Performance Metrics-Classification

FAIRLEIGH DICKINSON UNIVERSITY

Fortiter et Suaviter

ML Flow

Given: labeled training data $X, Y = \{< \boldsymbol{x}_i, y_i >\}_{i=1}^{n}$

- Assumes each $x_i \sim D(X)$ with $y_i = f_{target}(\boldsymbol{x}_i)$

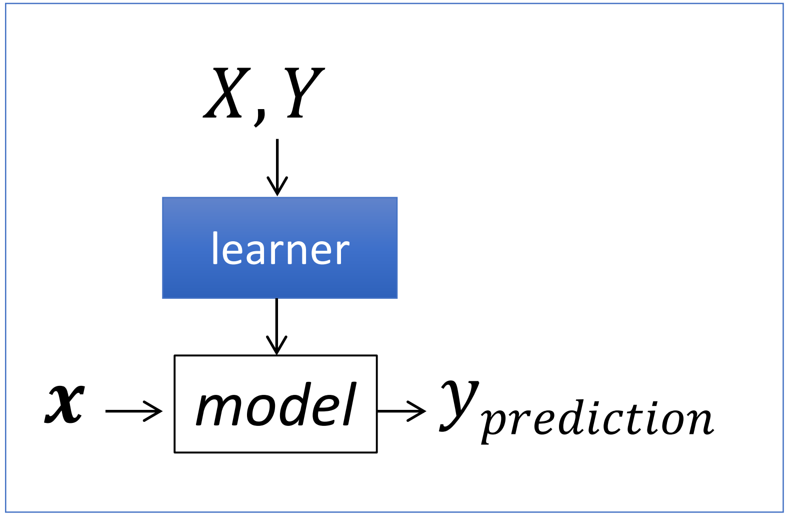
Train the model:

$$model \leftarrow classifier.train(X, Y)$$

Apply the model to new data:

- Given: new unlabeled instance $\boldsymbol{x} \sim D(X)$

$$y_{prediction} \leftarrow model.predict(\boldsymbol{x})$$



Key questions:

How to determine the quality of the model?

(i) measuring performance

(ii) understanding the significance of the results (is it better the other models?)

Why performance metrics are important?

- Training objective (cost function) is only a proxy for real world objectives.
- Metrics help capture a business goal into a quantitative target (not all errors are equal).
- Helps organize ML team effort towards that target.
  - Generally in the form of improving that metric on the dev set.
- Useful to quantify the “gap” between:
  - Desired performance and baseline (estimate effort initially).
  - Desired performance and current performance.
  - Measure progress over time.
- Useful for lower level tasks and debugging (e.g. diagnosing bias vs variance).
- Ideally training objective should be the metric, but not always possible. Still, metrics are useful and important for evaluation.

Metrics for Performance Evaluation

- Focus on the predictive capability of a model
  - Rather than how fast it takes to classify or build models, scalability, etc.

- **Confusion Matrix**:

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

a: TP (true positive)

b: FN (false negative)

c: FP (false positive)

d: TN (true negative)

Metrics for Performance Evaluation

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

a: TP (true positive)

b: FN (false negative)

c: FP (false positive)

d: TN (true negative)

Most widely-used metric

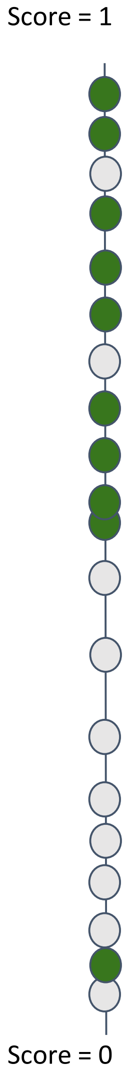
$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

Error=1- Accuracy

Binary Classification

- x is input
- y is binary output (0/1)
- Model is $\hat{y} = h(x)$
- Two types of models
  - Models that output a categorical class directly (K-nearest neighbor, Decision tree)
  - Models that output a real valued score (Logistic Regression, NN, SVM)
    - Score could be probability (LR, NN), margin:distance from the decision boundary(SVM)
    - Need to pick a threshold
    - We focus on this type (the other type can be interpreted as an instance)

Score Based Models



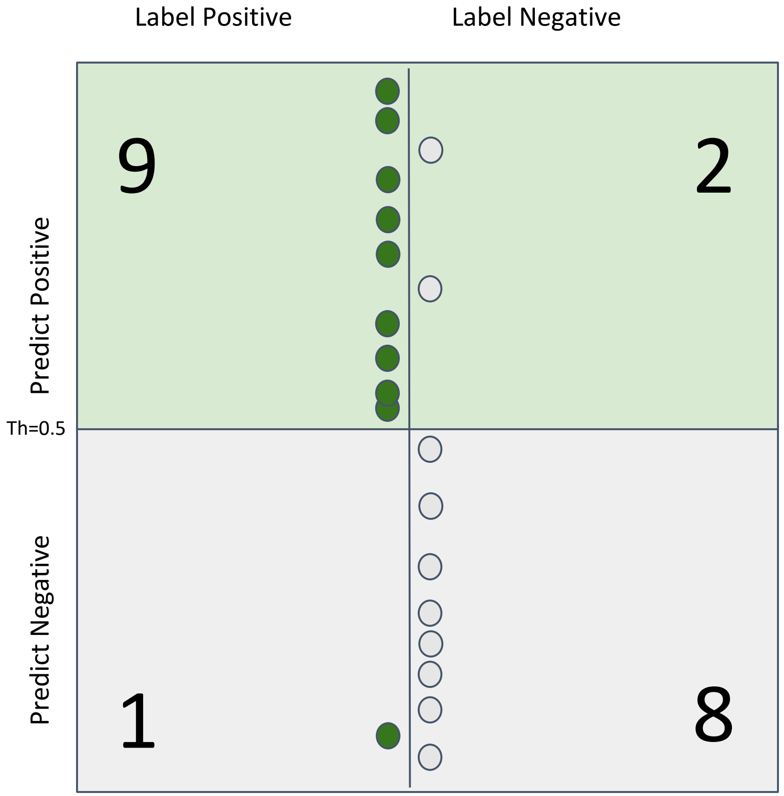
●	Positive example
○	Negative example

Example of Score: Output of logistic regression.
For most metrics: Only ranking matters.
If too many examples: Plot class-wise histogram.

$$\text{Prevalence} = \frac{\#\ \text{positive samples}}{\#\ \text{positive} + \#\ \text{negatives samples}}$$

Prevalence: tells you **how imbalanced** your dataset is

Confusion Matrix

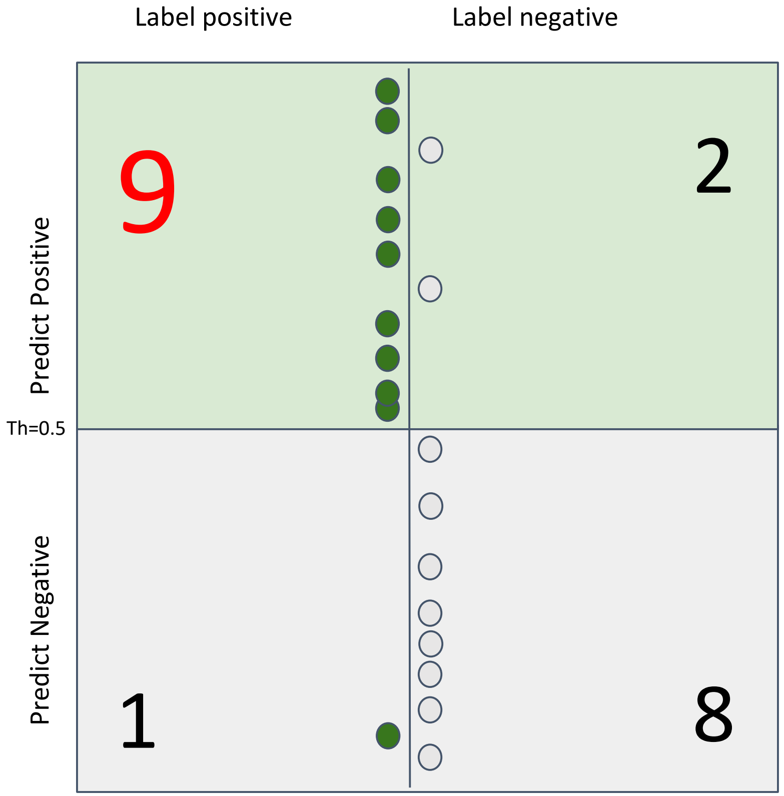


Th
0.5

Properties:

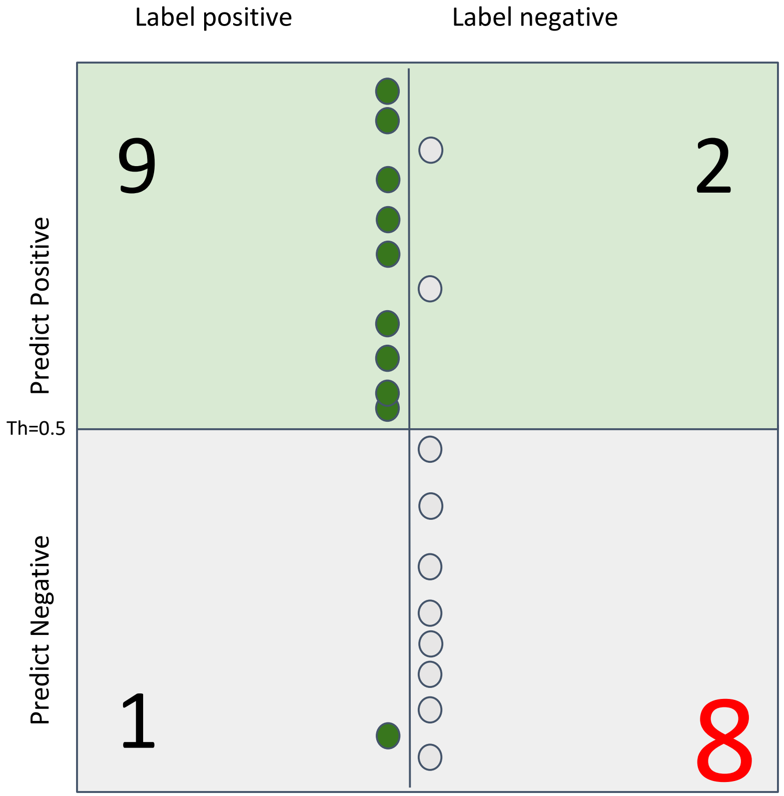
- Total sum is fixed (population).
- Column sums are fixed (class-wise population).
- Quality of model & threshold decide how columns are split into rows.
- We want diagonals to be “heavy”, off diagonals to be “light”.

True Positives



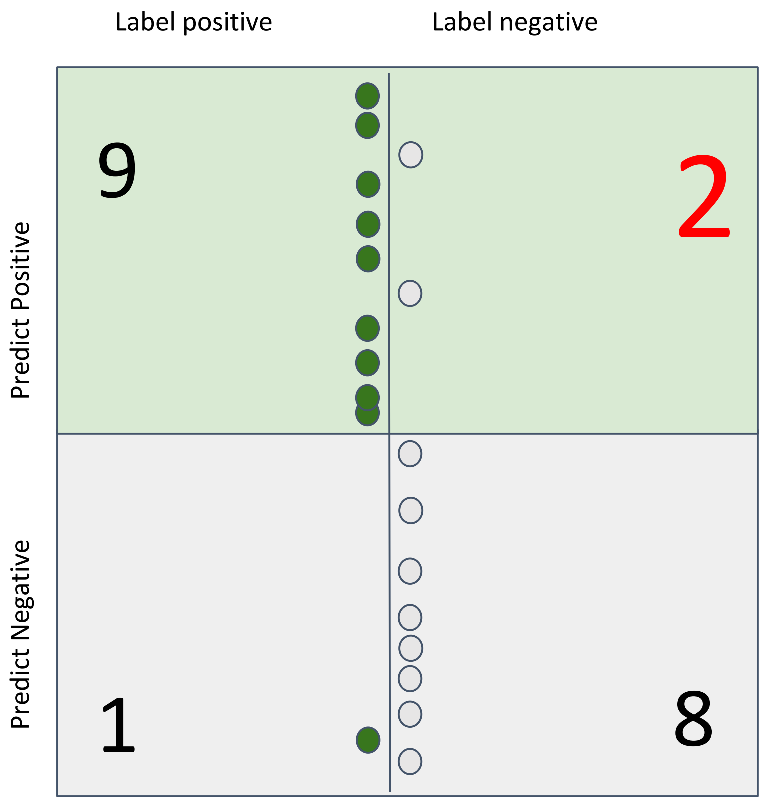
Th	TP
0.5	9

True Negatives



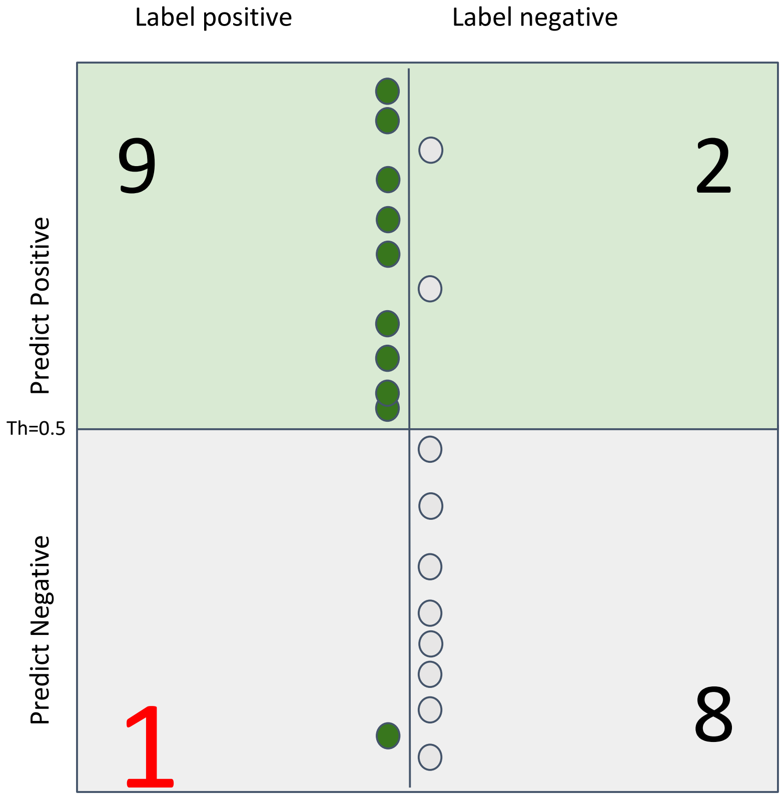
Th	TP	TN
0.5	9	8

False Positives



Th	TP	TN	FP
0.5	9	8	2

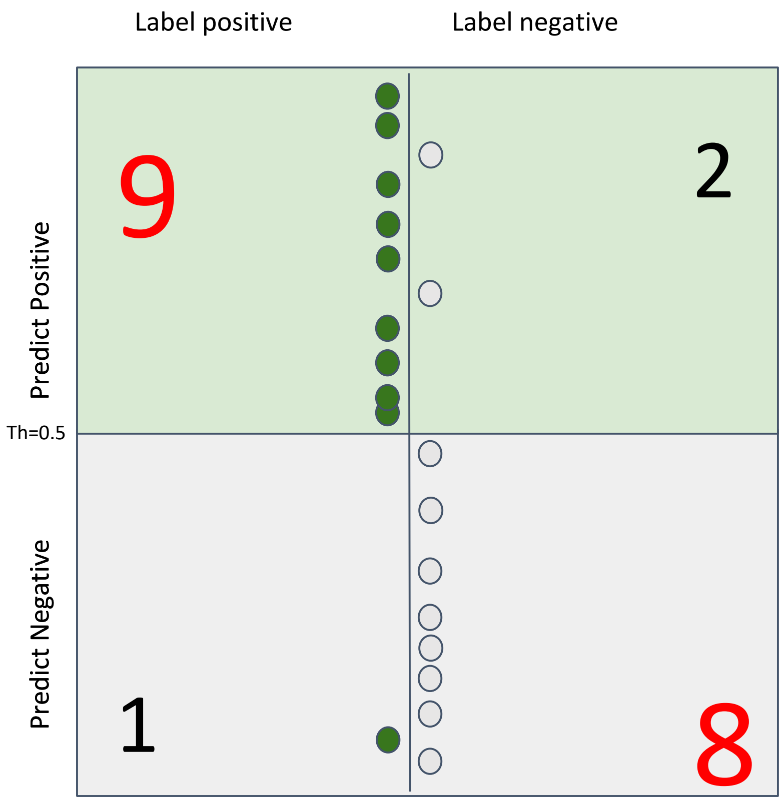
False Negatives



Th	TP	TN	FP	FN
0.5	9	8	2	1

FAIRLEIGH DICKINSON UNIVERSITY

Accuracy

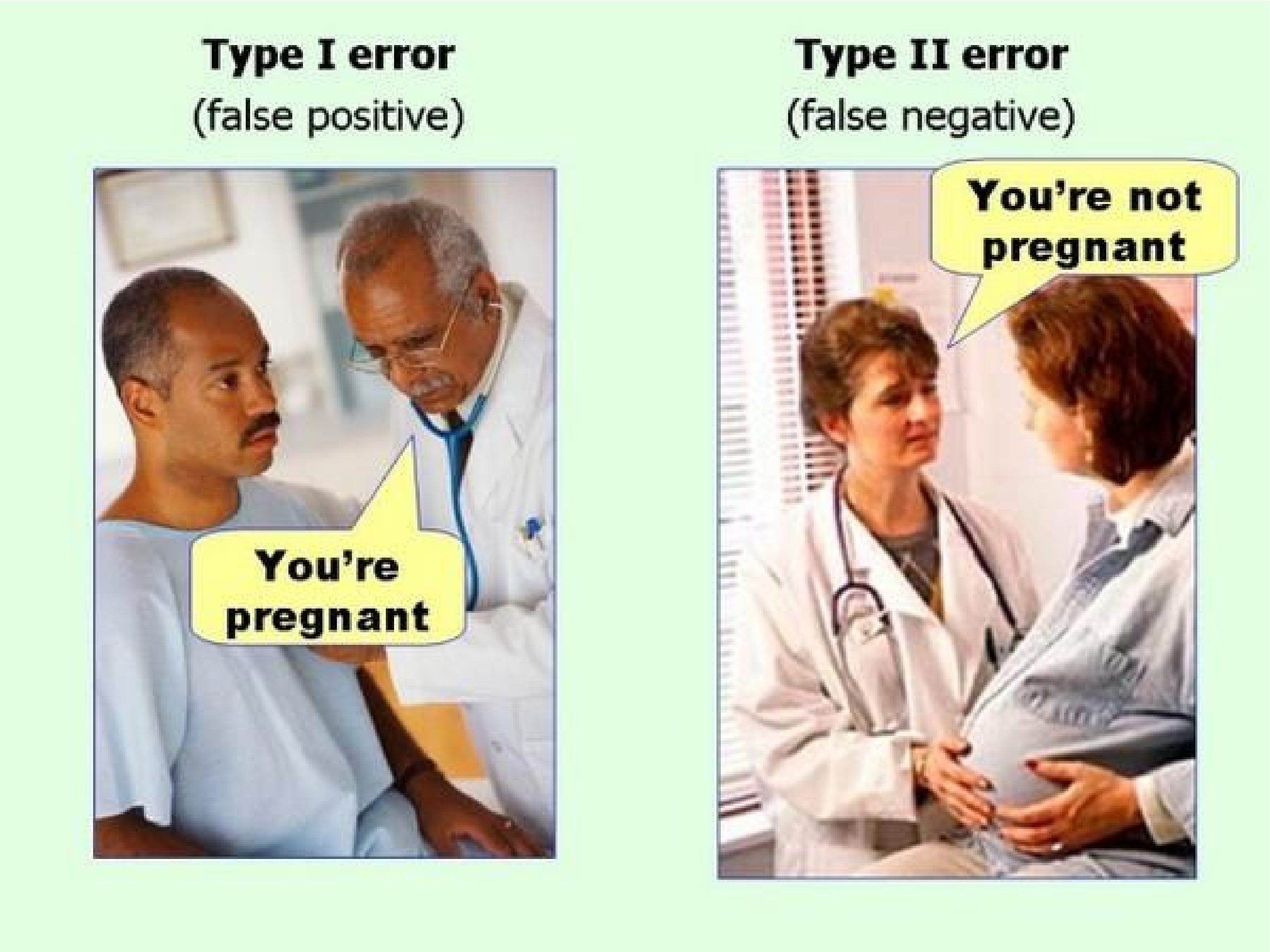


Th	TP	TN	FP	FN	Acc
0.5	9	8	2	1	.85

Equivalent to 0-1 Loss!

Any possible problem with it?

FP and FN also called Type-1 and Type-2 errors



Limitation of Accuracy

- Consider a 2-class problem (binary)
  - Number of Class 0 examples = 9990
  - Number of Class 1 examples = 10
- If model predicts everything to be class 0, accuracy is 9990/10000 = 99.9 %
  - Accuracy is misleading because model does not detect any class 1 example

Alternate Metrics

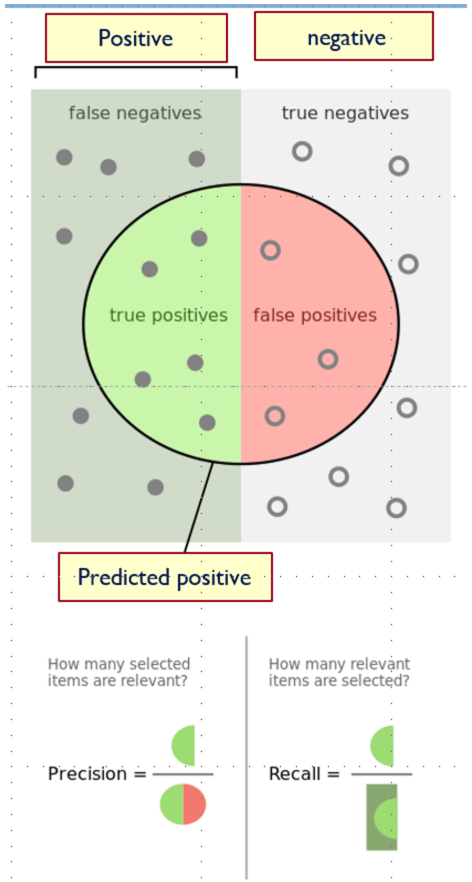
- If the Binary classification problem is biased
  - In many problems most examples are negative
- Or, in multiclass classification
  - The distribution over labels is often non-uniform
- Simple accuracy is not a useful metric.
  - Often we resort to task specific metrics
- We require to involve Recall and Precision

Recall: # true positives/ # all positive

Recall=TP/(TP+FN)

Precision: # true positives/ # predicted positive

Precision=TP/(TP+FP)



Alternate Metrics(cont.)

The notion of a confusion matrix can be usefully extended to the multiclass case (i,j) cell indicate how many of the i-labeled examples were predicted to be j

- Given a dataset of P positive instances and N negative instances:

		Predicted Class	
		Yes	No
Actual Class	Yes	TP	FN
	No	FP	TN

$$\text{accuracy} = \frac{TP + TN}{P + N}$$

- Imagine using classifier to identify positive cases (i.e., for information retrieval)

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

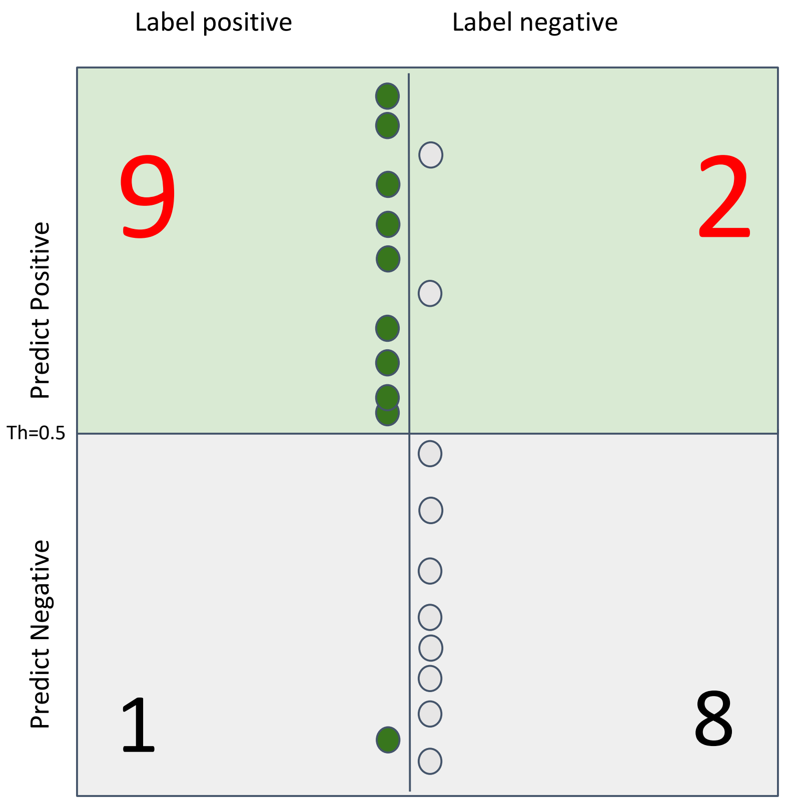
Probability that a randomly selected positive prediction is indeed positive

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

Probability that a randomly selected positive is identified

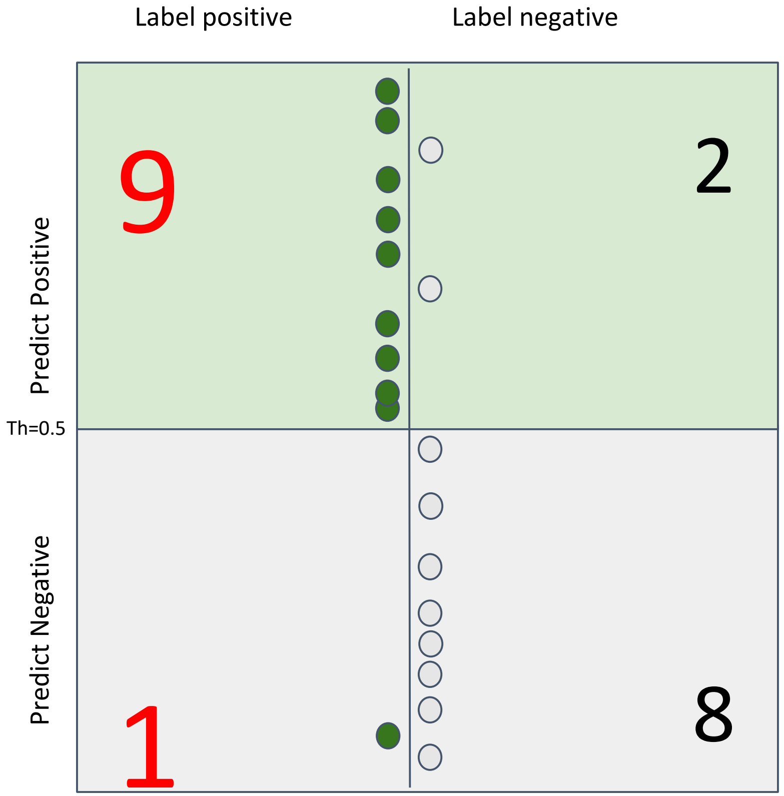
FAIRLEIGH DICKINSON UNIVERSITY

Precision



Th	TP	TN	FP	FN	Acc	Pr
0.5	9	8	2	1	.85	.81

Recall (Sensitivity)

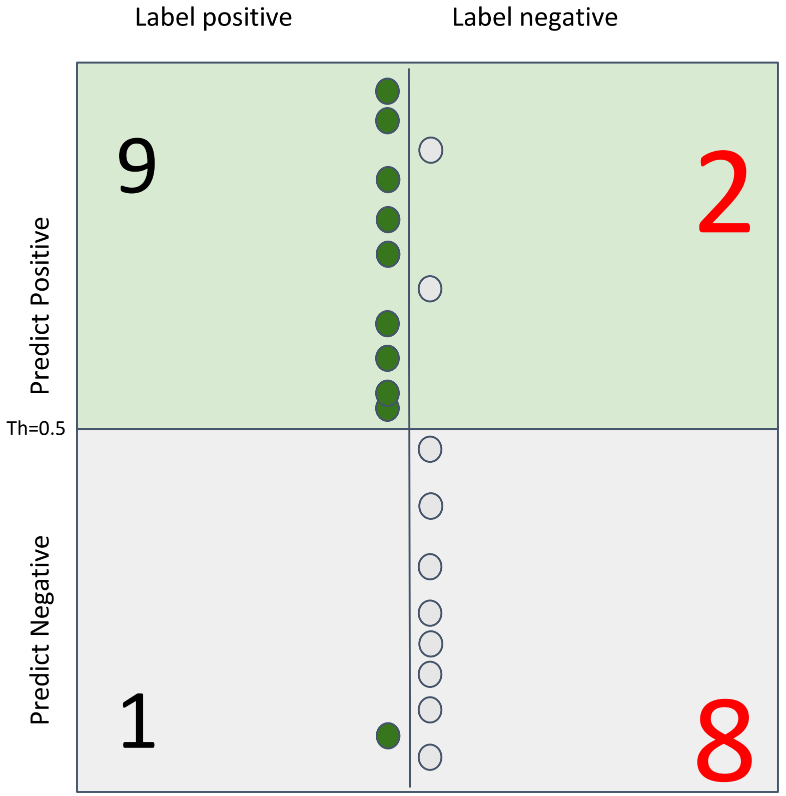


Th	TP	TN	FP	FN	Acc	Recall
0.5	9	8	2	1	.85	.9

Trivial 100% recall = pull everybody above the threshold.
Trivial 100% precision = push everybody below the threshold except 1 green on top.
(Hopefully no gray above it!)

Striving for good precision with 100% recall = pulling up the lowest green as high as possible in the ranking.
Striving for good recall with 100% precision = pushing down the top gray as low as possible in the ranking.

Negative Recall (Specificity)



Th	TP	TN	FP	FN	Acc	Pr	Recall	Spec
0.5	9	8	2	1	.85	.81	.9	0.8

Examples- Imbalanced Data

Scenario	What's more important	Reason
Medical diagnosis	Recall	Missing a positive case is dangerous
Fraud detection	Both	Missing fraud = loss; false alarms = frustration
Spam filtering	Precision	Avoid marking real emails as spam
Cybersecurity	Recall (with balanced Precision)	Missing an attack can be critical
Search engines	Precision	Users prefer fewer but relevant results

Example: Medical Diagnosis(e.g. Cancer Detection)

- **Goal:** Identify all patients who have cancer.
- **Challenge:** Very few positive cases → class imbalance.
- **Why Recall matters most:**
  - Missing a patient with cancer (**false negative**) can be life-threatening.
  - It's better to raise some false alarms (low precision) than to miss actual positives.

Example:

1000 patients → 10 have cancer

- Model predicts 8 of them correctly (Recall = 0.8), but also wrongly flags 20 healthy ones (Precision = 0.29).
- Accuracy might look 98%, but Recall tells us how well we're catching the real cases.

Example: Spam Email Filtering

- **Goal:** Identify spam emails automatically.
- **Why Precision matters most:**
  - If Precision is low, important legitimate emails get marked as spam (**false positives**), which frustrates users.
  - Recall can be slightly lower (missing a few spam emails) — that's acceptable.
  - Gmail's filters aim for very high precision so that real emails are never lost, even if a few spam ones slip through.

Example: Credit Card Fraud Detection

- **Goal:** Detect fraudulent transactions.
- **Challenge:** Fraud cases are <1% of total transactions.
- **Why both Precision and Recall matter:**
  - Low Recall: Miss fraud → financial loss.
  - Low Precision: Too many false alarms → annoy customers and waste resources.

- So, the goal is to find a good balance (often via **F1-score**) that minimizes both risks.

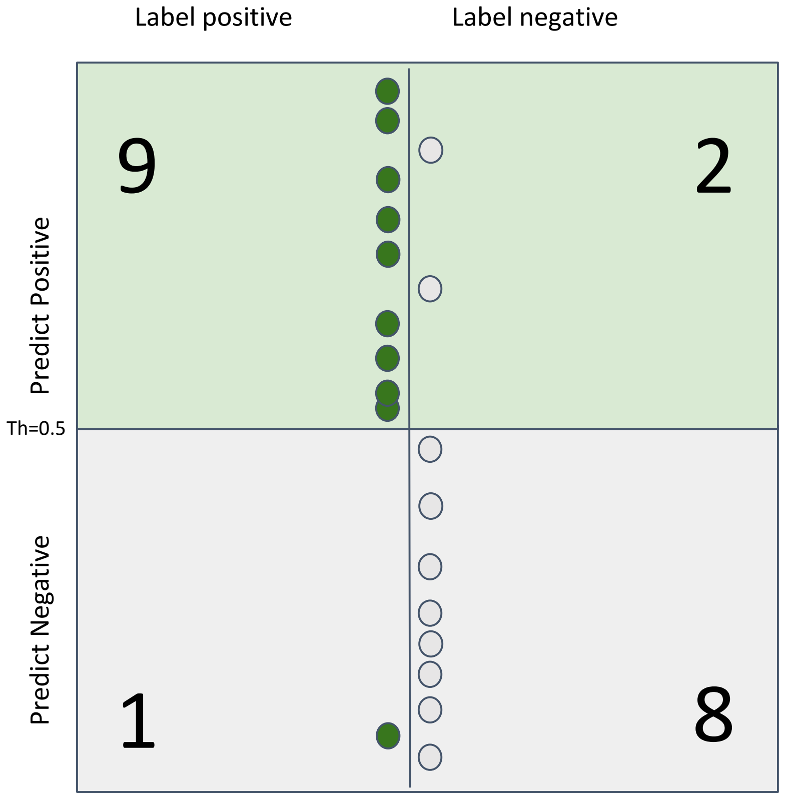
F1 Score

FAIRLEIGH DICKINSON UNIVERSITY

- Precision and recall often **conflict**
  - Increasing **recall** (catching more positives) may reduce precision (more false alarms).
  - Increasing **precision** (fewer false alarms) may reduce recall (missed positives).
- **Accuracy** is misleading on imbalanced data.
  - It makes sense to consider Recall and Precision together or combine them into a single metric.
- **F1** provides a *single, interpretable metric* when both Precision and Recall are important.
  - It **penalizes extreme imbalance** — a high F1 requires both precision and recall to be reasonably good.
- **F1 Score** is a measure that **combines precision and recall** is the harmonic mean of precision and recall.

$$F_1 = \left(\frac{2}{\mathrm{recall}^{-1} + \mathrm{precision}^{-1}}\right) = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}.$$

F1 Score



Th	TP	TN	FP	FN	Acc	Pr	Recall	Spec	F1
0.5	9	8	2	1	.85	.81	.9	.8	.857

$$F_1 = \left(\frac{2}{\mathrm{recall}^{-1} + \mathrm{precision}^{-1}}\right) = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}.$$

Example

Suppose we have a medical test for detecting a rare disease.

Metric	Value	Interpretation
Recall	0.5	50% of all actual patients were detected
Precision	0.8	80% of detected cases were correct
F1 Score	?	Combines the two

F1=2×(0.5*0.8)/(0.5*0.8)=0.615

F1 = 0.615, showing moderate performance — better than either metric alone.

Why harmonic mean (not arithmetic)?

Because the harmonic mean **penalizes imbalance** — if one of precision or recall is very low, F1 drops sharply.

Example:

Precision = 1.0, Recall = 0.1 → F1 = 0.18

Precision = 0.55, Recall = 0.55 → F1 = 0.55

So, the F1 score emphasizes **consistency** between precision and recall.

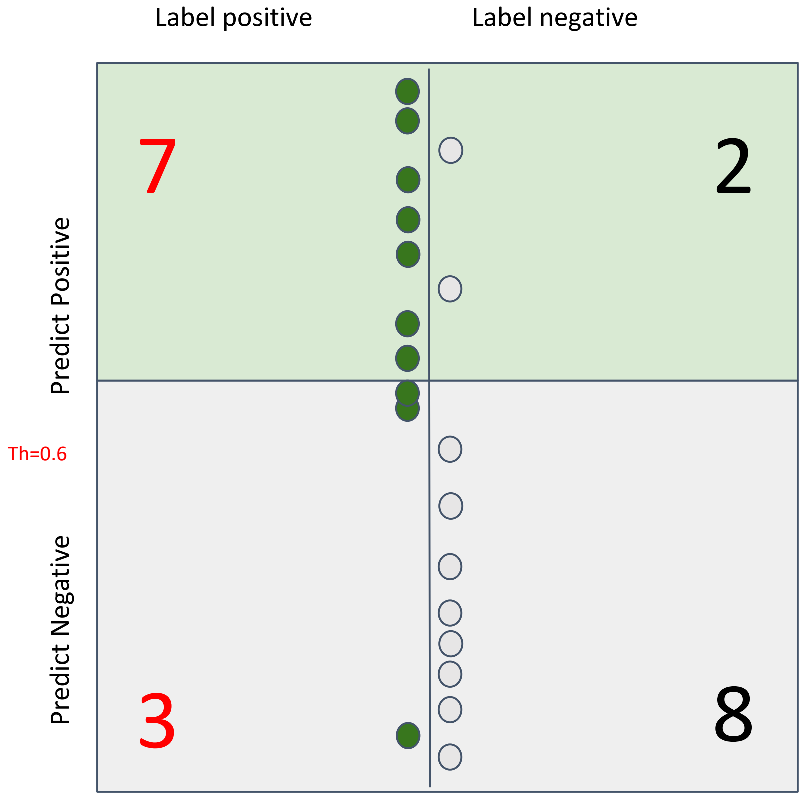
How to improve F1?

FAIRLEIGH DICKINSON UNIVERSITY

Approach	Description
Threshold tuning	Adjust classification threshold (e.g., 0.5 → 0.3) to increase recall or precision as needed
Class weighting	Penalize false negatives or positives differently during training
Resampling	Oversample minority class or undersample majority class
Better features or model	Use richer data, embedding, or ensemble models to improve overall discrimination

FAIRLEIGH DICKINSON UNIVERSITY

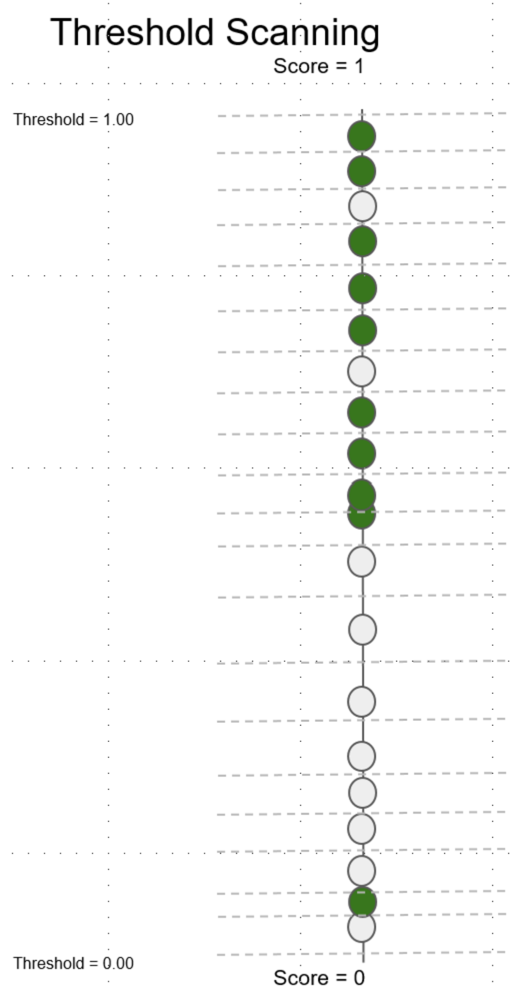
Changing threshold



Th	TP	TN	FP	FN	Acc	Pr	Recall	Spec	F1
0.6	7	8	2	3	.75	.77	.7	.8	.733

effective thresholds = # examples + 1

Threshold Tuning



Threshold	TP	TN	FP	FN	Accuracy	Precision	Recall	Specificity	F1
1.00	0	10	0	10	0.50	1	0	1	0
0.95	1	10	0	9	0.55	1	0.1	1	0.182
0.90	2	10	0	8	0.60	1	0.2	1	0.333
0.85	2	9	1	8	0.55	0.667	0.2	0.9	0.308
0.80	3	9	1	7	0.60	0.750	0.3	0.9	0.429
0.75	4	9	1	6	0.65	0.800	0.4	0.9	0.533
0.70	5	9	1	5	0.70	0.833	0.5	0.9	0.625
0.65	5	8	2	5	0.65	0.714	0.5	0.8	0.588
0.60	6	8	2	4	0.70	0.750	0.6	0.8	0.667
0.55	7	8	2	3	0.75	0.778	0.7	0.8	0.737
0.50	8	8	2	2	0.80	0.800	0.8	0.8	0.800
0.45	9	8	2	1	0.85	0.818	0.9	0.8	0.857
0.40	9	7	3	1	0.80	0.750	0.9	0.7	0.818
0.35	9	6	4	1	0.75	0.692	0.9	0.6	0.783
0.30	9	5	5	1	0.70	0.643	0.9	0.5	0.750
0.25	9	4	6	1	0.65	0.600	0.9	0.4	0.720
0.20	9	3	7	1	0.60	0.562	0.9	0.3	0.692
0.15	9	2	8	1	0.55	0.529	0.9	0.2	0.667
0.10	9	1	9	1	0.50	0.500	0.9	0.1	0.643
0.05	10	1	9	0	0.55	0.526	1	0.1	0.690
0.00	10	0	10	0	0.50	0.500	1	0	0.667

Question?