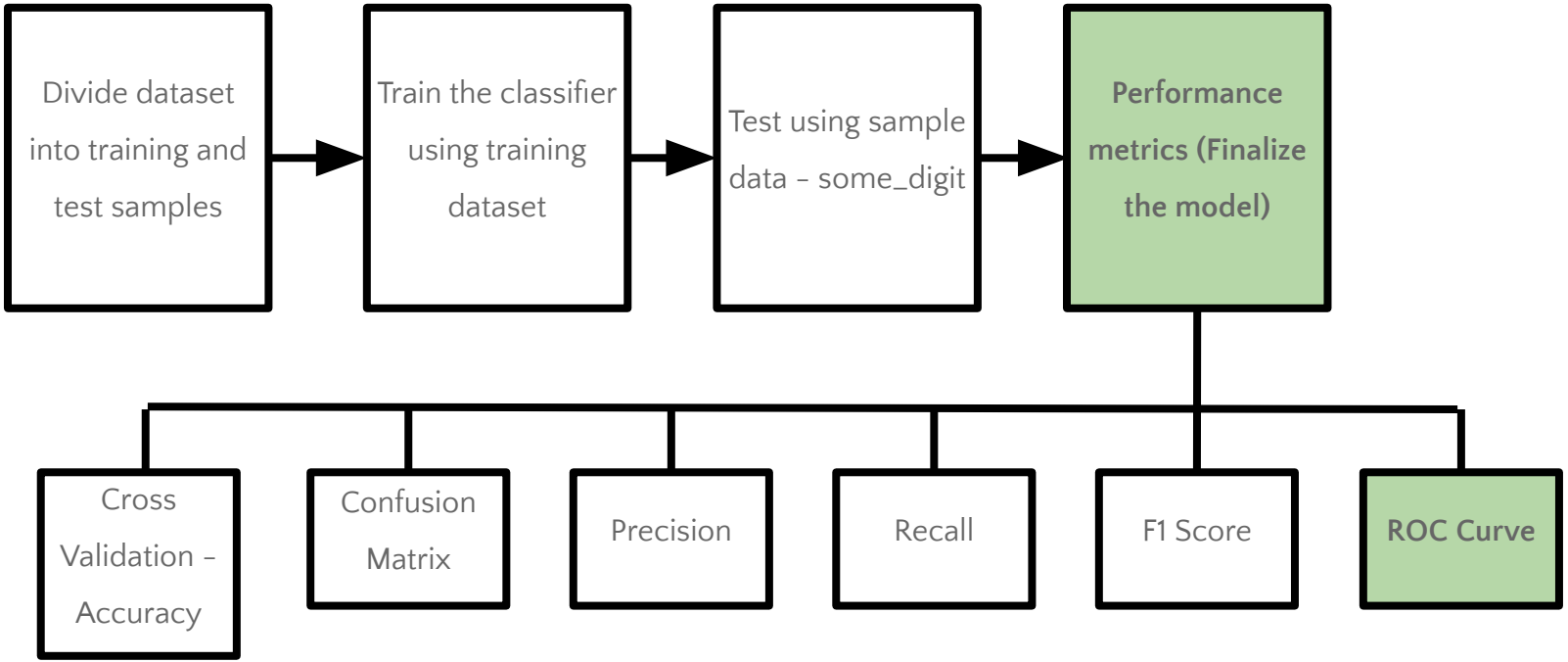
In the high recall, the security guard might catch and examine some non shoplifters also but we will achieve our goal of catching almost all the shoplifters.

High **Recall**

CLOUD x LAB

Pause for Questions

Steps



CLOUD x LAB

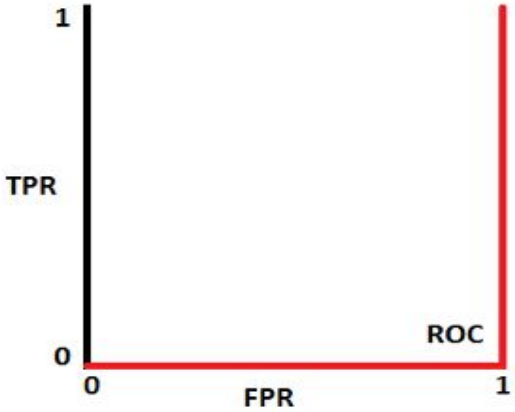
The ROC (Receiver Operating Characteristic) Curve

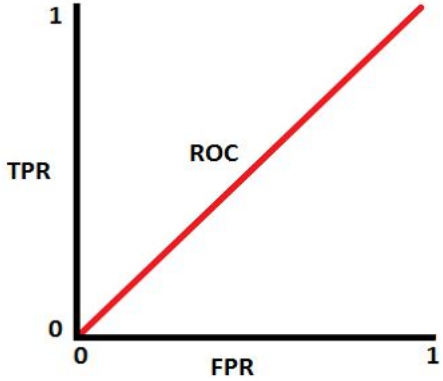
CLOUD x LAB

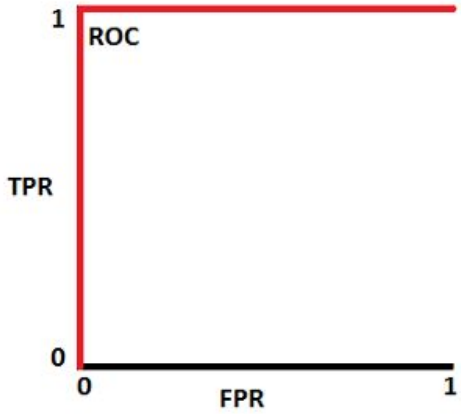
Performance Measures – ROC Curve

- Is helpful in comparing performances of different models
- Provides a tradeoff between true positives and false positives like precision/recall curve
- ROC is a probability curve and AUC represents degree or measure of separability.
- Let see ROC Curve of two different models
    - SGD Classifier
    - Random Forest Classifier
- Before that lets learn about “Area Under the Curve”

Performance Measures – ROC Curve





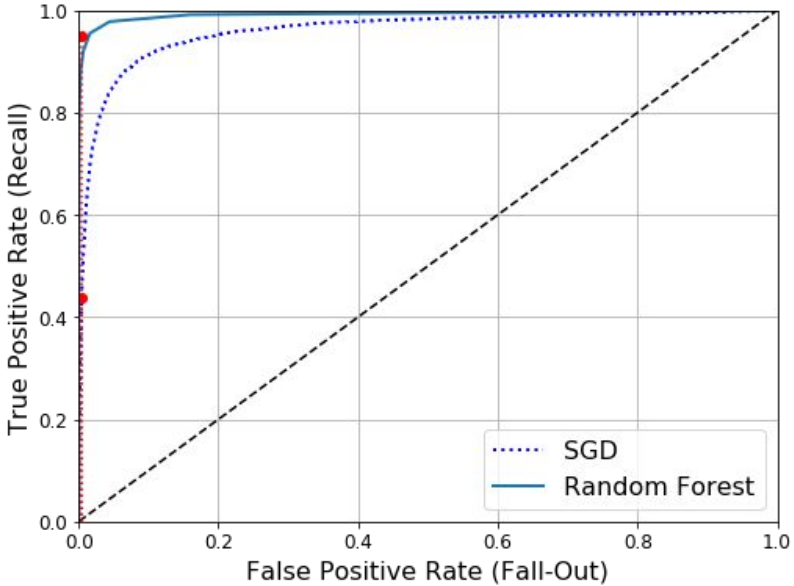


- A model is reciprocating the classes is represented at the point (1,0), ROC travels from bottom left to bottom right and, then to the top right and has an AUC of 0.
- A model with no skill is represented at the point (0.5, 0.5), ROC travels by a diagonal line from the bottom left of the plot to the top right and has an AUC of 0.5.
- Perfect skill Model is represented at a point (0,1), ROC travels from bottom left to top left, then across the top to the top right and has an AUC of 1.

CLOUD x LAB

Performance Measures – ROC Curve

Area Under the Curve is higher for Random Forest, so its better model than SGD

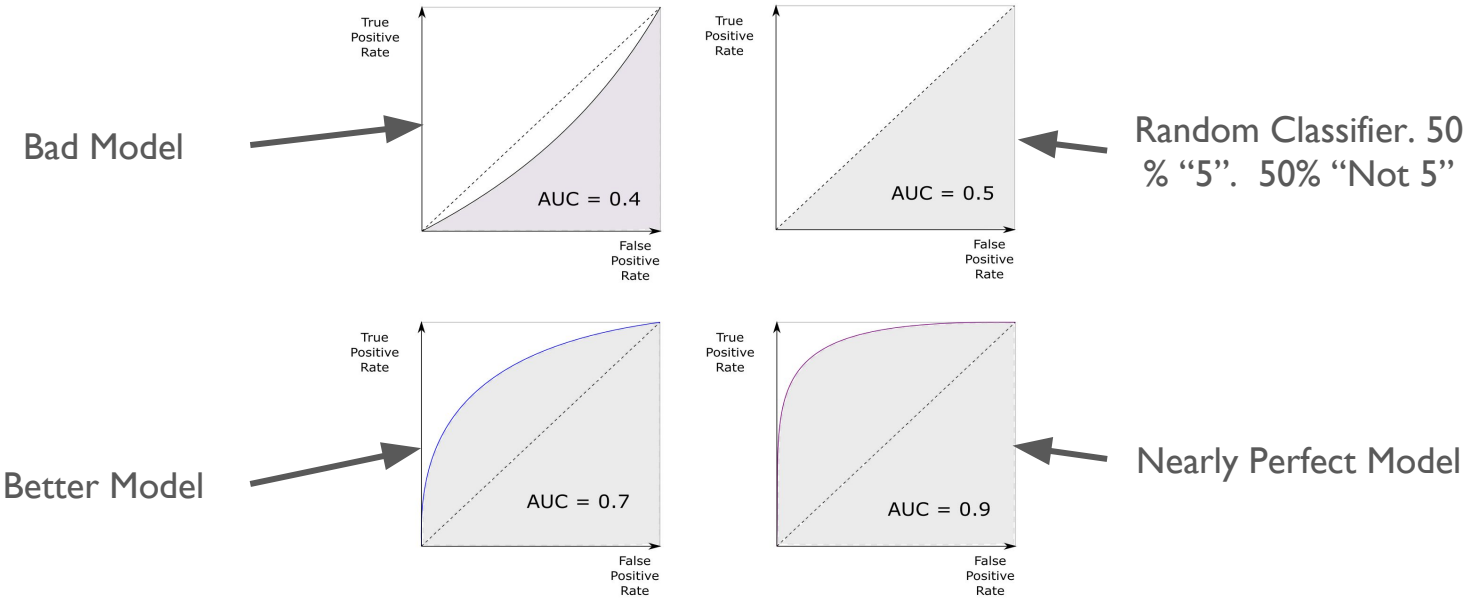


ROC Curve of 5 Detector for SGD and Random Forest

CLOUD × LAB

Performance Measures – ROC Curve

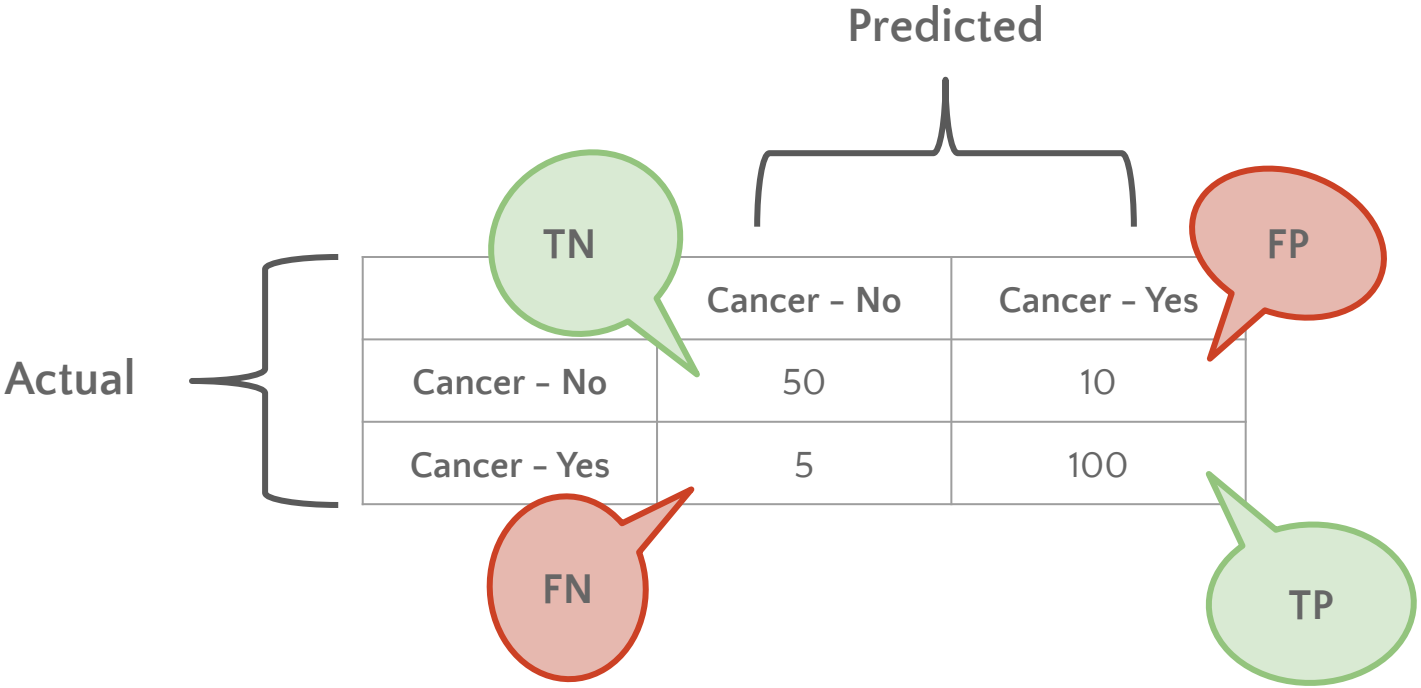
Higher the area under the curve, better the model is



AUC - Area Under the Curve

CLOUD x LAB

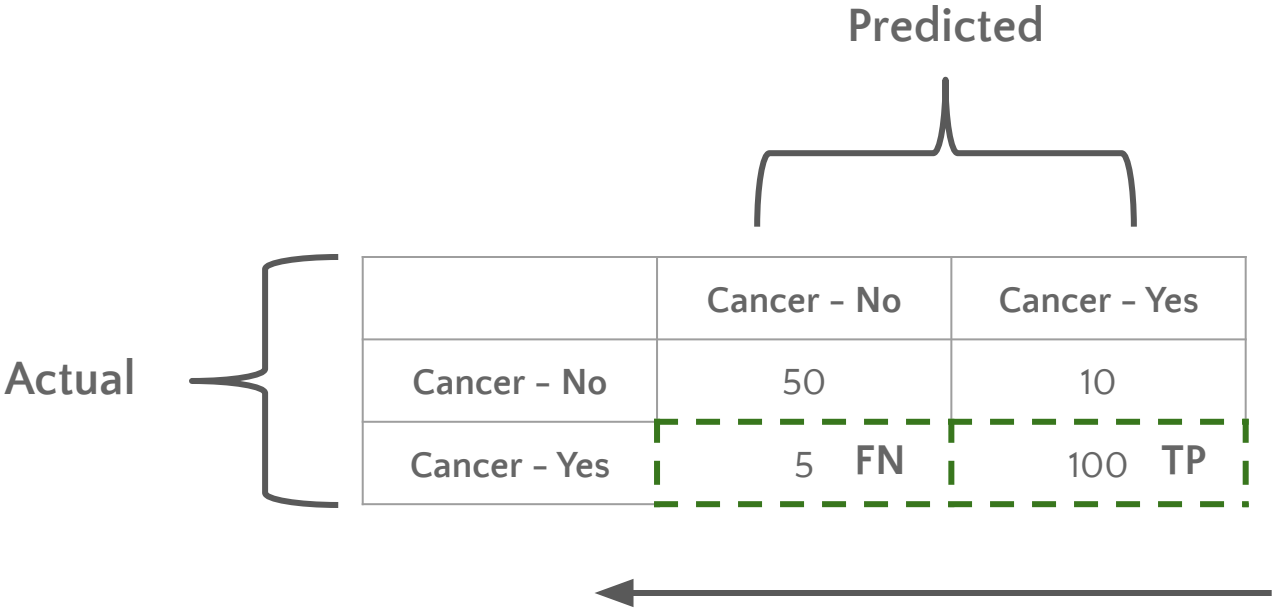
Performance Measures – ROC Curve



	Cancer – No	Cancer – Yes
Cancer – No	50	10
Cancer – Yes	5	100

CLOUD × LAB

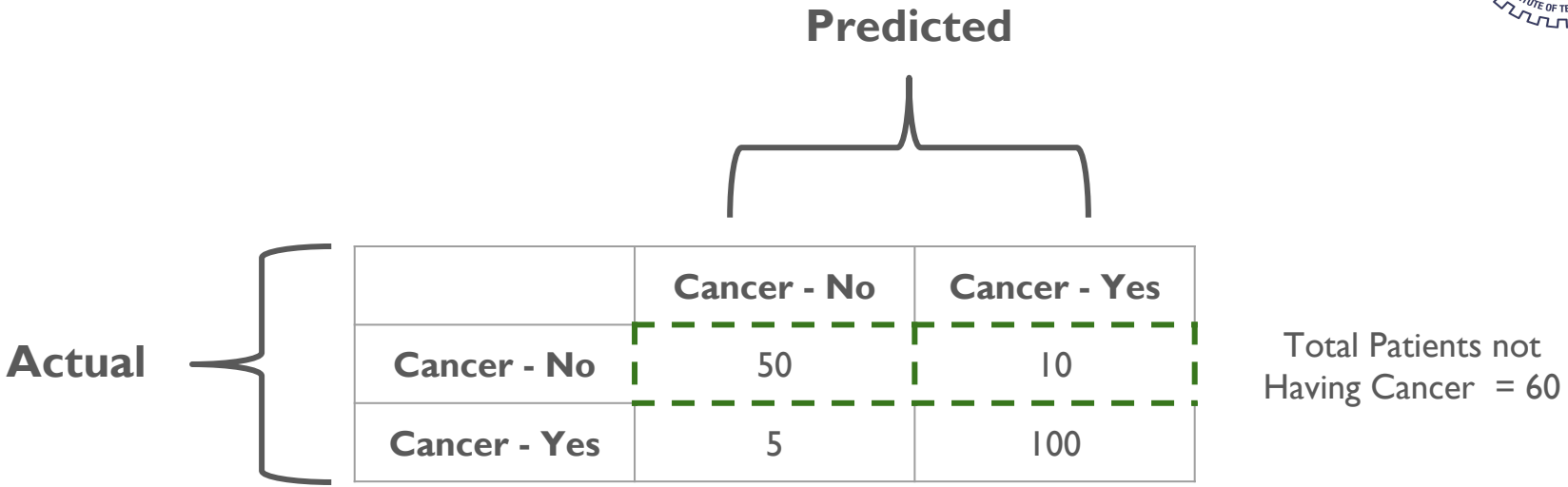
Performance Measures – ROC Curve



Recall or True Positive Rate = 100 out of 105 = 0.95

CLOUD × LAB

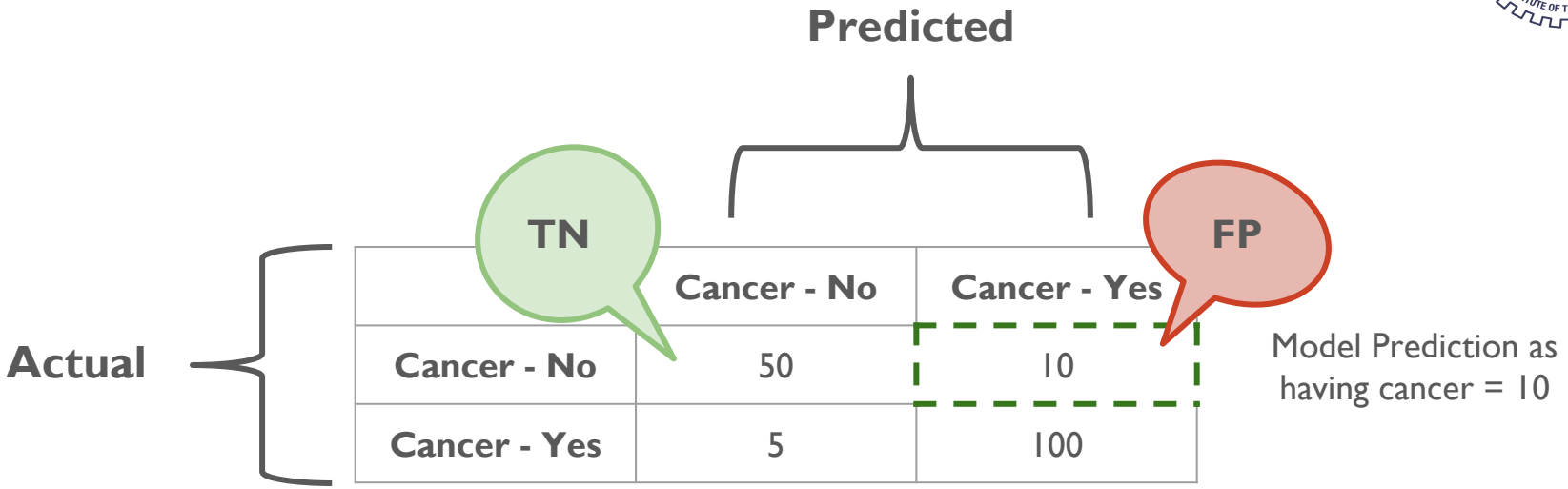
Performance Measures – ROC Curve



False Positive Rate tells us what proportion of patients that actually were not having cancer are classified as having cancer by the model.

CLOUD x LAB

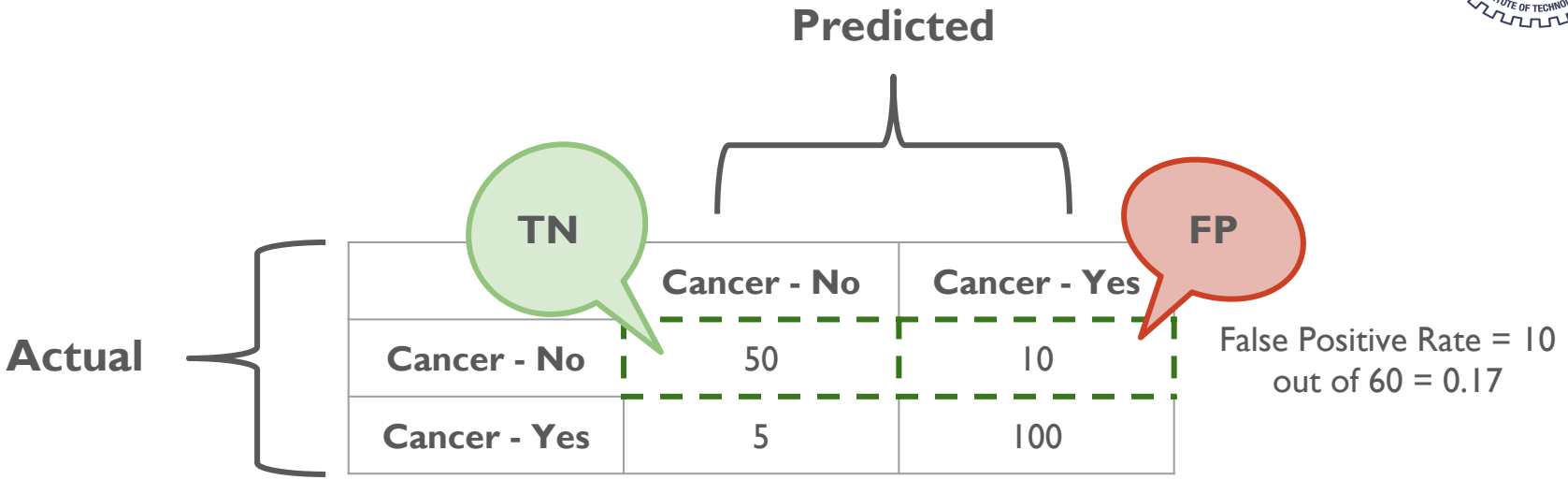
Performance Measures – ROC Curve



False Positive Rate tells us what proportion of patients that actually were not having cancer are classified as having cancer by the model.

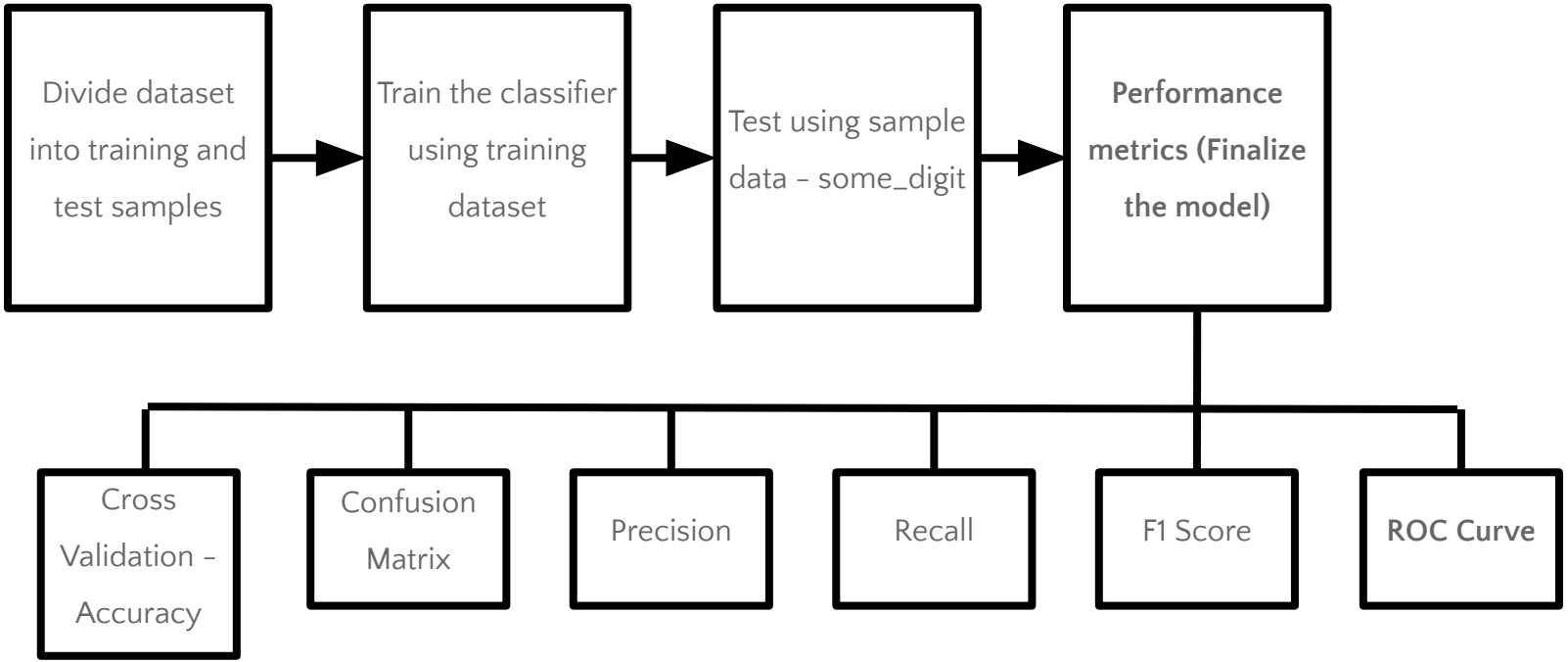
CLOUD x LAB

Performance Measures – ROC Curve



False Positive Rate tells us what proportion of patients that actually were not having cancer are classified as having cancer by the model.

CLOUD x LAB


Divide dataset into training and test samples
Train the classifier using training dataset
Test using sample data – some_digit
Performance metrics (Finalize the model)
Cross Validation – Accuracy
Confusion Matrix
Precision
Recall
F1 Score
ROC Curve

CLOUD x LAB

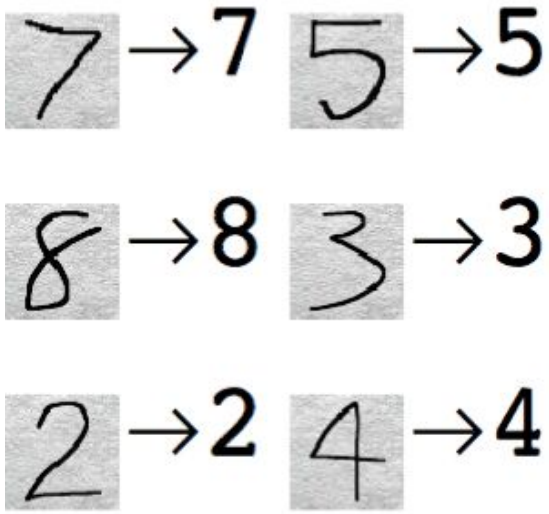
Multiclass Classification

So far we have discussed 5-detector. Let's come back to the original problem

CLOUD x LAB

Multiclass Classification

Classify Handwritten Digits – Which is a MultiClass Problem



CLOUD x LAB

Multiclass Classification

Multiclass Classifier Using SGDClassifier

```
>>> sgd_clf.fit(X_train, y_train)
>>> sgd_clf.predict([some_digit])
```

Multiclass Classification

Check Accuracy Using Cross Validation

```
>>> cross_val_score(sgd_clf, X_train, y_train, cv=3,
scoring="accuracy")
```


CLOUD x LAB

Multiclass Classification

Improving Accuracy by Scaling the Features

```
>>> from sklearn.preprocessing import StandardScaler
>>> scaler = StandardScaler()
>>> X_train_scaled = scaler.fit_transform(X_train.astype(np.float64))
>>> cross_val_score(sgd_clf, X_train_scaled, y_train, cv=3,
scoring="accuracy")
```

What do we do next?

Completed:

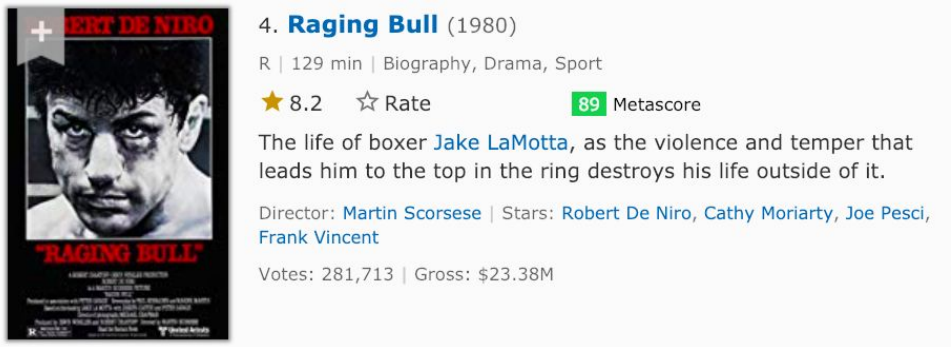
- Binary Classification
- Multi-class Classification

Next:

- Multi-label Classification
- Multi-output Classification

Multilabel Classification

Output multiple classes



Multiple Classes

BioGraphy	1
Drama	1
Sport	1
Sci-fi	0

Categories in which model is trained

CLOUD x LAB

Multilabel Classification

Output multiple classes

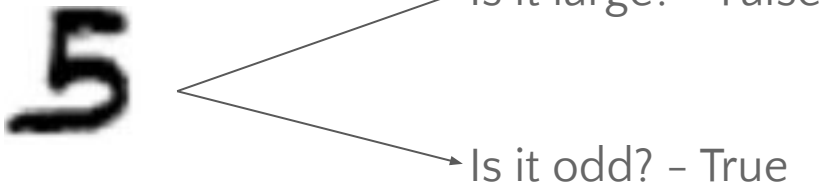
Multiple Classes

BioGraphy	0
Drama	1
Sport	0
Sci-fi	0

Categories in which model is trained

Multilabel Classification

- Build a model to classify an image into two classes
  - Greater than or equal to 7
  - Is Odd

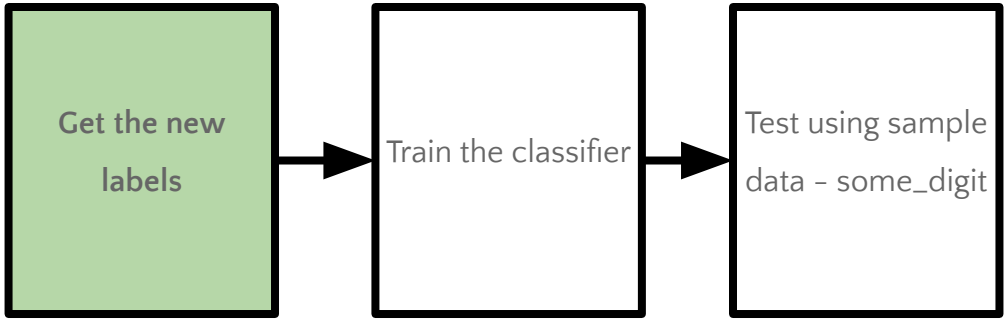


Classifier's output will be - [False, True]

CLOUD x LAB

Multilabel Classification - Example

- Classification of image into whether [large, odd], how do we do it?



CLOUD x LAB

Multilabel Classification – Example

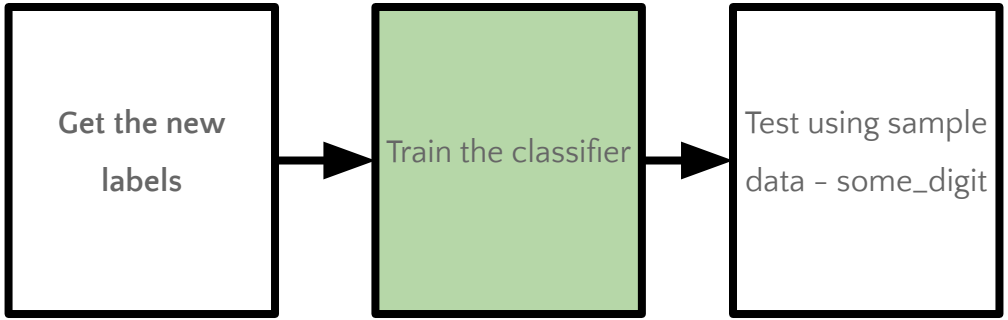
- Step 1: Get the new labels

```
>>> y_train_large = (y_train >= 7)
>>> y_train_odd = (y_train % 2 == 1)
>>> y_multilabel = np.c_[y_train_large, y_train_odd]

# np.c_ is used to concatenate the two arrays element wise
```

Run it on Notebook

Multilabel Classification

- Step 2: Train the model
  - KNeighbors Classifier supports multi-label, classification



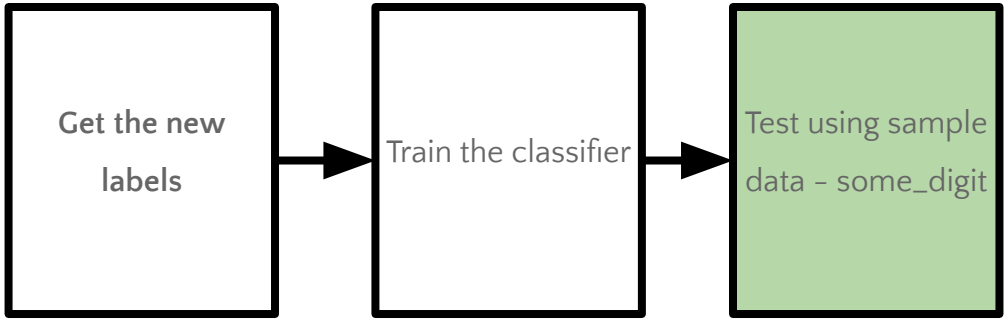

```
>>> from sklearn.neighbors import KNeighborsClassifier
>>> knn_clf = KNeighborsClassifier()
>>> knn_clf.fit(X_train, y_multilabel)
```

Run it on Notebook

CLOUD x LAB

Multilabel Classification

- Step 3: Test it




```
>>> knn_clf.predict([some_digit])
array([[False, True]], dtype=bool)
```

- Digit 5 is not large (i.e. it not 7 or 8 or 9) but is odd

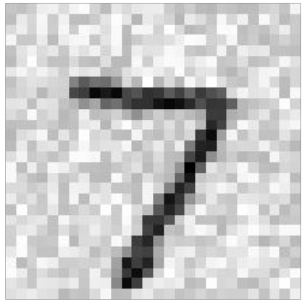
CLOUD x LAB

Multioutput Classification

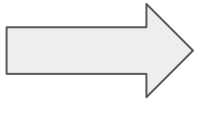
- It is simply a generalization of multilabel classification where each label can be multiclass (i.e., it can have more than two possible values)
- Multi-output classification example - **Removing noise from images**

Multioutput Classification

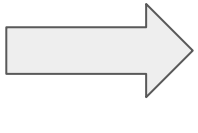
- We'll build a model that removes noise from images. It will take as input a noisy digit image, and it will output a clean digit image, represented as an array of pixel intensities, just like the MNIST images.

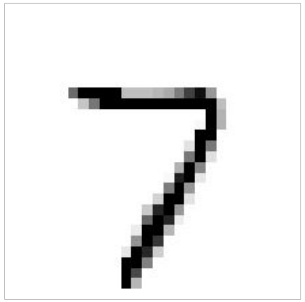


Input Image with Noise



Model



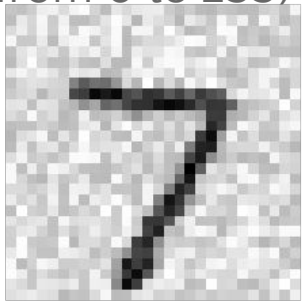


Cleaned Image

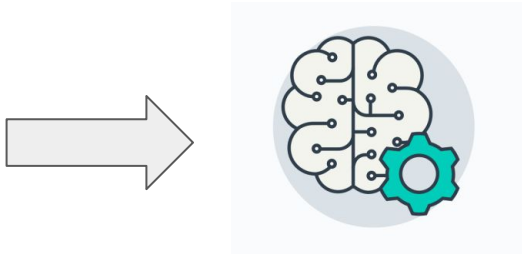
CLOUD x LAB

Multioutput Classification

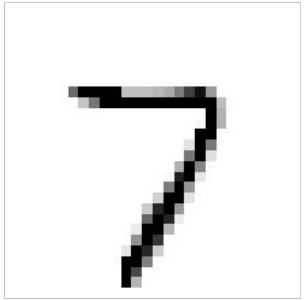
- Notice that the models' output is
  - **Multilabel (one label per pixel) - so 784 labels //we want values for all labels to be predicted together**
  - **Multiclass**: And each label can have **multiple values** (pixel intensity ranges from 0 to 255) - 256 classes.



Input Image with Noise



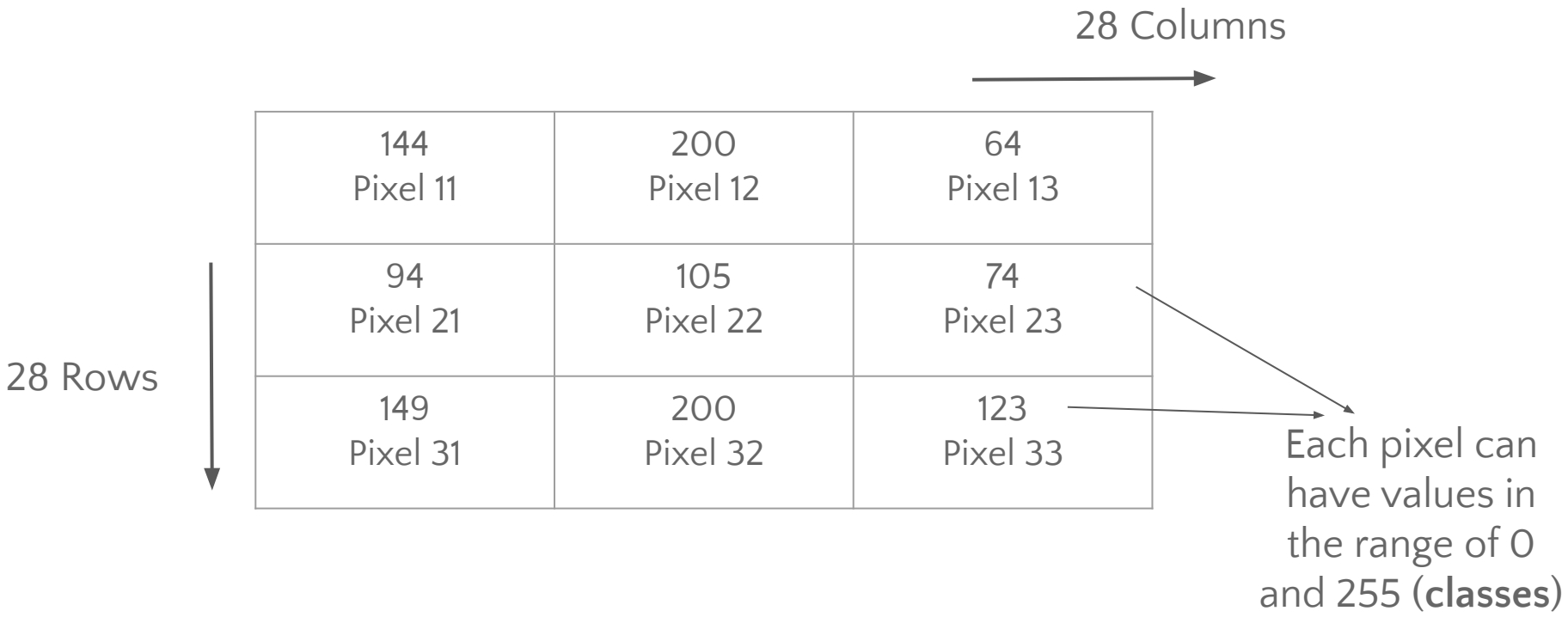
Model



Cleaned Image

CLOUD x LAB

Multioutput Classification



CLOUD x LAB

Multioutput Classification

Hands-on - image noise removal

- Adding noise to the training set #to make our data noisy

```
>>> import numpy.random as rnd
>>> noise_train = rnd.randint(0, 100, (len(X_train), 784))
>>> X_train_mod = X_train + noise_train
```

Run it on Notebook

Multioutput Classification

Hands-on - image noise removal

- Adding noise to the test set

```
>>> noise_test = rnd.randint(0, 100, (len(X_test), 784))
>>> X_test_mod = X_test + noise_test
```

Run it on Notebook

Multioutput Classification

Hands-on - image noise removal

- Setting clean image as the label (y_train and y_test)

```
>>> y_train_mod = X_train
>>> y_test_mod = X_test
```

Run it on Notebook

Multioutput Classification

Hands-on - image noise removal

- Let's view the noise image at index 0

```
>>> some_index = 0
>>> plot_digit(X_test_mod[some_index])
```

Run it on Notebook

Multioutput Classification

Hands-on - image noise removal

- Clean the image using the KNN Classifier

```
>>> knn_clf.fit(X_train_mod, y_train_mod)
```

Run it on Notebook

Multioutput Classification

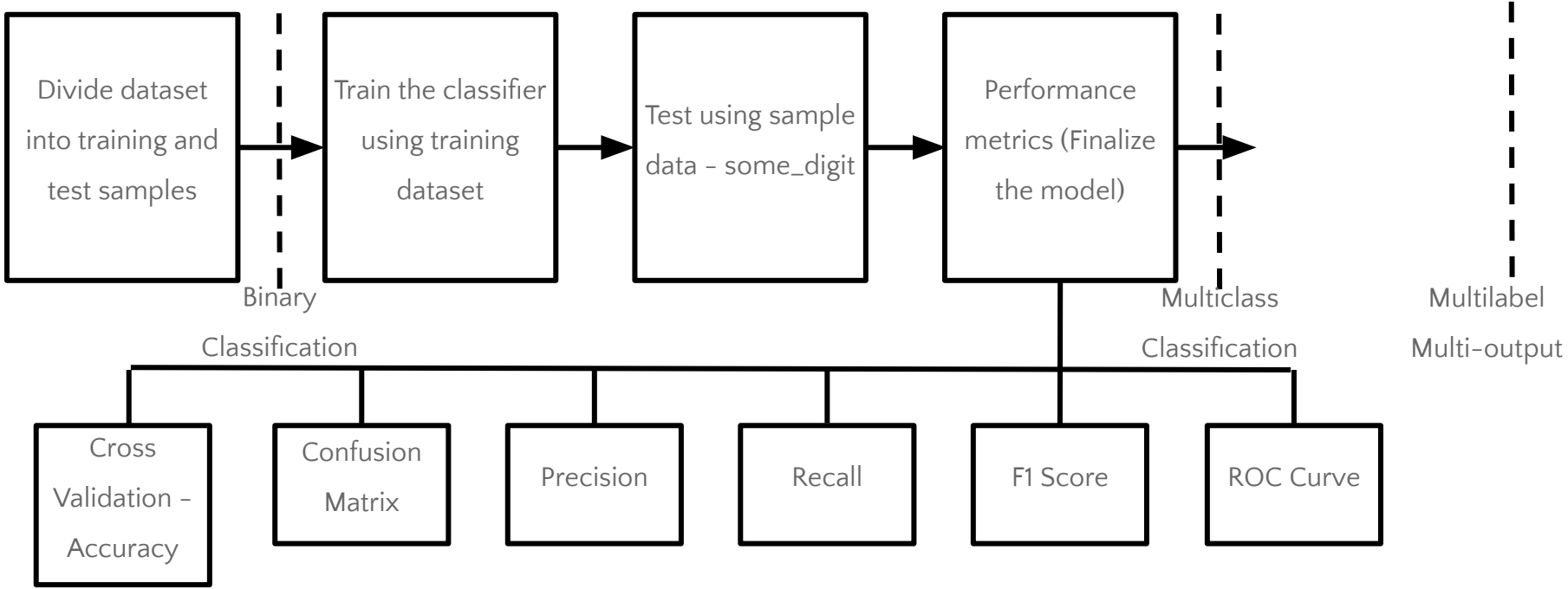
Hands-on - image noise removal

- View cleaned image at index 0

```
>>> clean_digit = knn_clf.predict([X_test_mod[some_index]])
>>> plot_digit(clean_digit)
```

Run it on Notebook

Review till now?



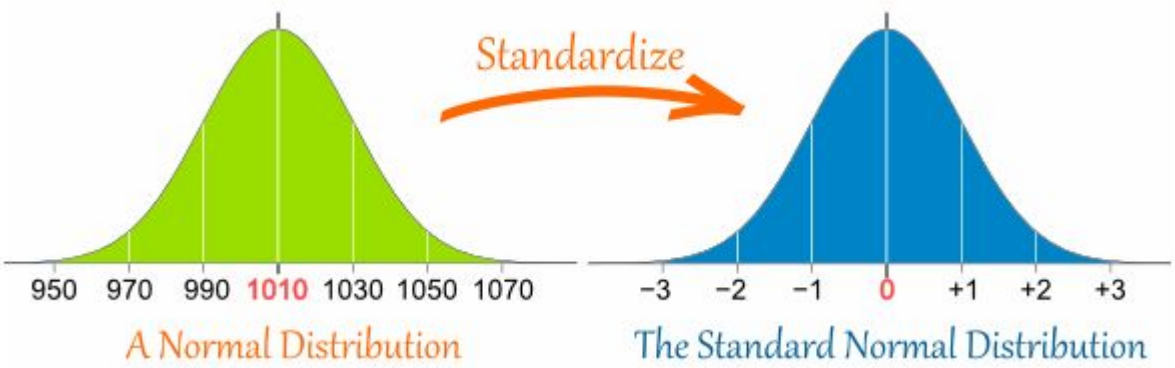
CLOUD x LAB

Questions?

https://discuss.cloudxlab.com

reachus@cloudxlab.com

Feature Scaling – Standardization



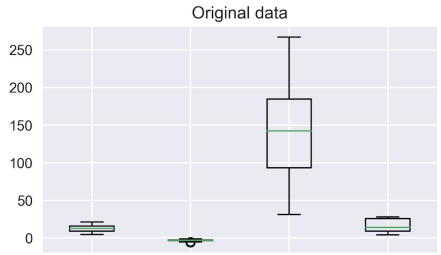
$$\bar{x} = 0 \; and \; \sigma = 1$$

Zero Mean

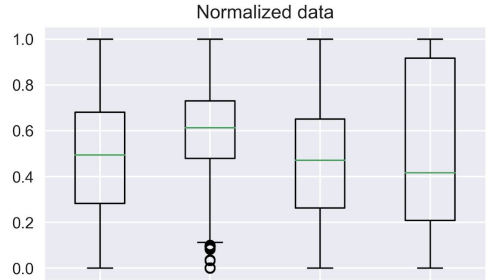
Unit Standard Deviation

CLOUD x LAB

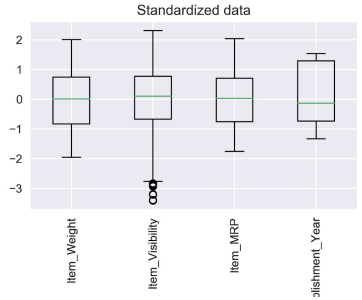
Feature Scaling – Standardization



Original Data



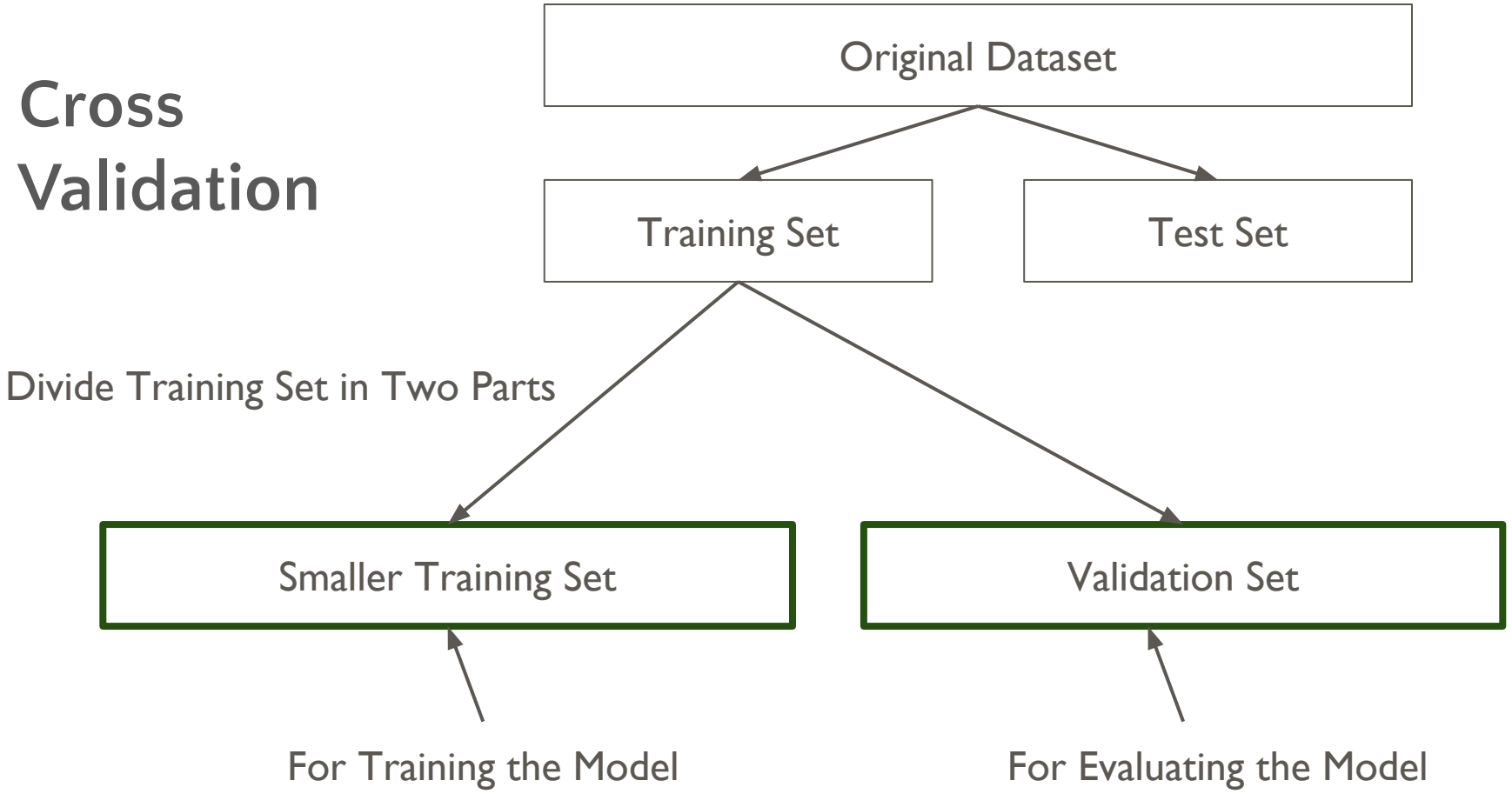
Normalized Data



Standardized Data

CLOUD x LAB

Cross Validation



Cross Validation

With 10 Folds

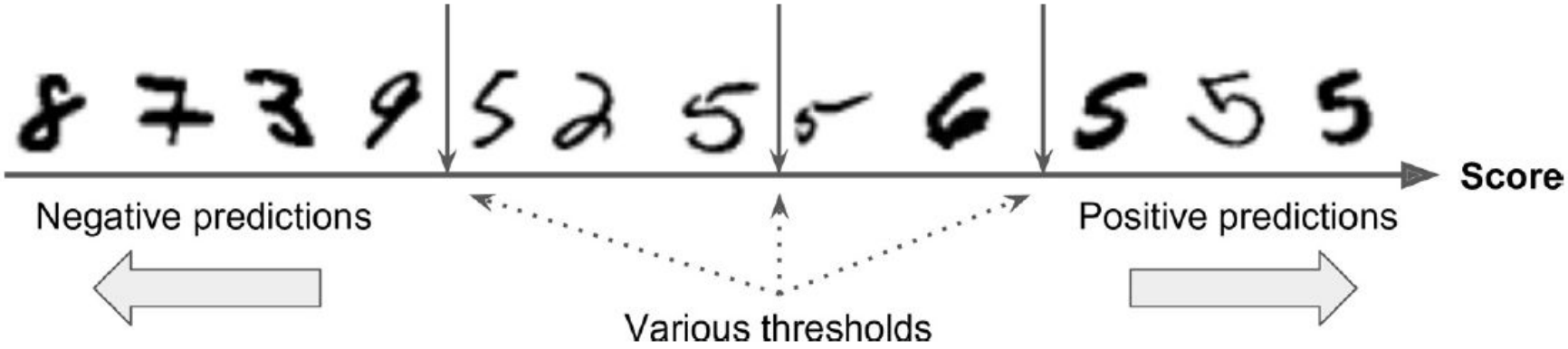
Performance Measures – ROC Curve

- ROC Curve Similar to F1 score but uses a different metric
- Uses True Positive Rate (TPR) = Recall = TP / (TP + FN)
- False Positive Rate (FPR) = FP / (FP + TN)
  = 1 – True Negative Rate (TNR)
- TNR = TN / (FP + TN)

Receiver Operating Characteristics (ROC) plots:
TPR versus FPR

CLOUD x LAB

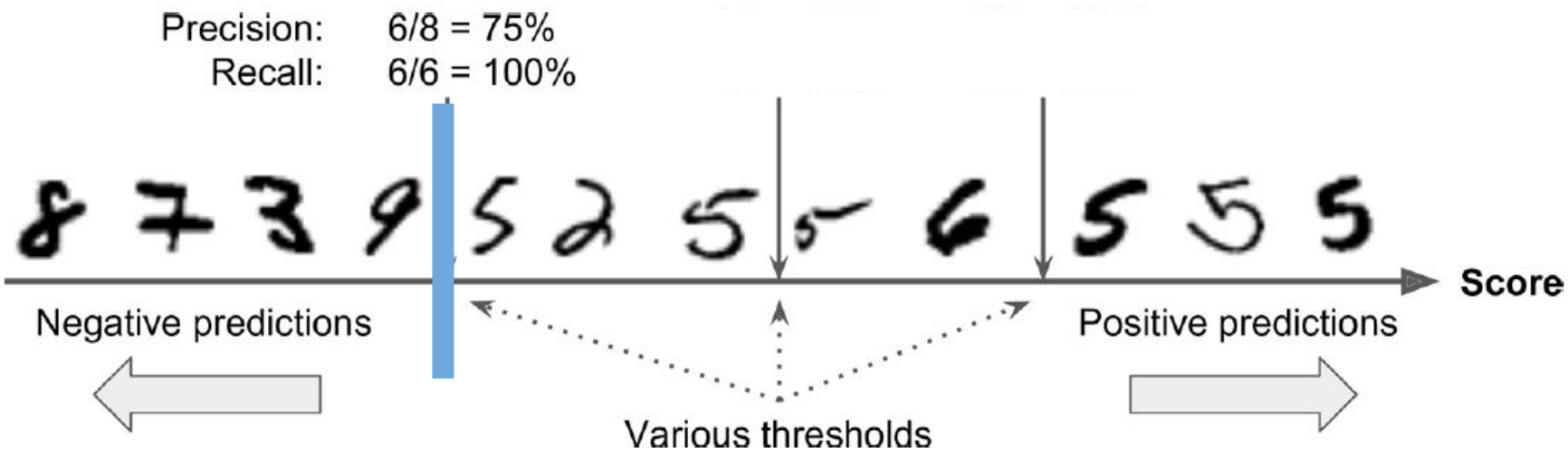
Precision / Recall Tradeoff – Thresholds



Thresholds can be set to achieve certain precision and recall as required

CLOUD x LAB

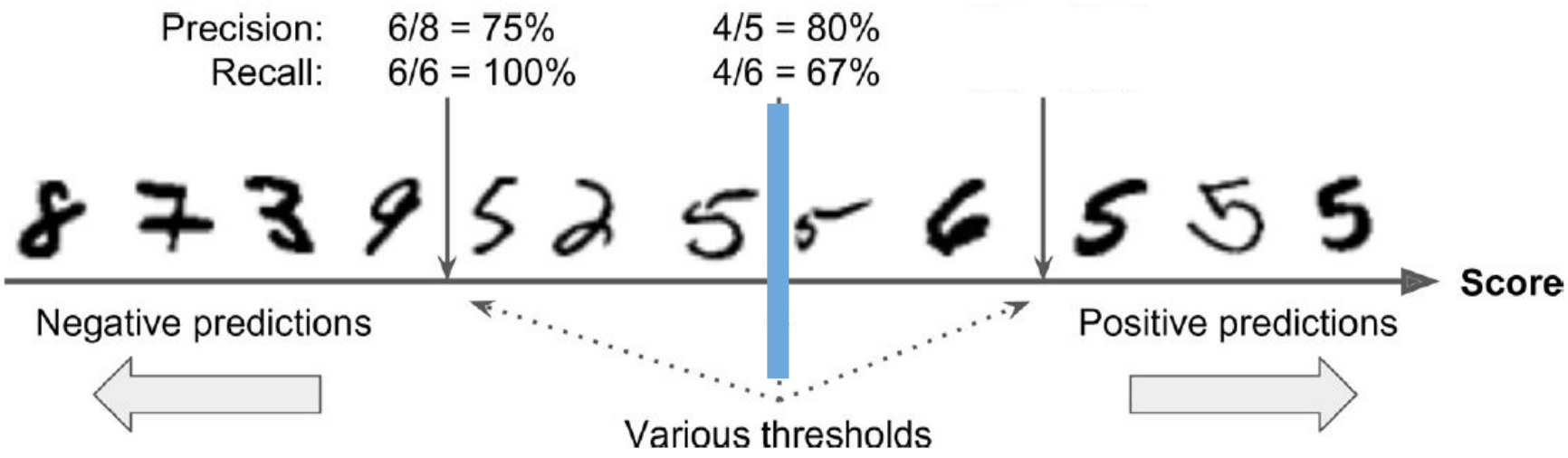
Precision / Recall Tradeoff



In this precision/recall trade-off, images are ranked by their classifier score, and those above the chosen decision threshold are considered positive; the higher the threshold, the lower the recall, but (in general) the higher the precision

CLOUD x LAB

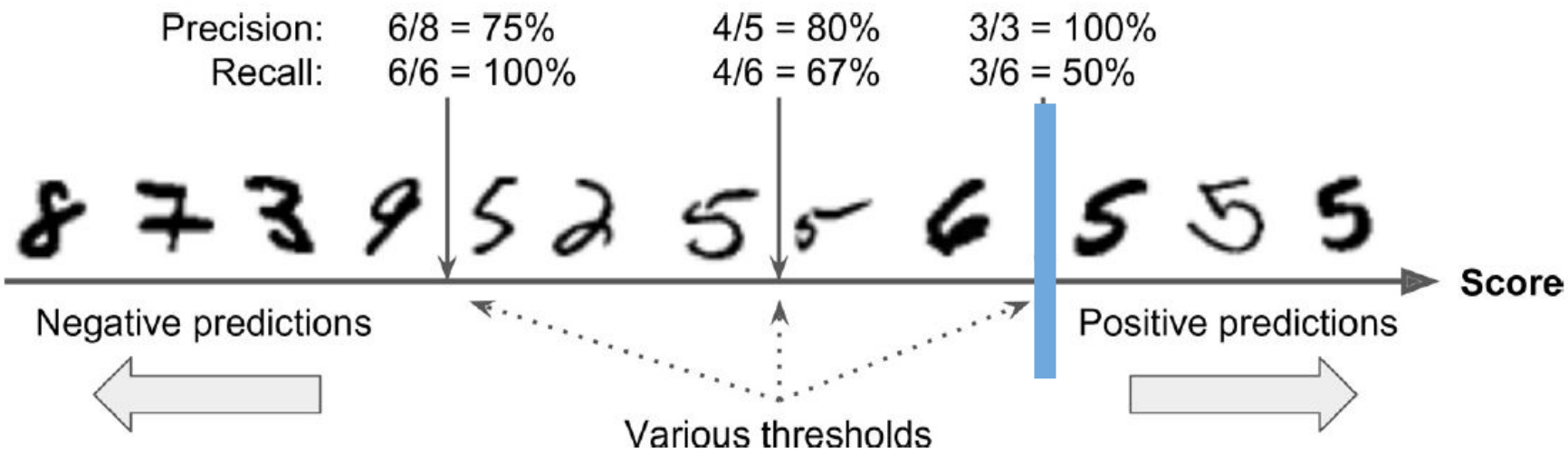
Precision / Recall Tradeoff



In this precision/recall trade-off, images are ranked by their classifier score, and those above the chosen decision threshold are considered positive; the higher the threshold, the lower the recall, but (in general) the higher the precision

CLOUD x LAB

Precision / Recall Tradeoff

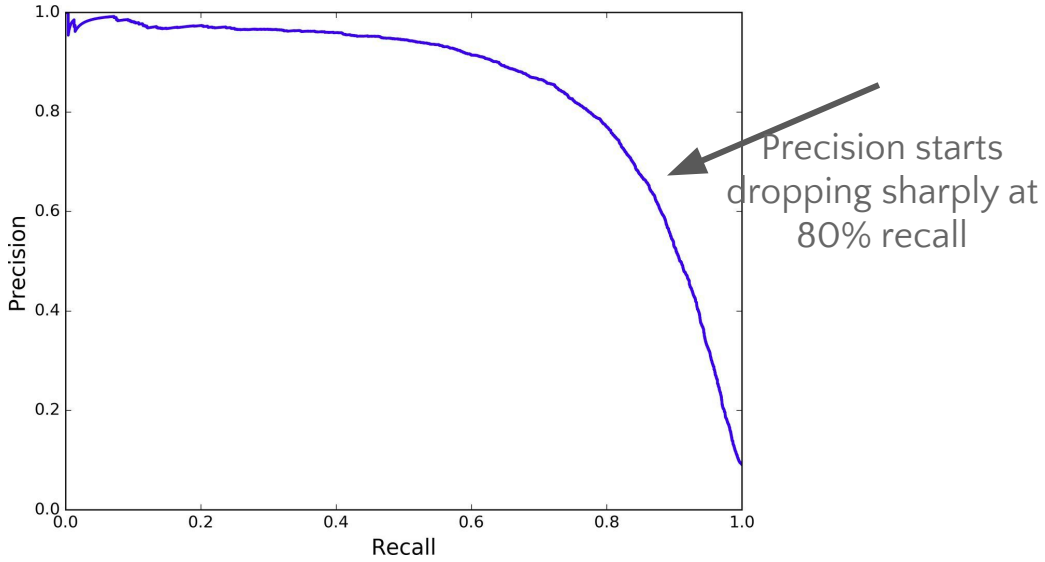


In this precision/recall trade-off, images are ranked by their classifier score, and those above the chosen decision threshold are considered positive; the higher the threshold, the lower the recall, but (in general) the higher the precision

CLOUD x LAB

Precision vs Recall Curve

- Another way to select a good precision/recall tradeoff is to
  - plot **precision directly against recall**.



Precision versus recall

CLOUD x LAB

Precision vs Recall Curve

- Another way to select a good precision/recall tradeoff is to
  - plot **precision directly against recall**.

```
>>> def plot_precision_vs_recall(precisions, recalls):
    plt.plot(recalls, precisions, "b-", linewidth=2)
    plt.xlabel("Recall", fontsize=16)
    plt.ylabel("Precision", fontsize=16)
    plt.axis([0, 1, 0, 1])
    plt.grid(True)

>>> plt.figure(figsize=(8, 6))
>>> plot_precision_vs_recall(precisions, recalls)
>>> plt.show()
```

Run it on Notebook

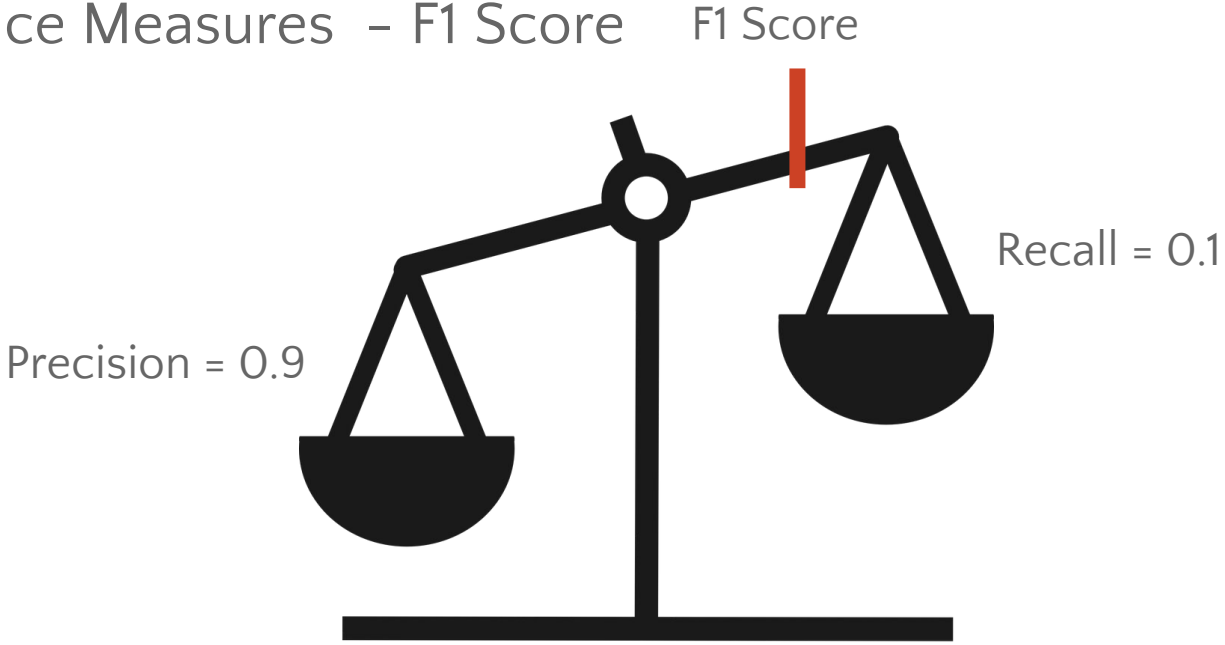
Another Example of Multioutput Classification

A	B	C	D	E	F	G
5	133.5	27	284	638	31	220
5	111.9	27	285	702	36	230
5	99.3	25	310	713	39	227
5	102.5	25	311	670	34	218
5	114.8	25	312	685	34	222

F, 37
C, 26

CLOUD x LAB

Performance Measures – F1 Score

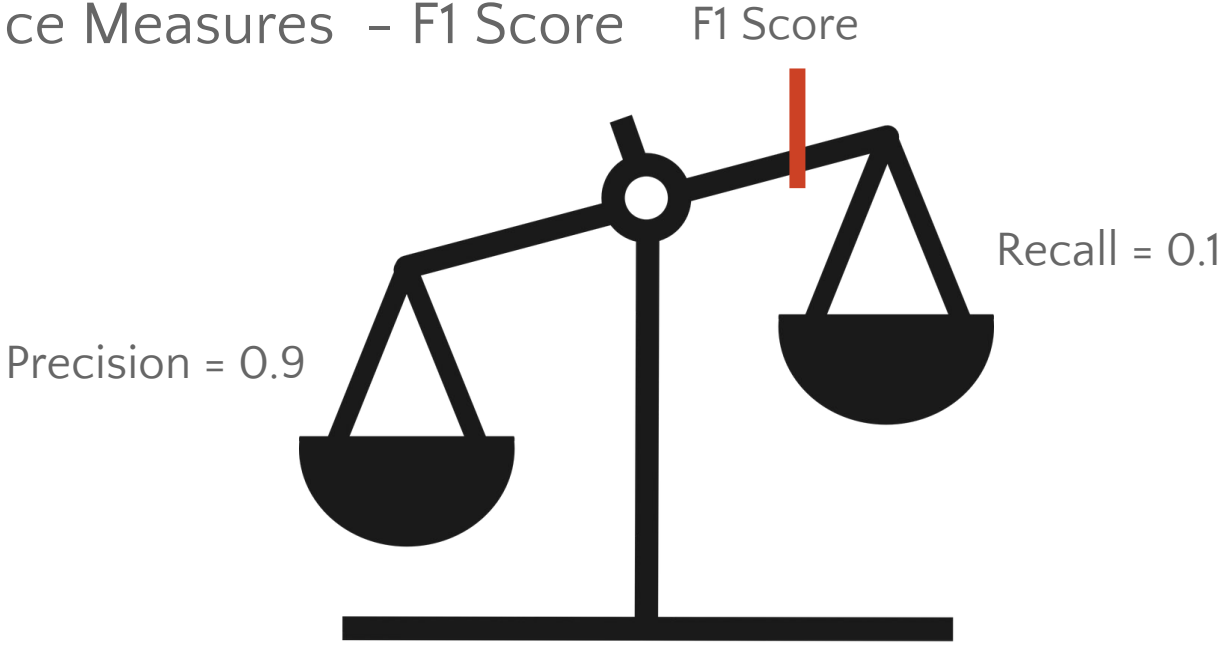


Recall is really small compared to **Precision**

F1 score will be closer to the smaller number than the bigger one

CLOUD x LAB

Performance Measures – F1 Score



Recall is really small compared to **Precision**

Hence F1 score is closer to Recall than Precision

CLOUD x LAB

High Precision or High Recall?

High recall means the model will try to maximize the number of videos that are classified as safe.

1 - Not Safe

2 - Safe ✓

3 - Not Safe

4 - Safe ✓

5 - Not Safe

6 - Safe ✓

High **Recall**

CLOUD x LAB

High Precision or High Recall?

In high recall model may mistake while classifying video as safe. This is because recall is more about classifying all the “safe for kids” videos as “safe” rather than classifying all the videos correctly.

1 – Not Safe

2 – Safe

3 – Not Safe

4 – Safe

5 – Not Safe

6 – Safe

High **Recall**

CLOUD x LAB