

CS5228 LECTURE 5: CLASSIFICATION AND REGRESSION

Bryan Hooi
School of Computing
National University of Singapore

ANNOUNCEMENTS

- No lecture next week (recess week).
- If you filled in the 1st iteration of the project group matching survey, you should have received your matching results. The 2nd iteration survey has been released; if you still would like to be matched, please fill it out (ending 22 Feb).

Week	Date	Topics	Tutorials	Important Dates
1	13 Jan	Introduction		
2	20 Jan	No class (public holiday)		
3	27 Jan	Clustering I	Tutorial 1	
4	3 Feb	Clustering II		Release A1 + project
5	10 Feb	Association Rules	Tutorial 2	
6	17 Feb	Regression & Classification I		
Recess		No class		
7	3 Mar	Regression & Classification II	Tutorial 3	A1 due (Sunday 11.59pm), release A2
8	10 Mar	Regression & Classification III		
9	17 Mar	Recommender Systems	Tutorial 4	
10	24 Mar	Graph Mining		
11	31 Mar	Data Stream Mining	Tutorial 5	A2 due (Sunday 11.59pm)
12	7 Apr	No class (public holiday)		
13	14 Apr	Review & Outlook		Project due (Sunday 11.59pm)

REVIEW: ASSOCIATION RULE MINING

Given: a minimum support threshold minsup , and a minimum confidence threshold minconf

Find: all association rules with support $\geq \text{minsup}$, and confidence $\geq \text{minconf}$

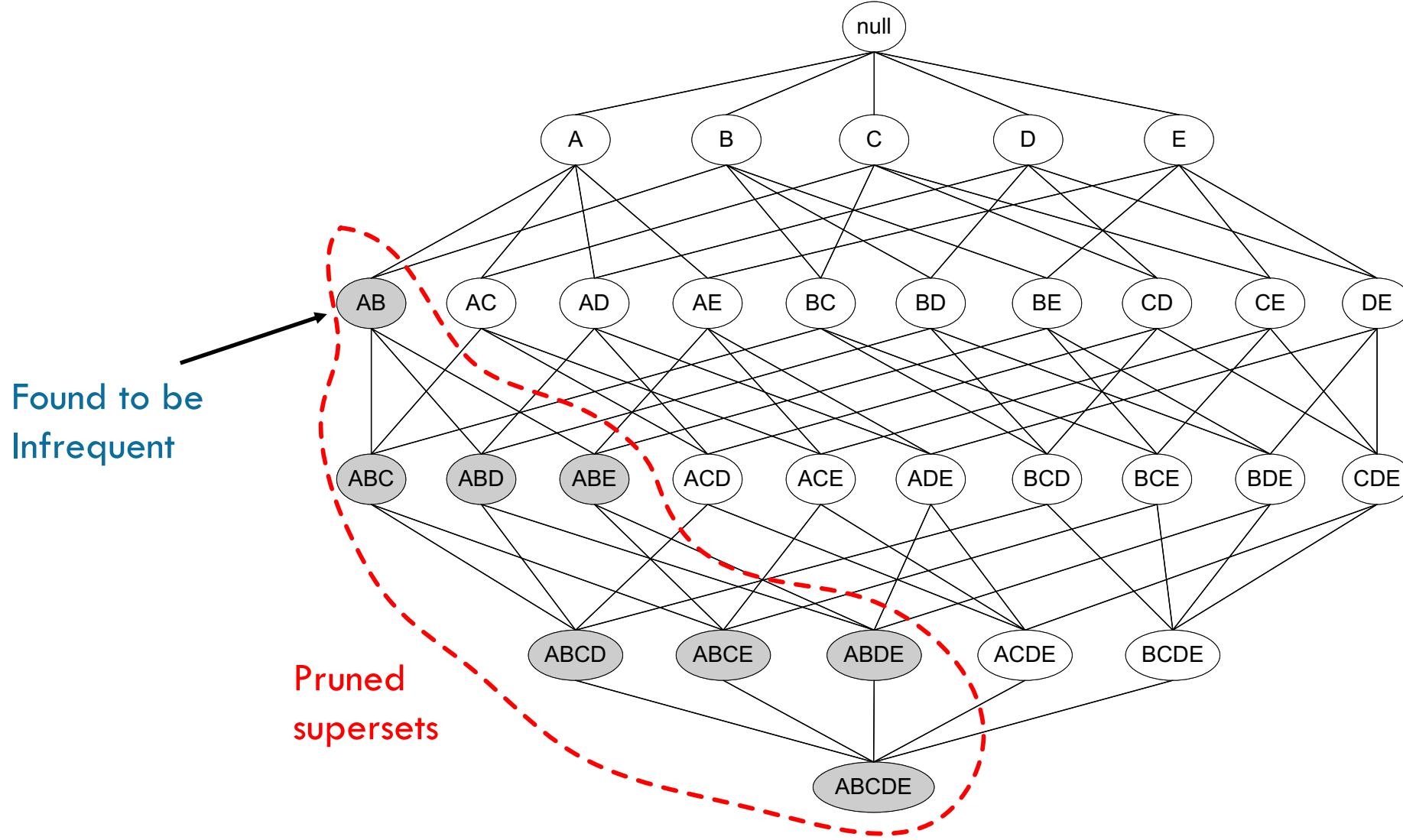
<i>TID</i>	<i>Items</i>
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

$$\text{minsup} = 0.6 \quad \text{minconf} = 1$$

Diaper \rightarrow Beer,
Milk \rightarrow Diaper,
Bread \rightarrow Diaper,
Milk \rightarrow Bread,

Beer \rightarrow Diaper,
Diaper \rightarrow Milk,
Diaper \rightarrow Bread,
Bread \rightarrow Milk

REVIEW: APRIORI PRINCIPLE

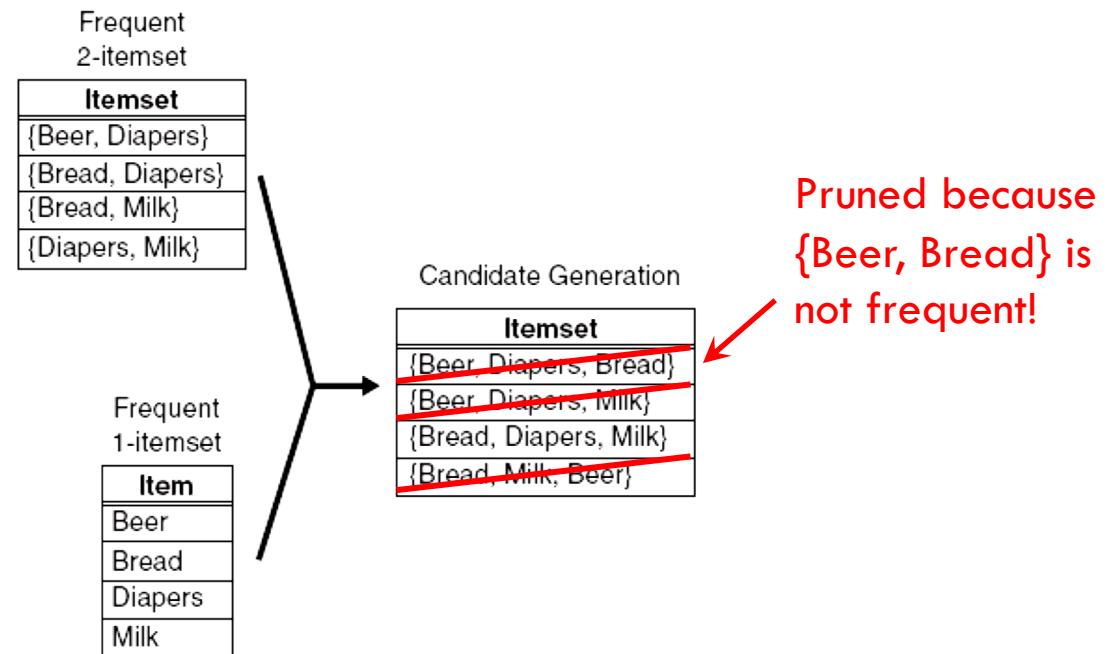


REVIEW: GENERATION STEP ($F_{k-1} \times F_1$ METHOD)

- **Merge** frequent $(k-1)$ and 1-itemsets, then
- **Prune** resulting k -itemsets if they have a $(k-1)$ subset which is not frequent

Apriori Algorithm:

- For $k = 1, 2, \dots$
 - **Generate** candidate frequent k -itemsets
 - **Filter** candidates to get all frequent k -itemsets



REVIEW: FILTER STEP

Apriori Algorithm:

- For $k = 1, 2, \dots$
- **Generate** candidate frequent k -itemsets
- **Filter** candidates to get all frequent k -itemsets

- Query the database to compute support of each candidate
- Filter away candidates with support $< \text{minsup}$

Itemset	SC
ab	1
ac	2
ae	1
bc	2
be	3
ce	2

CLASSIFICATION & REGRESSION: OVERVIEW

1. Problem Setup
2. Evaluating Classifiers
3. Nearest Neighbor Methods
4. Tree-Based Methods
5. Ensemble Methods
6. Logistic Regression
7. Deep Learning

Today's
Lecture



OVERVIEW

1. Problem Setup
2. Evaluating Classifiers
3. Nearest Neighbor Methods
4. Tree-Based Methods
5. Ensemble Methods
6. Logistic Regression
7. Deep Learning

HEART RHYTHM CLASSIFICATION

BBC | Sign in

News Sport Reel Worklife Travel Future More

NEWS

Home Video World Asia UK Business Tech Science Stories Entertainment & Arts

The proven health trackers saving thousands of lives

By Matthew Wall
Technology of Business editor

15 November 2016

f Share



Medical ECG devices

Wearable devices



Classification



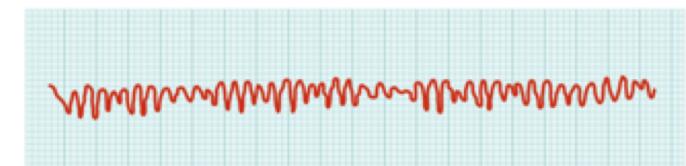
(a) Second-degree (partial) block



(b) Atrial fibrillation



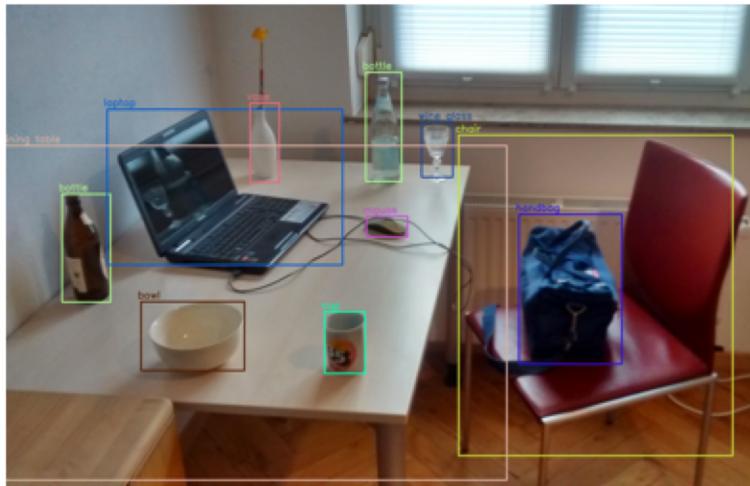
(c) Ventricular tachycardia



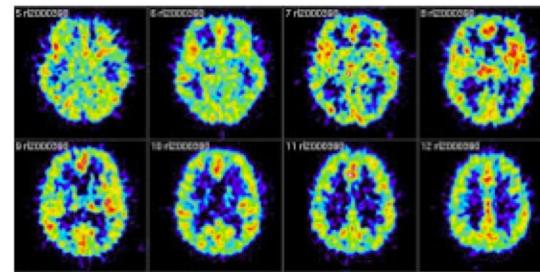
(d) Ventricular fibrillation

• • •

MORE APPLICATIONS



Object Recognition



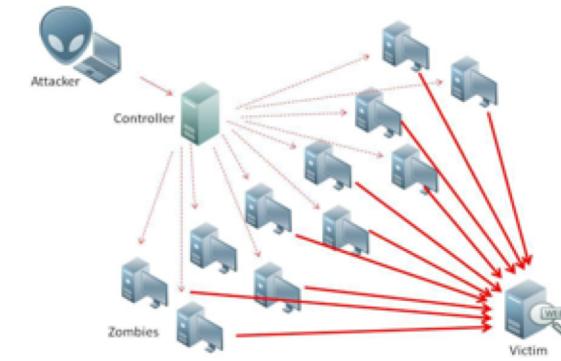
Medical Imaging

[PayPal](#)

PayPal Customer Care

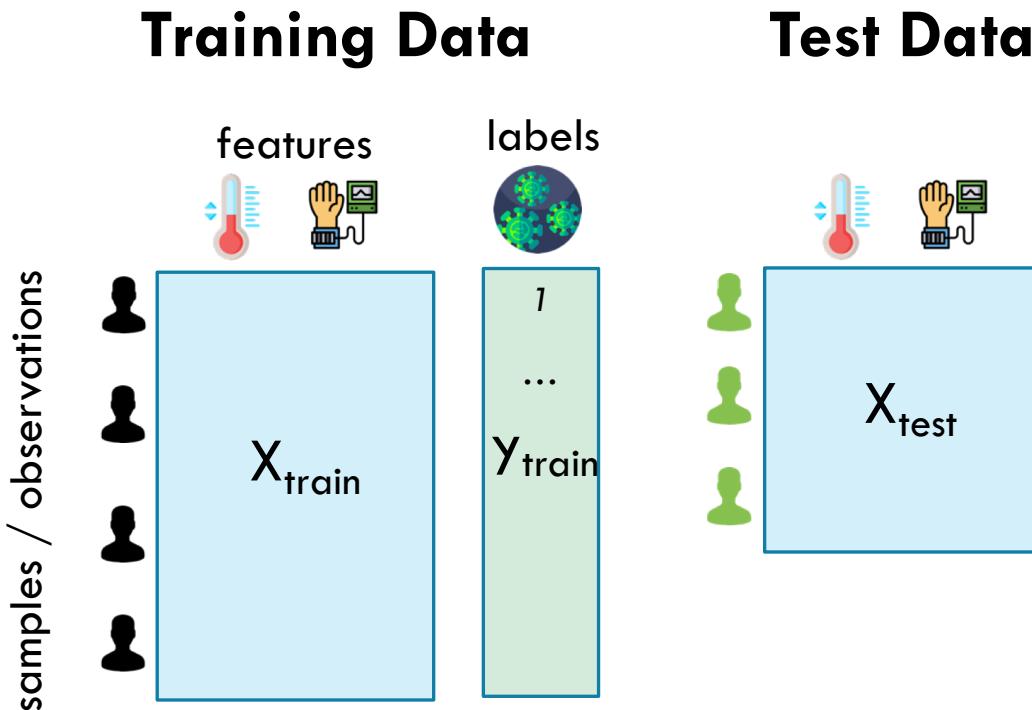
Hi,
Dear customer
At first Thank you for paying attention to PayPal Customer Care.
We contact you for confirming your PayPal account because of security reason
you have to confirm your account in PayPal again. Our log on your account shows
us some illegal usage then we want you to pay some time and Confirm your
account again. for confirming just login to PayPal with attached form just from

Spam Detection



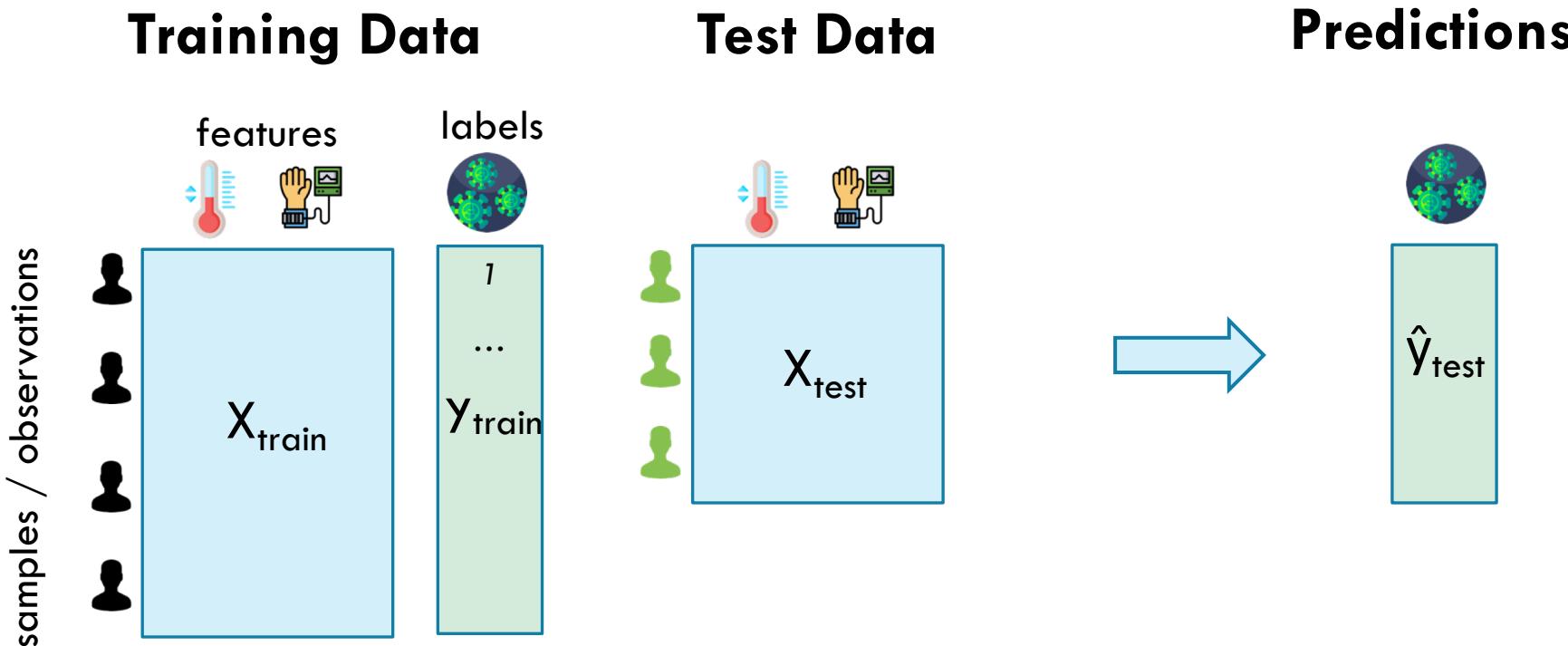
Network Intrusion
Detection

CLASSIFICATION: SETUP



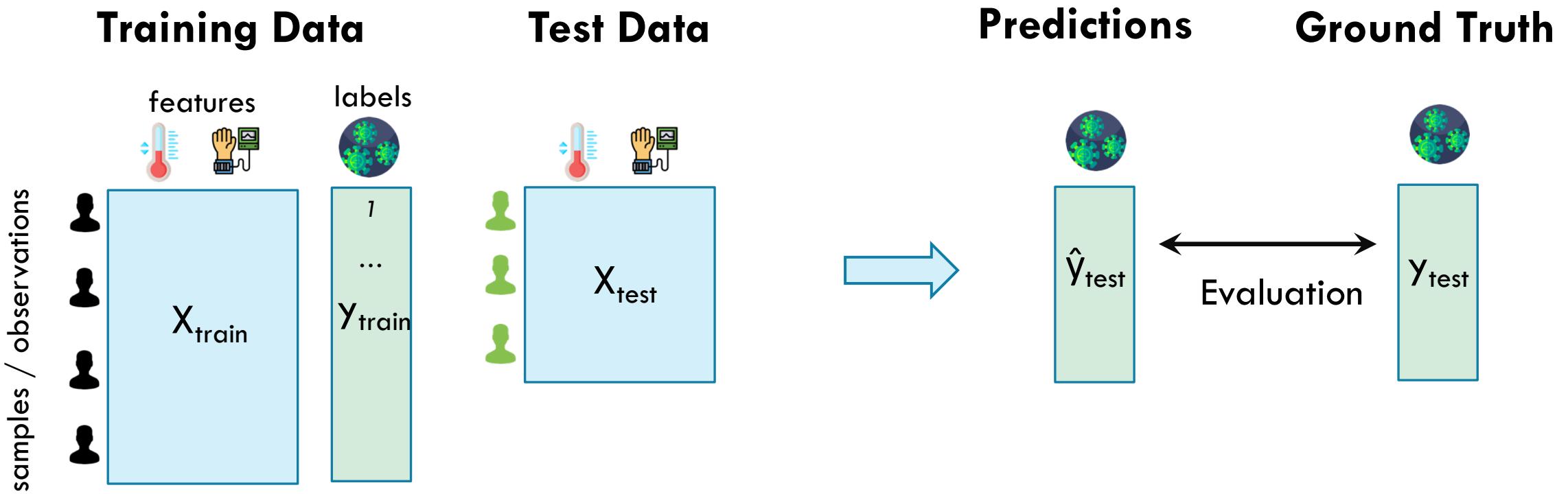
Categorize samples based on a **discrete** label, given training data

CLASSIFICATION: SETUP



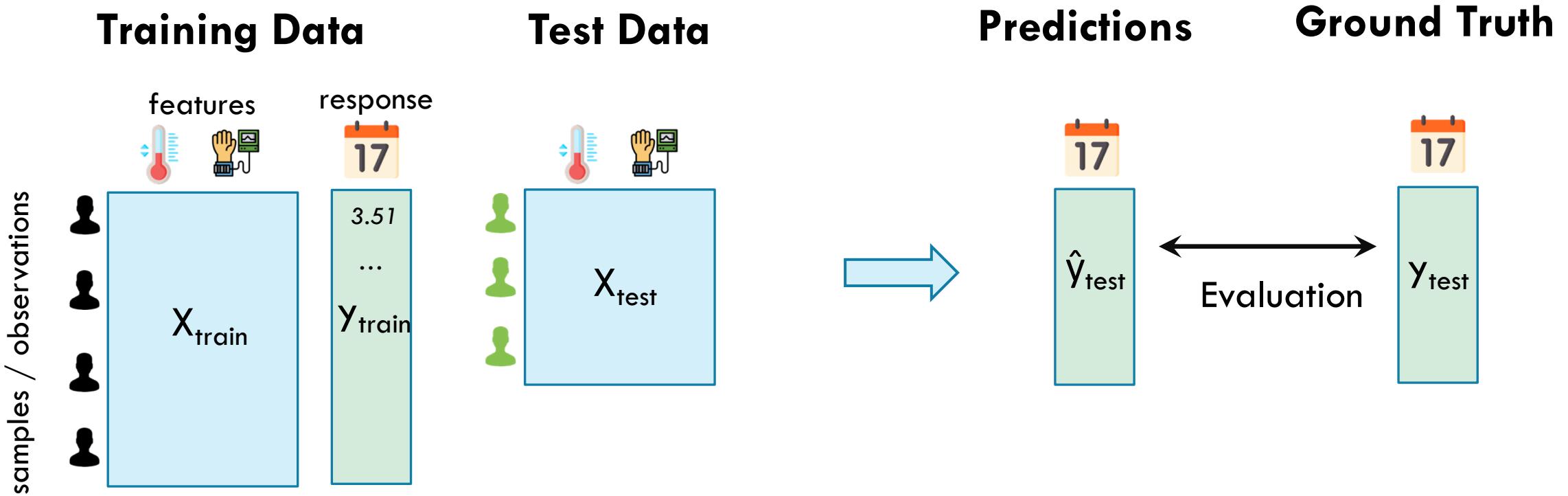
Categorize samples based on a **discrete** label, given training data

CLASSIFICATION: SETUP



Categorize samples based on a **discrete** label, given training data

REGRESSION: SETUP

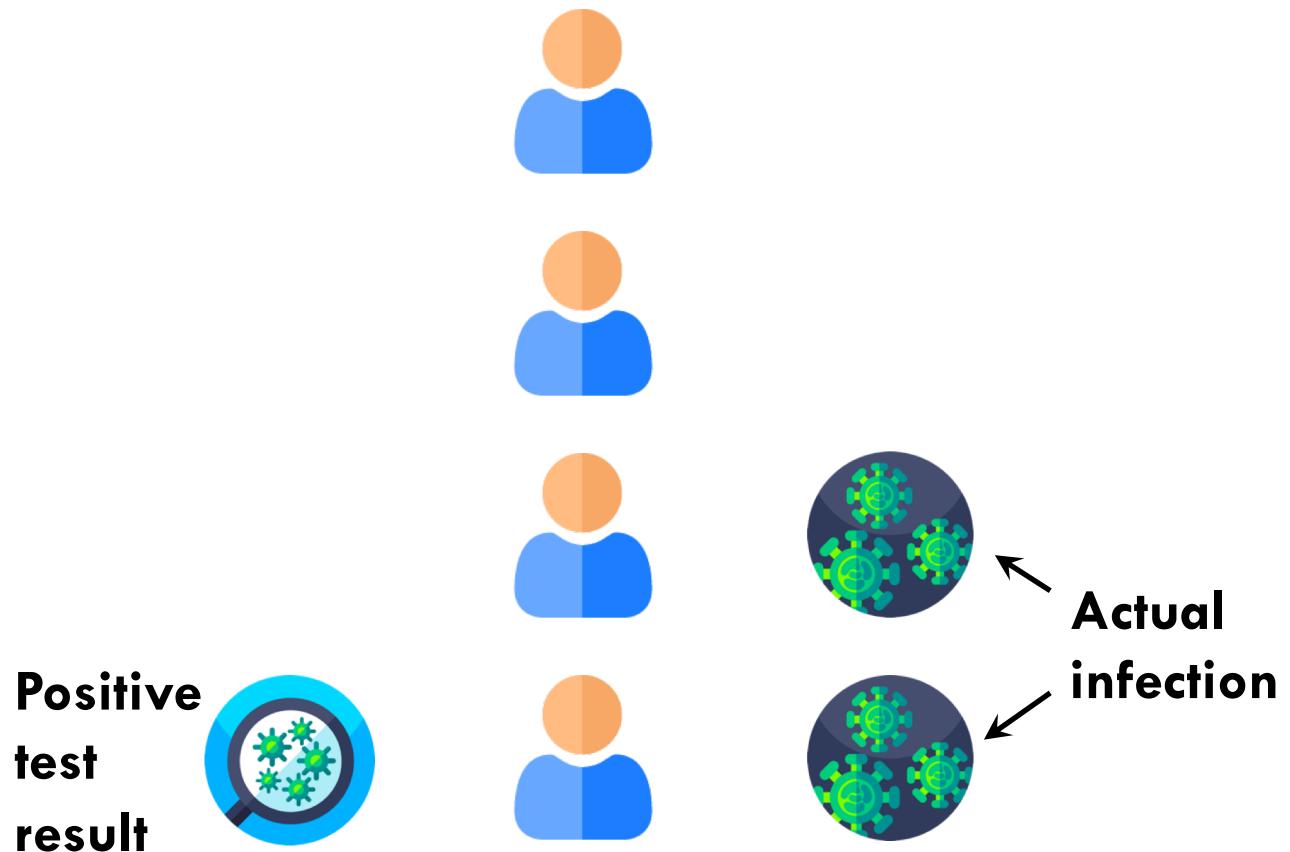


Predict a **numeric** response variable, given training data

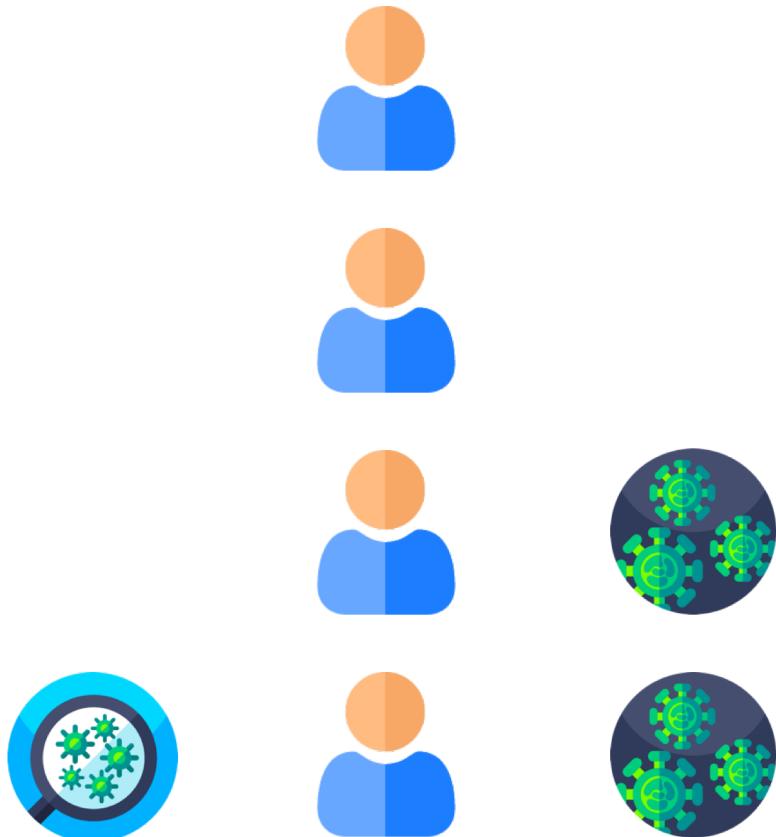
OVERVIEW

1. Problem Setup
2. Evaluating Classifiers
3. Nearest Neighbor Methods
4. Tree-Based Methods
5. Ensemble Methods
6. Logistic Regression
7. Deep Learning

BINARY CLASSIFICATION SETTING

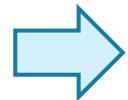


BINARY CLASSIFICATION SETTING



Predicted Label (\hat{y})	Ground Truth Label (y)
0	0
0	0
0	1
1	1

CONFUSION MATRIX

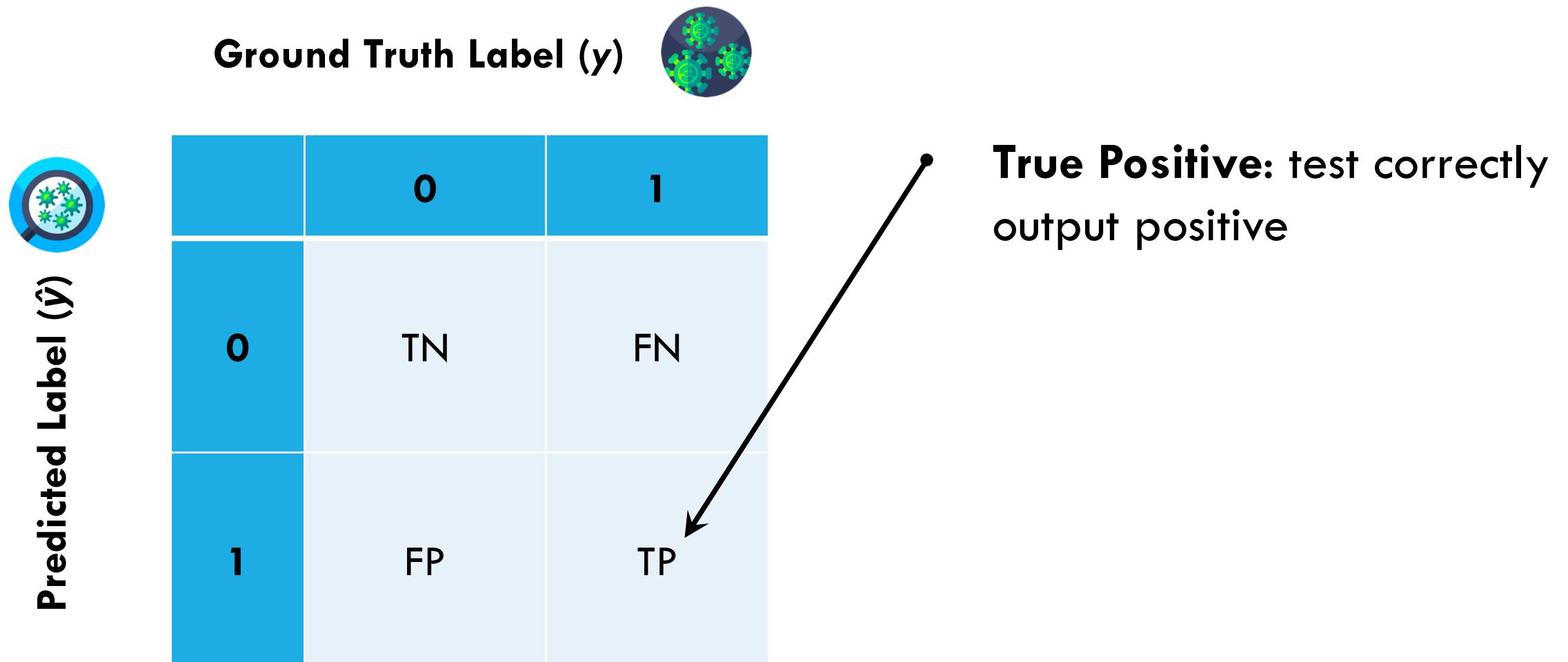


Predicted Label (\hat{y})	Ground Truth Label (y)
0	0
0	0
0	1
1	1

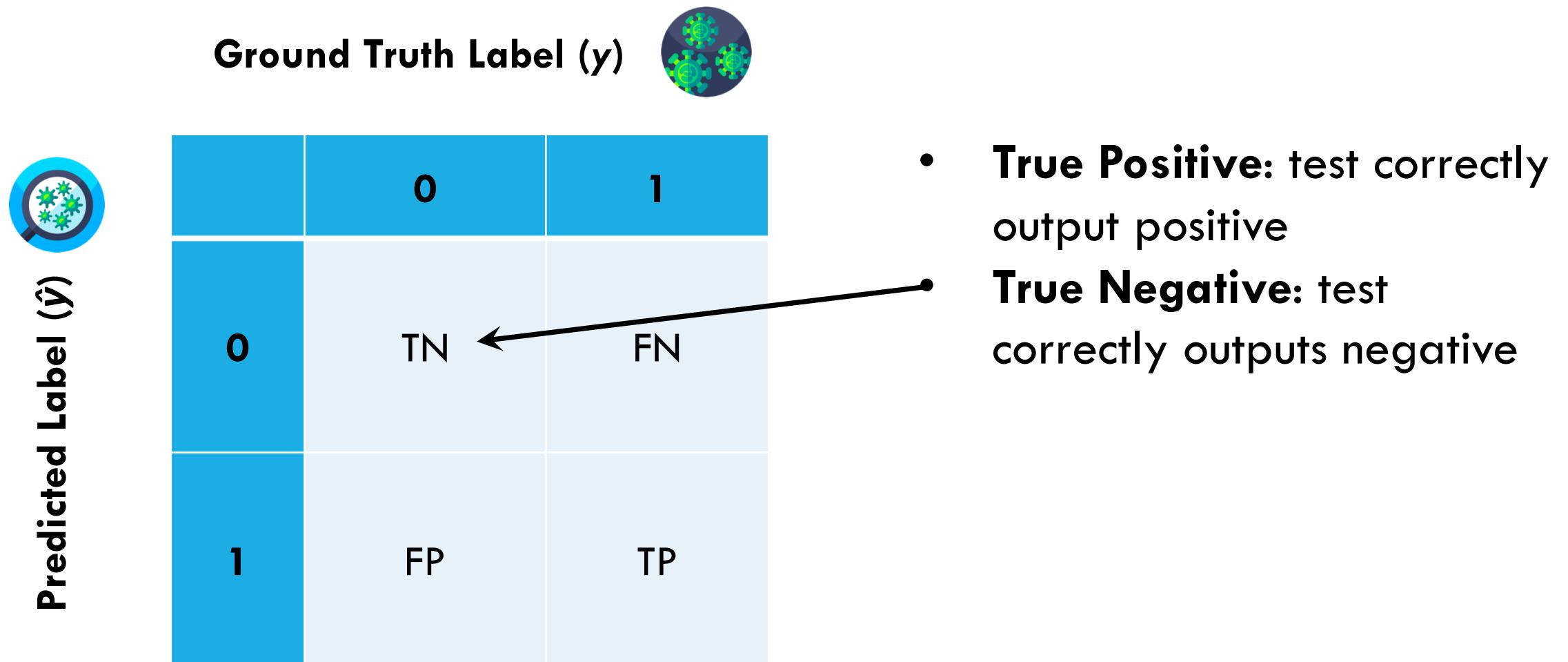
Ground Truth Label (y)

	0	1
0	2	0
1	1	1

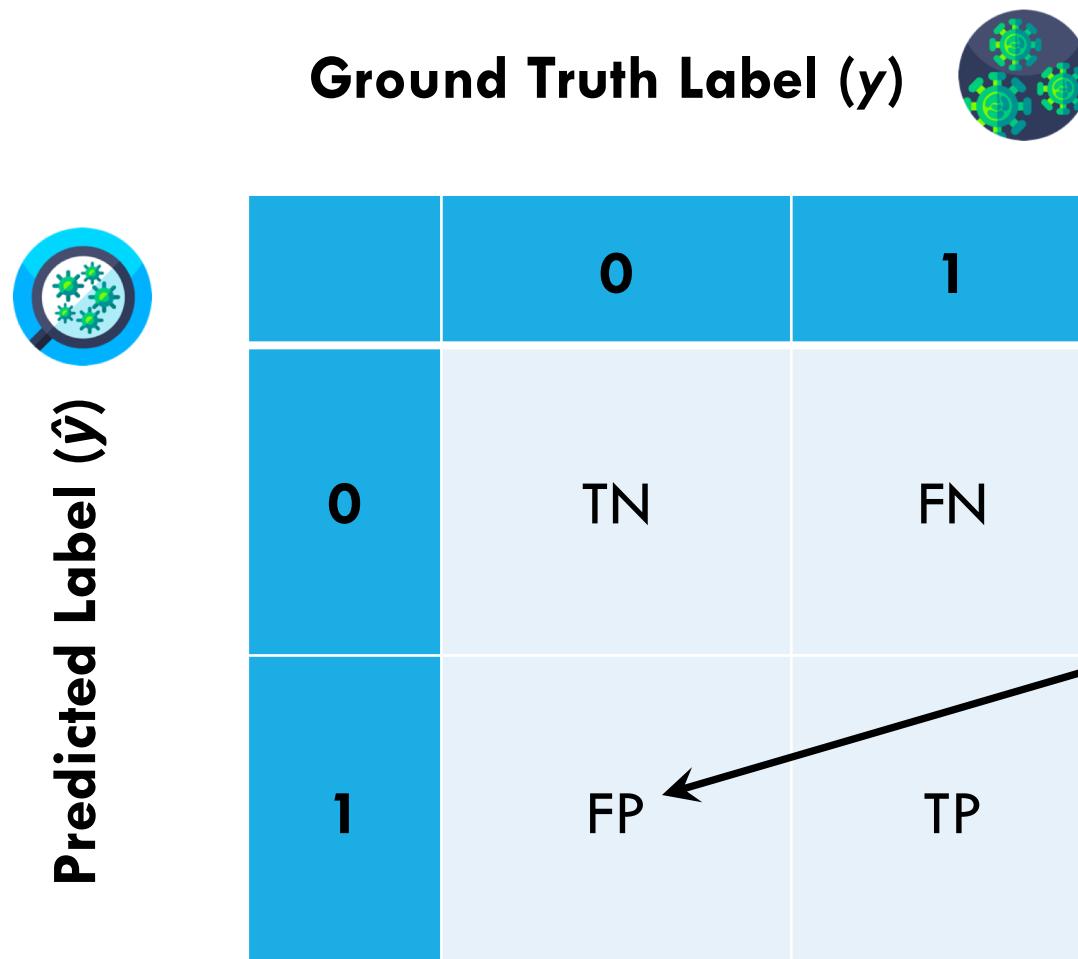
FALSE POSITIVES, FALSE NEGATIVES, ETC



FALSE POSITIVES, FALSE NEGATIVES, ETC



FALSE POSITIVES, FALSE NEGATIVES, ETC



- **True Positive:** test correctly output positive
- **True Negative:** test correctly outputs negative
- **False Positive:** test wrongly outputs positive

FALSE POSITIVES, FALSE NEGATIVES, ETC

		Ground Truth Label (y)	
		0	1
Predicted Label (\hat{y})	0	TN	FN
	1	FP	TP

- **True Positive:** test correctly output positive
- **True Negative:** test correctly outputs negative
- **False Positive:** test wrongly outputs positive
- **False Negative:** test wrongly outputs negative

FALSE POSITIVES, FALSE NEGATIVES, ETC

		Ground Truth Label (y)	
		0	1
Predicted Label (\hat{y})	0	TN	FN
	1	FP	TP

N = negatives **P = positives**

= TN + FP **= FN + TP**

- **True Positive:** test correctly output positive
- **True Negative:** test correctly outputs negative
- **False Positive:** test wrongly outputs positive
- **False Negative:** test wrongly outputs negative



QUIZ: FALSE POSITIVES & NEGATIVES

Q: Suppose a positive test almost always correctly indicates the disease; but, the test often misses the disease. How would you describe the (proportion of) false positives and negatives?

1. High false positives, high false negatives
2. High false positives, low false negatives
3. Low false positives, high false negatives
4. Low false positives, low false negatives

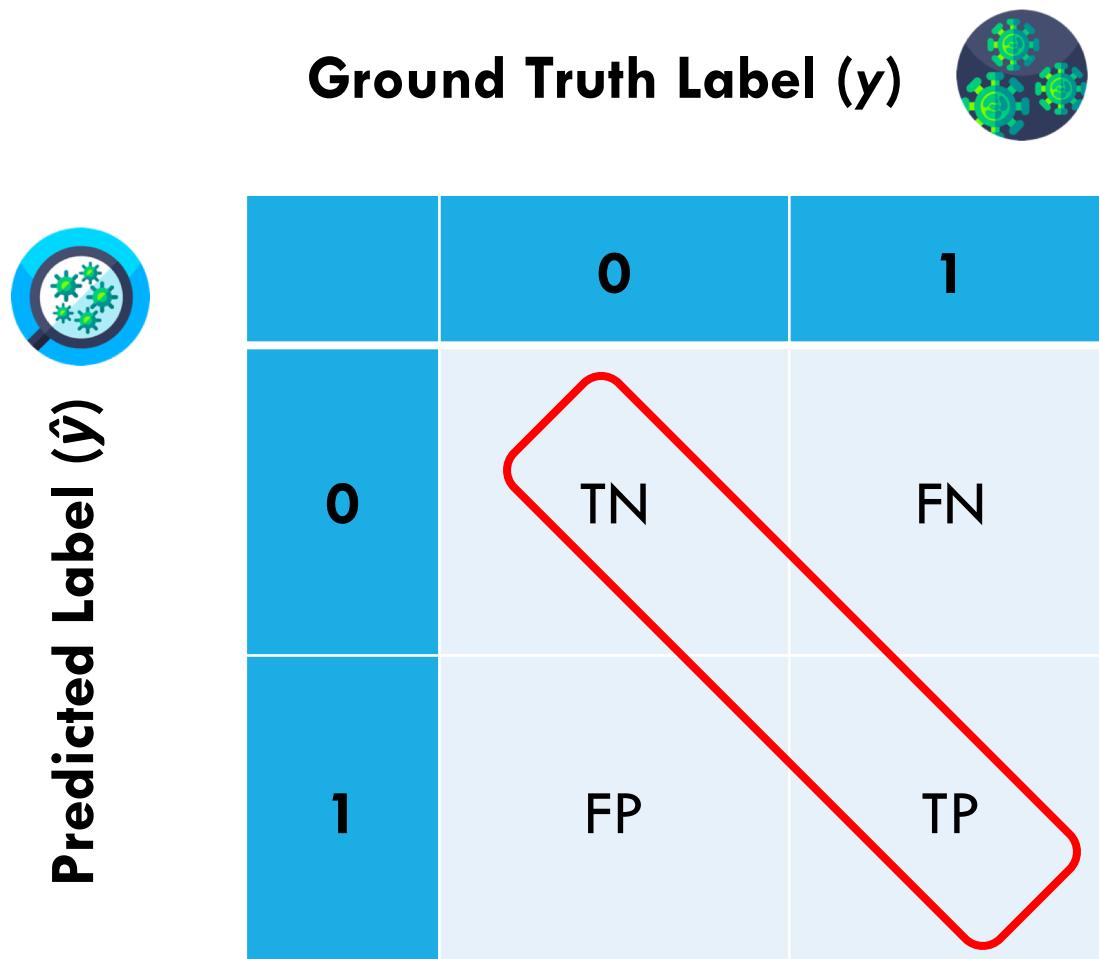


QUIZ: FALSE POSITIVES & NEGATIVES

Q: Suppose a positive test almost always correctly indicates the disease, but, the test often misses the disease. How would you describe the (proportion of) false positives and negatives?

1. High false positives, high false negatives
2. High false positives, low false negatives
3. Low false positives, high false negatives
4. Low false positives, low false negatives

ACCURACY



Accuracy: fraction of predictions that are correct

$$= (TN + TP) / (TN + FN + FP + TP)$$

PRECISION

		Ground Truth Label (y)	
		0	1
Predicted Label (\hat{y})	0	TN	FN
	1	FP	TP

Precision: fraction of positive test results that are correct

$$= \text{TP} / (\text{TP} + \text{FP})$$

Interpretation: if the test returns 1, how likely is the patient to be actually sick?

RECALL, OR SENSITIVITY

		Ground Truth Label (y)	
		0	1
Predicted Label (\hat{y})	0	TN	FN
	1	FP	TP

Recall: fraction of actual positives that are detected

$$= \text{TP} / (\text{TP} + \text{FN})$$

Interpretation: if a patient is sick, how likely is the test to detect this?

F-MEASURE / F1 SCORE

		Ground Truth Label (y)	
		0	1
Predicted Label (\hat{y})	0	TN	FN
	1	FP	TP

A magnifying glass icon with a virus inside is positioned next to the column header 'Ground Truth Label (y)'.

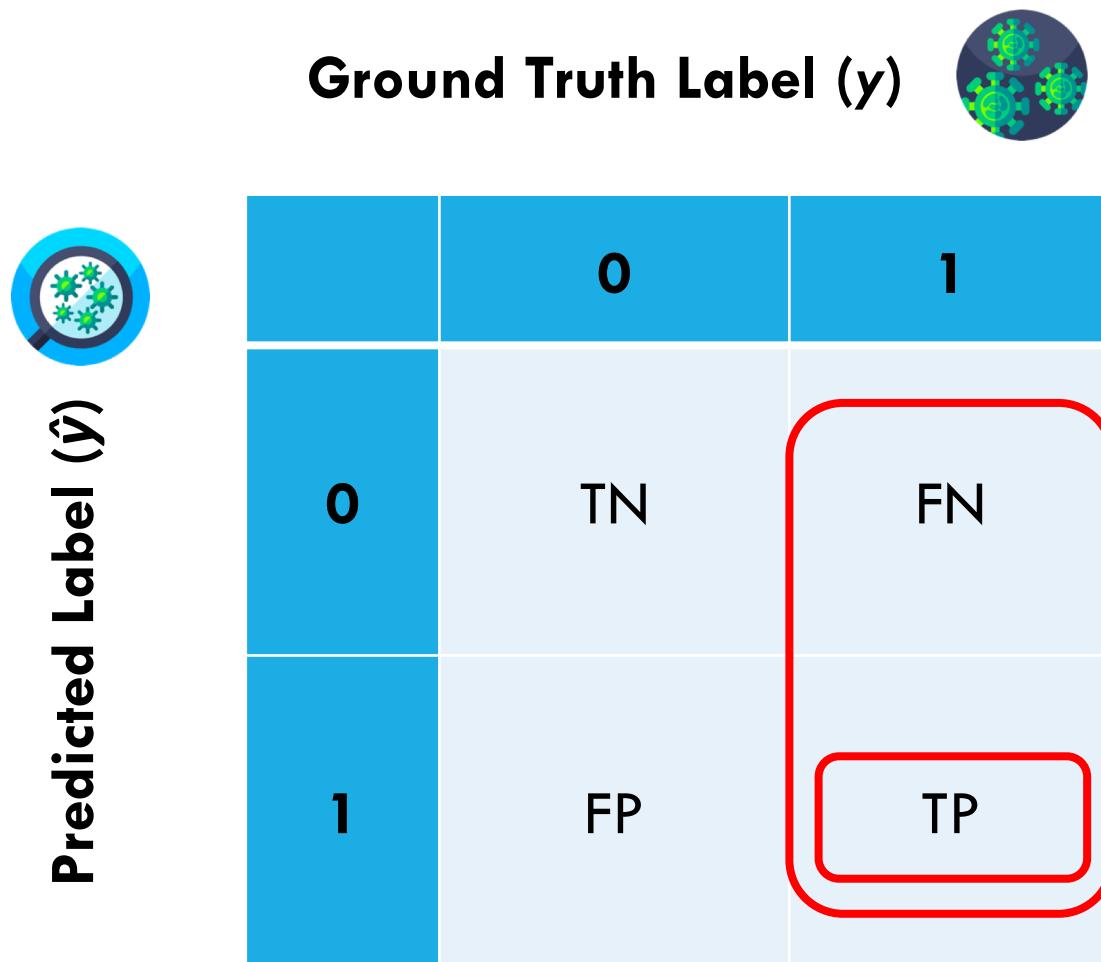
F-Measure:

$$= 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Interpretation: it is a ‘pessimistic average’ of precision and recall:

- i.e. F-measure is always in between precision and recall, and closer to whichever of them is lower
- E.g. if precision = 1, recall = 0.1, then F1=0.18
- Formally, F-measure is the **harmonic mean** of precision and recall

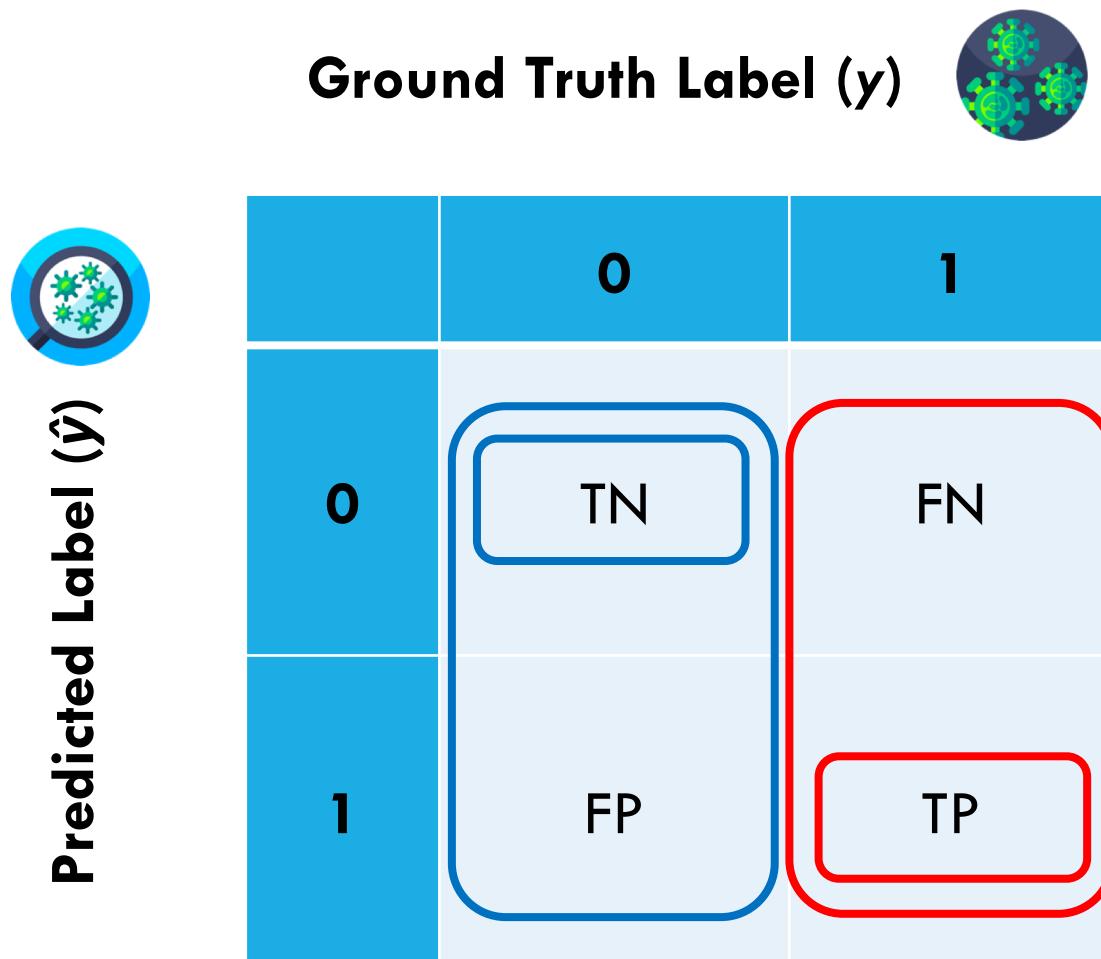
SENSITIVITY AND SPECIFICITY



Sensitivity: same as recall (fraction of actual positives that are correctly identified)

$$= \text{TP} / (\text{TP} + \text{FN})$$

SENSITIVITY AND SPECIFICITY



Sensitivity: same as recall (fraction of actual positives that are correctly identified)

$$= \text{TP} / (\text{TP} + \text{FN})$$

Specificity: fraction of actual negatives that are correctly identified

$$= \text{TN} / (\text{TN} + \text{FP})$$

SENSITIVITY AND SPECIFICITY

		Ground Truth Label (y)	
		0	1
Predicted Label (\hat{y})	0	TN	FN
	1	FP	TP

Sensitivity: same as recall (fraction of actual positives that are correctly identified)

$$= \text{TP} / (\text{TP} + \text{FN})$$

Specificity: fraction of actual negatives that are correctly identified

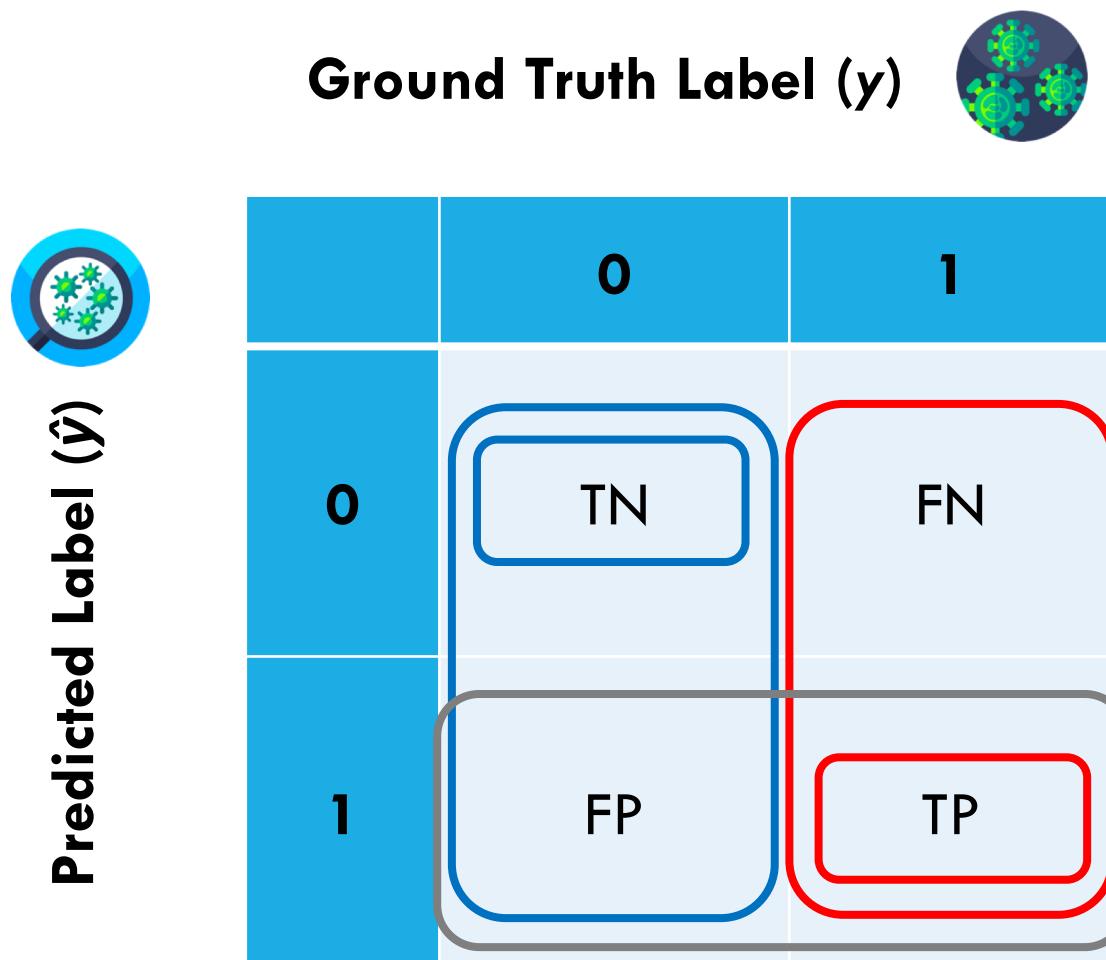
$$= \text{TN} / (\text{TN} + \text{FP})$$

Interpretation:

Sensitivity = accuracy among positives ($y=1$)

Specificity = accuracy among negatives ($y=0$)

SENSITIVITY AND SPECIFICITY



Sensitivity: same as recall (fraction of actual positives that are correctly identified)

$$= \text{TP} / (\text{TP} + \text{FN})$$

Specificity: fraction of actual negatives that are correctly identified

$$= \text{TN} / (\text{TN} + \text{FP})$$

Interpretation:

Sensitivity = accuracy among positives ($y=1$)

Specificity = accuracy among negatives ($y=0$)

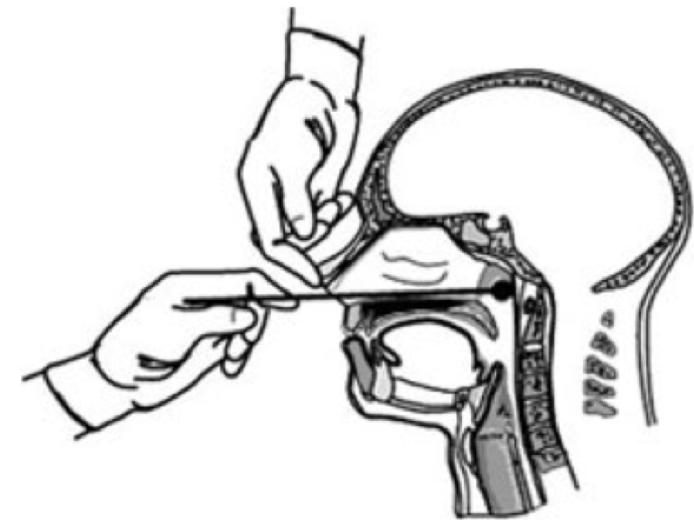
Precision = accuracy among detected ($\hat{y}=1$)



QUIZ: SENSITIVITY & SPECIFICITY

Q: RT-PCR is a common test for COVID-19. A study [1] found that 78.2% of actual COVID-19 cases are correctly detected, while among patients without COVID-19, positive tests occur only 1.2% of the time. What is the recall (or sensitivity)?

1. 0.782
2. 0.218
3. 0.012
4. 0.988

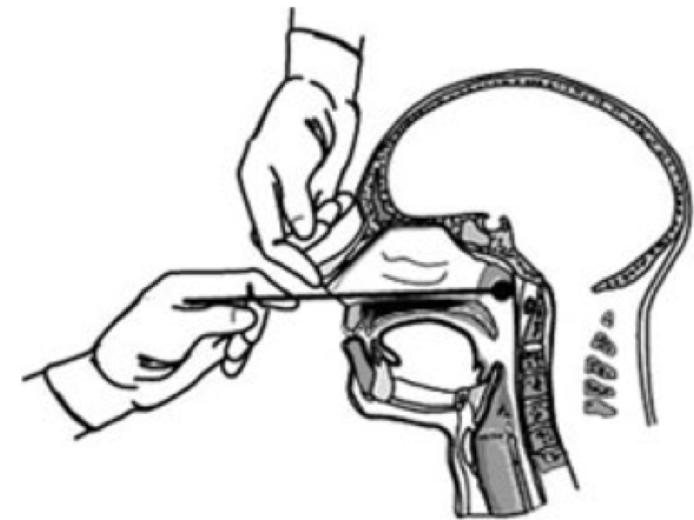




QUIZ: SENSITIVITY & SPECIFICITY

Q: RT-PCR is a common test for COVID-19. A study [1] found that 78.2% of actual COVID-19 cases are correctly detected, while among patients without COVID-19, positive tests occur only 1.2% of the time. What is the recall (or sensitivity)?

1. 0.782
2. 0.218
3. 0.012
4. 0.988

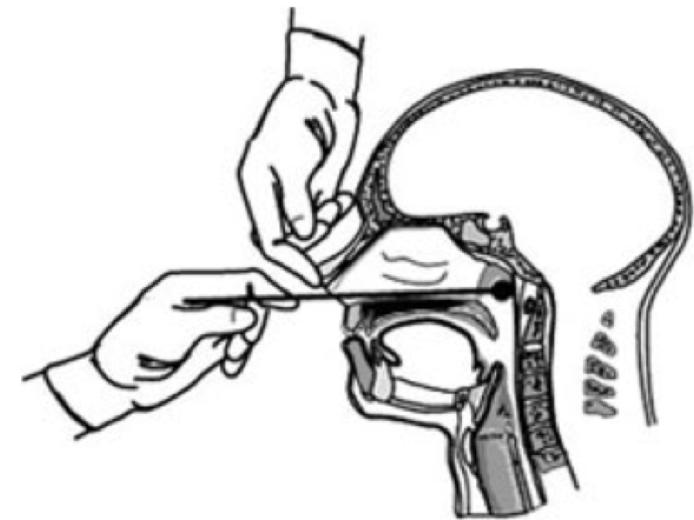




QUIZ: SENSITIVITY & SPECIFICITY

Q: RT-PCR is a common test for COVID-19. A study [1] found that 78.2% of actual COVID-19 cases are correctly detected, while among patients without COVID-19, positive tests occur only 1.2% of the time. What is the specificity?

1. 0.782
2. 0.218
3. 0.012
4. 0.988

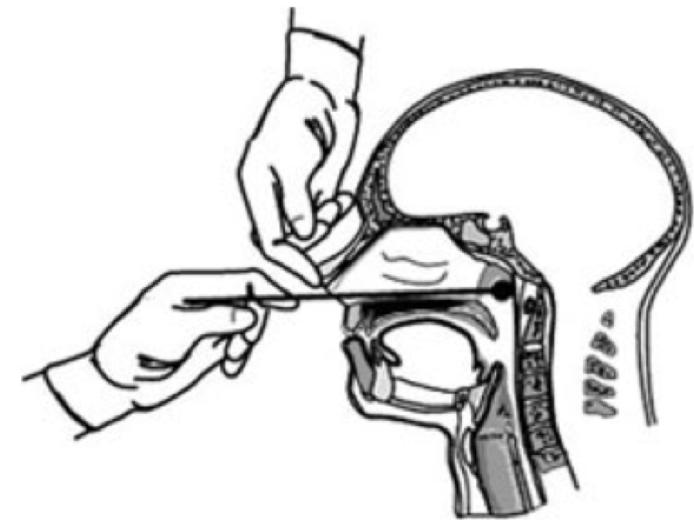




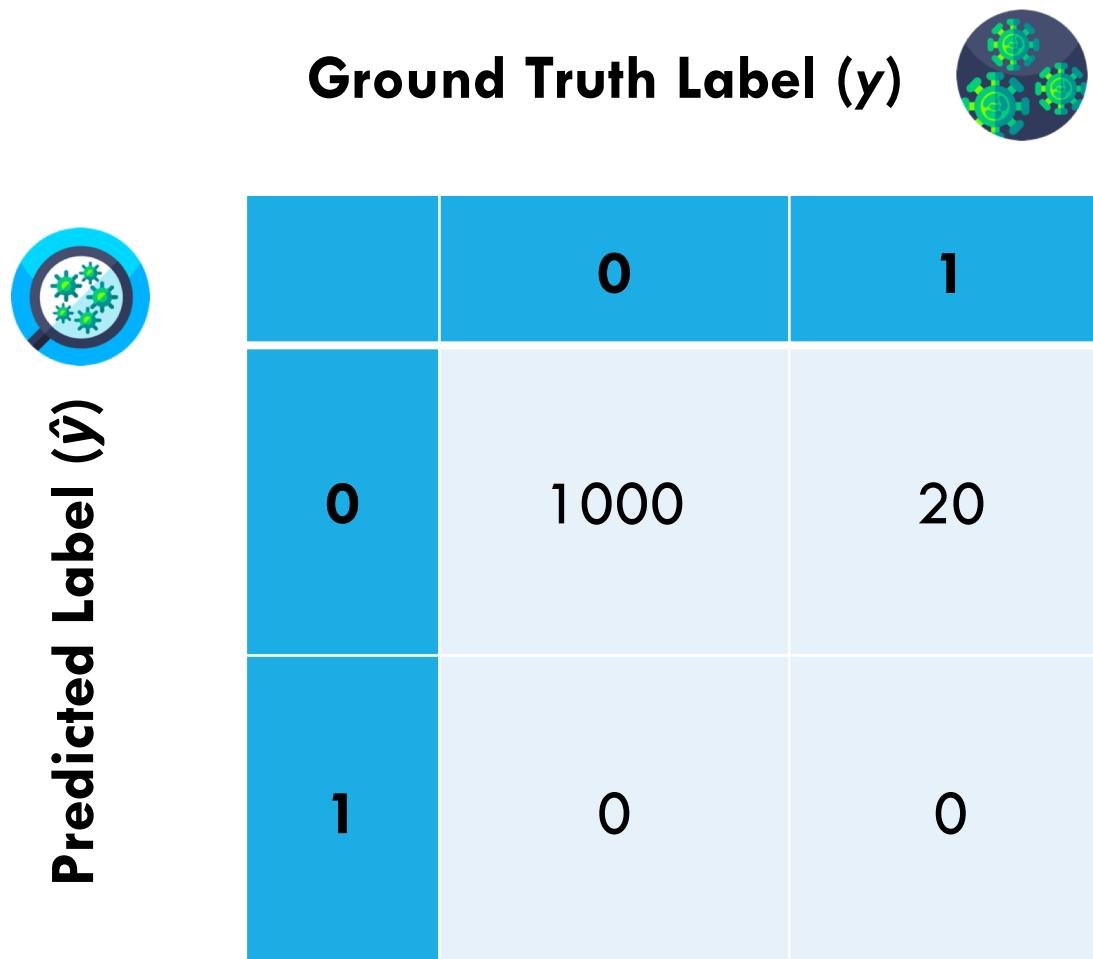
QUIZ: SENSITIVITY & SPECIFICITY

Q: RT-PCR is a common test for COVID-19. A study [1] found that 78.2% of actual COVID-19 cases are correctly detected, while among patients without COVID-19, positive tests occur only 1.2% of the time. What is the specificity?

1. 0.782
2. 0.218
3. 0.012
4. 0.988



PROBLEM: HIGHLY UNBALANCED DATA?



In real data, **highly unbalanced data** is common: e.g. most people don't have any given disease.



PROBLEM: HIGHLY UNBALANCED DATA?

Ground Truth Label (y)



		0	1
Predicted Label (\hat{y})	0	1000	20
	1	0	0

In real data, **highly unbalanced data** is common: e.g. most people don't have any given disease.

In the example on the left, how would you describe the accuracy and the specificity?

1. High accuracy, high specificity
2. High accuracy, low specificity
3. Low accuracy, high specificity
4. Low accuracy, low specificity



PROBLEM: HIGHLY UNBALANCED DATA?

Ground Truth Label (y)



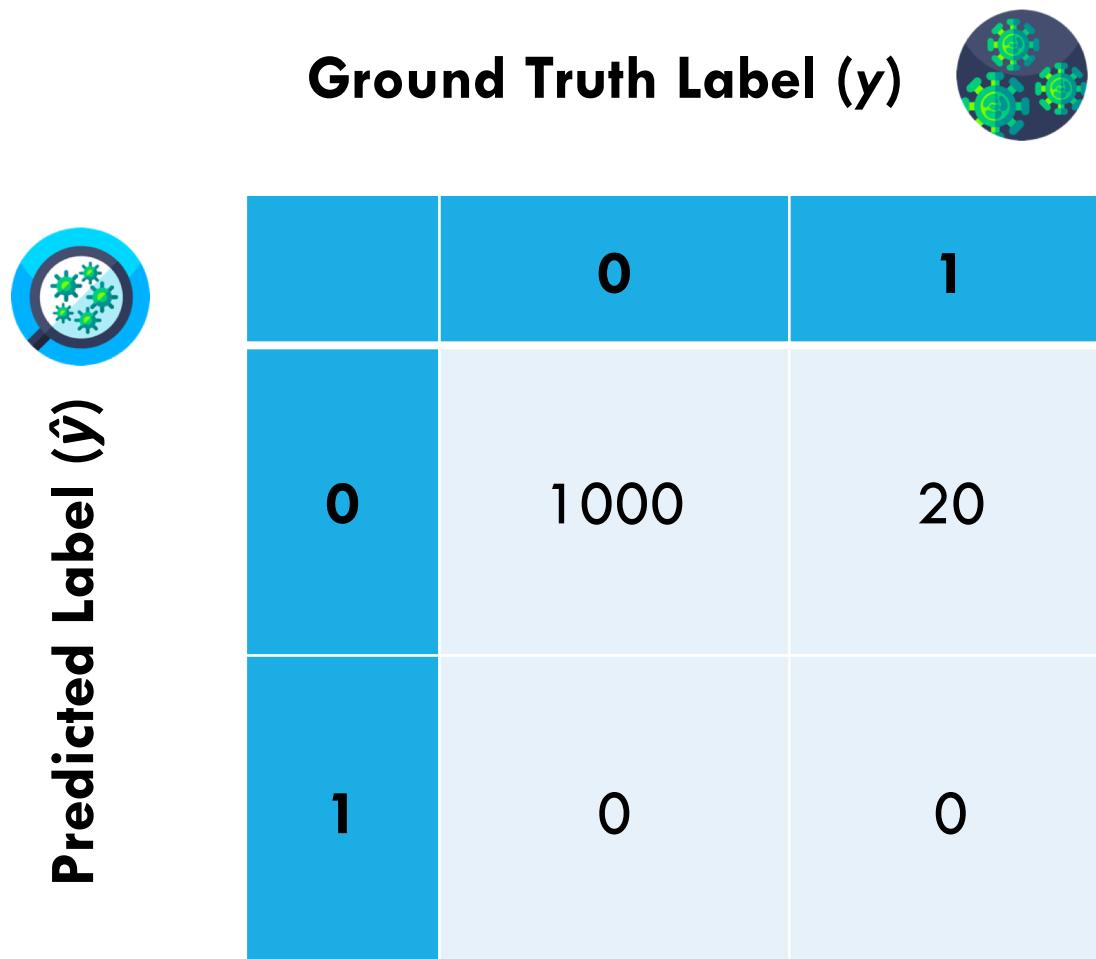
Predicted Label (\hat{y})	0	1
0	1000	20
1	0	0

In real data, **highly unbalanced data** is common: e.g. most people don't have any given disease.

In the example on the left, how would you describe the accuracy and the specificity?

1. High accuracy, high specificity
2. High accuracy, low specificity
3. Low accuracy, high specificity
4. Low accuracy, low specificity

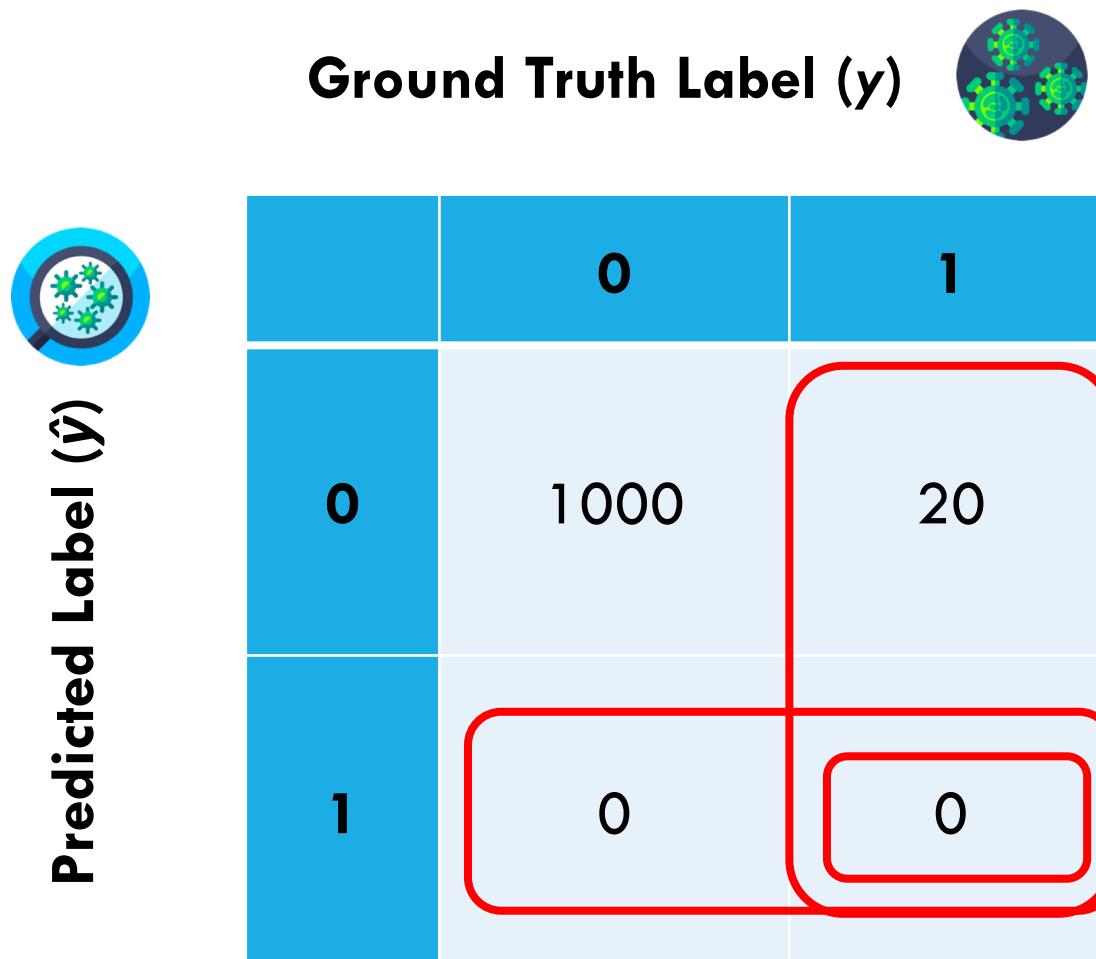
PROBLEM: HIGHLY UNBALANCED DATA?



In real data, **highly unbalanced data** is common: e.g. most people don't have any given disease.

The test on the left shows **very high accuracy and perfect specificity**, despite a test that always returns negative!

PROBLEM: HIGHLY UNBALANCED DATA?



Main problem is that the top-left entry can be extremely large (i.e. many true negatives)

Solution: for unbalanced data, use precision and recall, which ignore the true negatives

Use F-measure instead of accuracy

EVALUATING NUMERIC SCORES

In some cases, our “predictions” are in the form of numeric scores.

Examples:

- Tests may output a “risk score” instead of a 0 or 1 prediction
- Data mining algorithms often output probabilities as their prediction; i.e. $P(y=1)$



Numeric prediction (\hat{y})	Ground Truth Label (y)
0.05	0
0.1	0
0.6	1
0.9	1

THRESHOLDING

Given any threshold (e.g. 0.5), we can convert \hat{y} to a binary variable by setting everything above the threshold to 1

Problem: hard to know what threshold to use

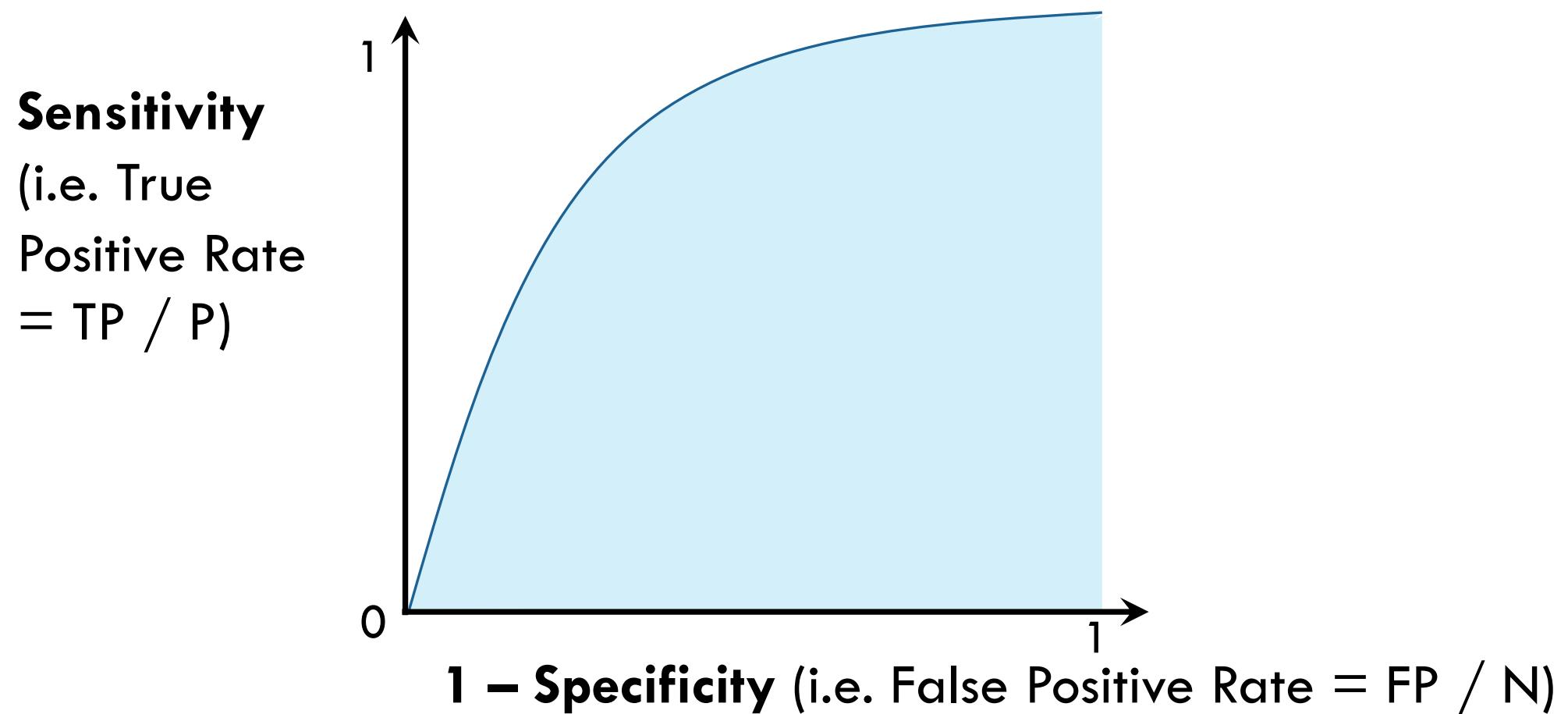
Solution: try all thresholds and plot the resulting curve!



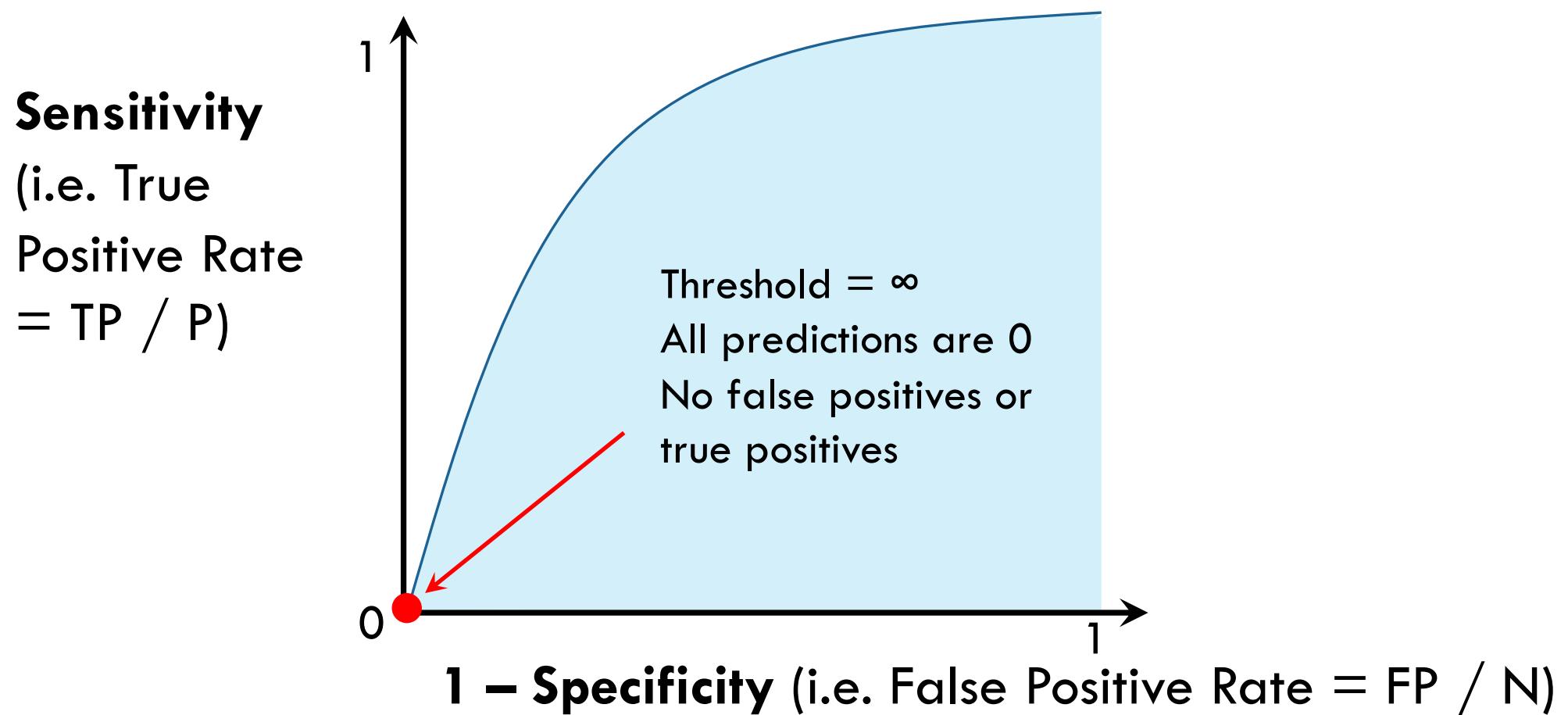
Numeric prediction (\hat{y})	Ground Truth Label (y)
0.05 → 0	0
0.1 → 0	0
0.6 → 1	1
0.9 → 1	1

threshold = 0.5

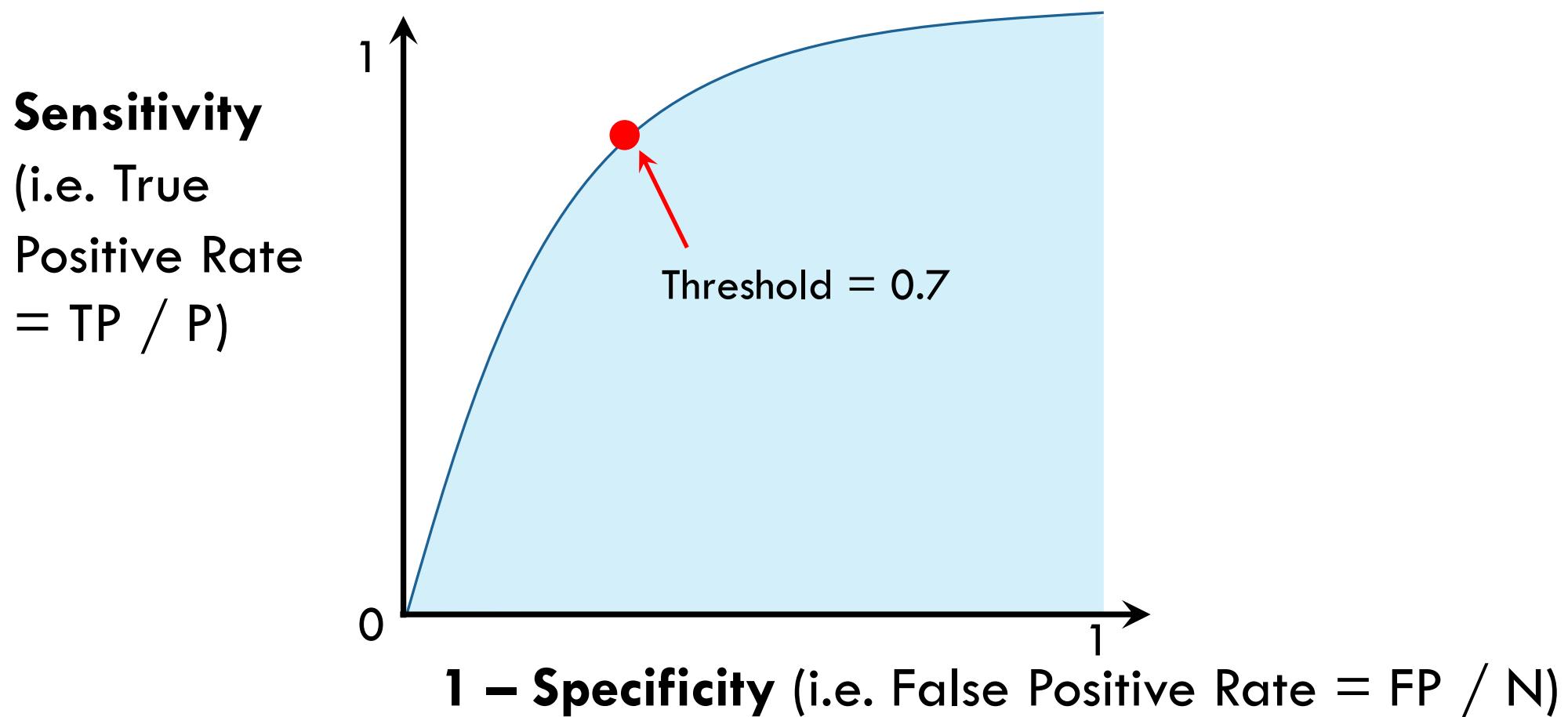
RECEIVER-OPERATING CHARACTERISTIC (ROC) CURVE



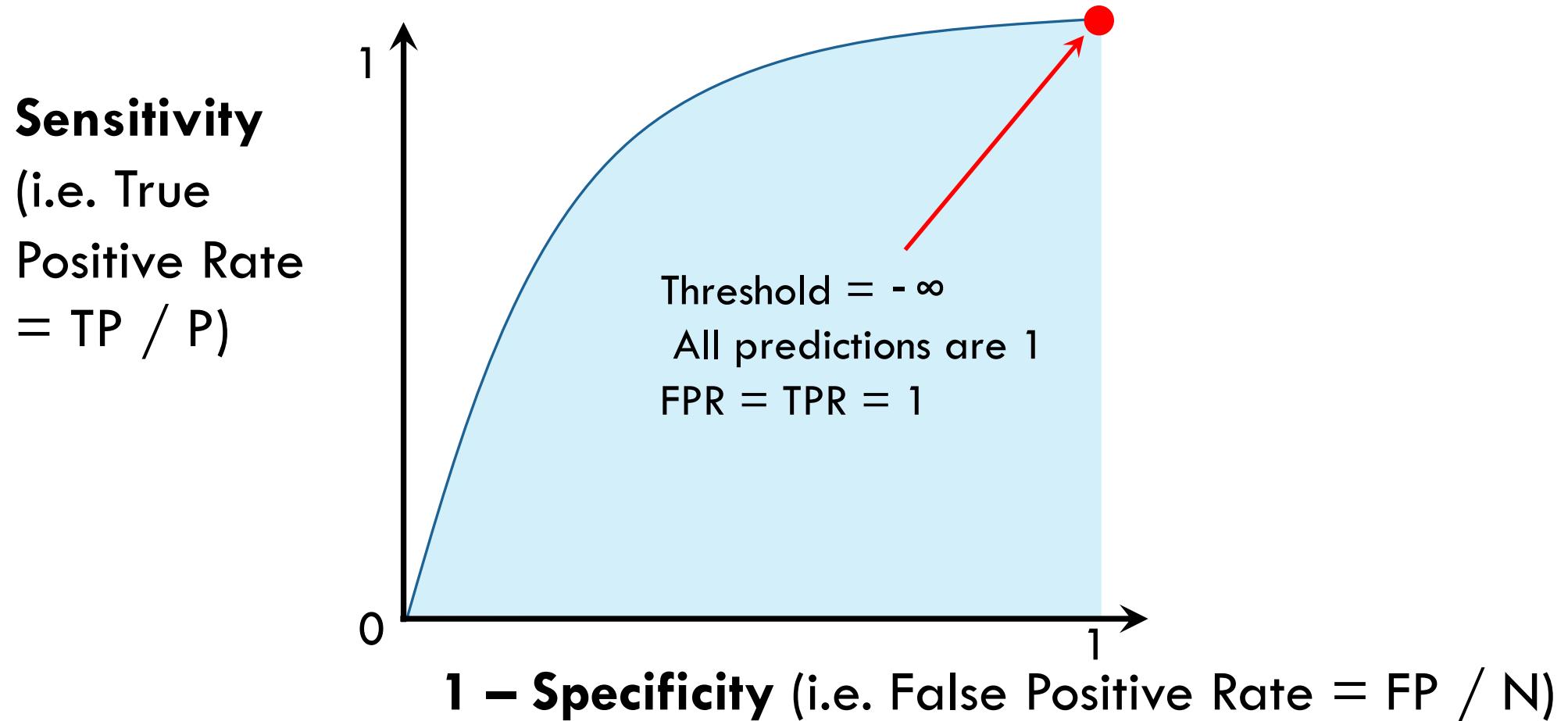
RECEIVER-OPERATING CHARACTERISTIC (ROC) CURVE



RECEIVER-OPERATING CHARACTERISTIC (ROC) CURVE



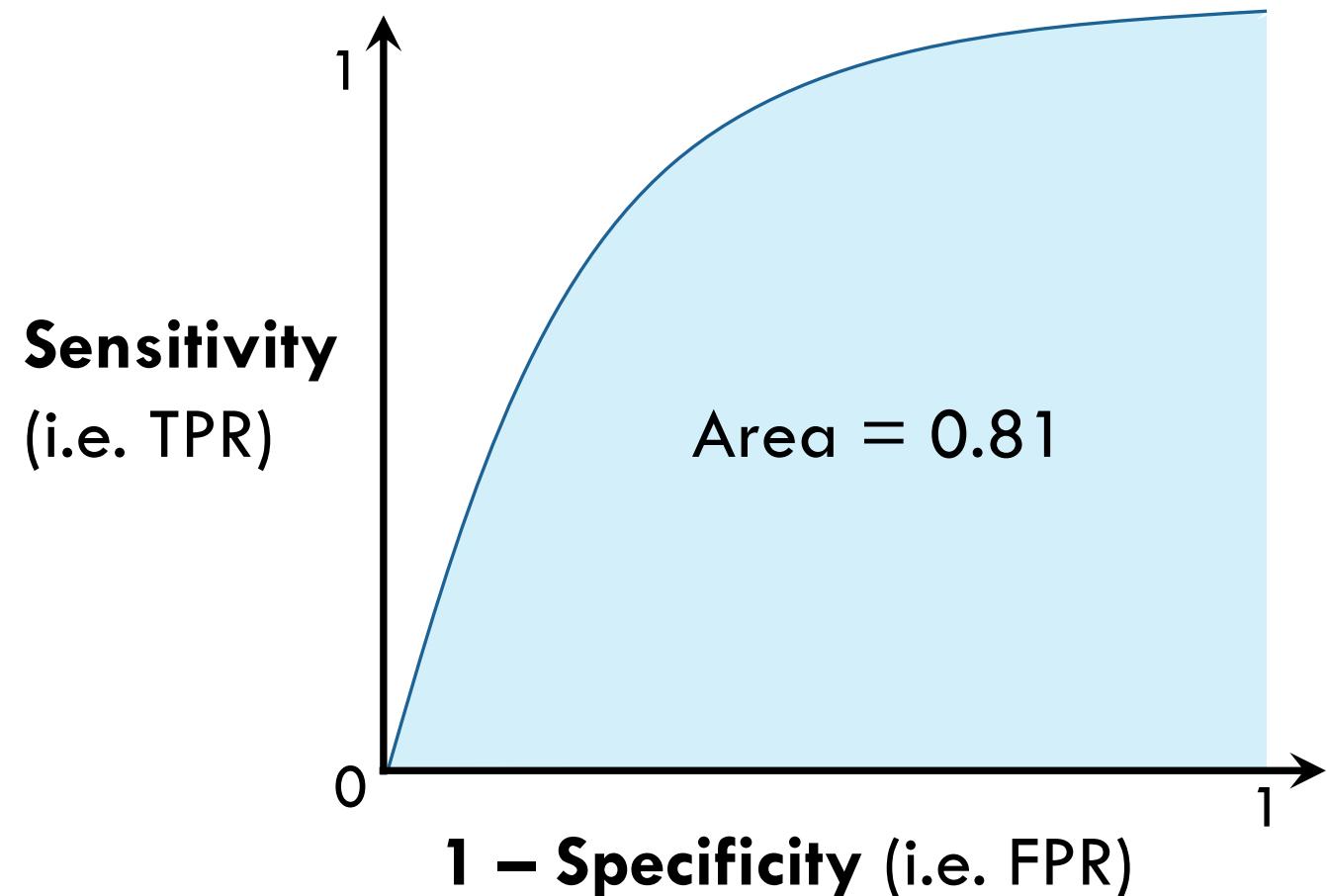
RECEIVER-OPERATING CHARACTERISTIC (ROC) CURVE



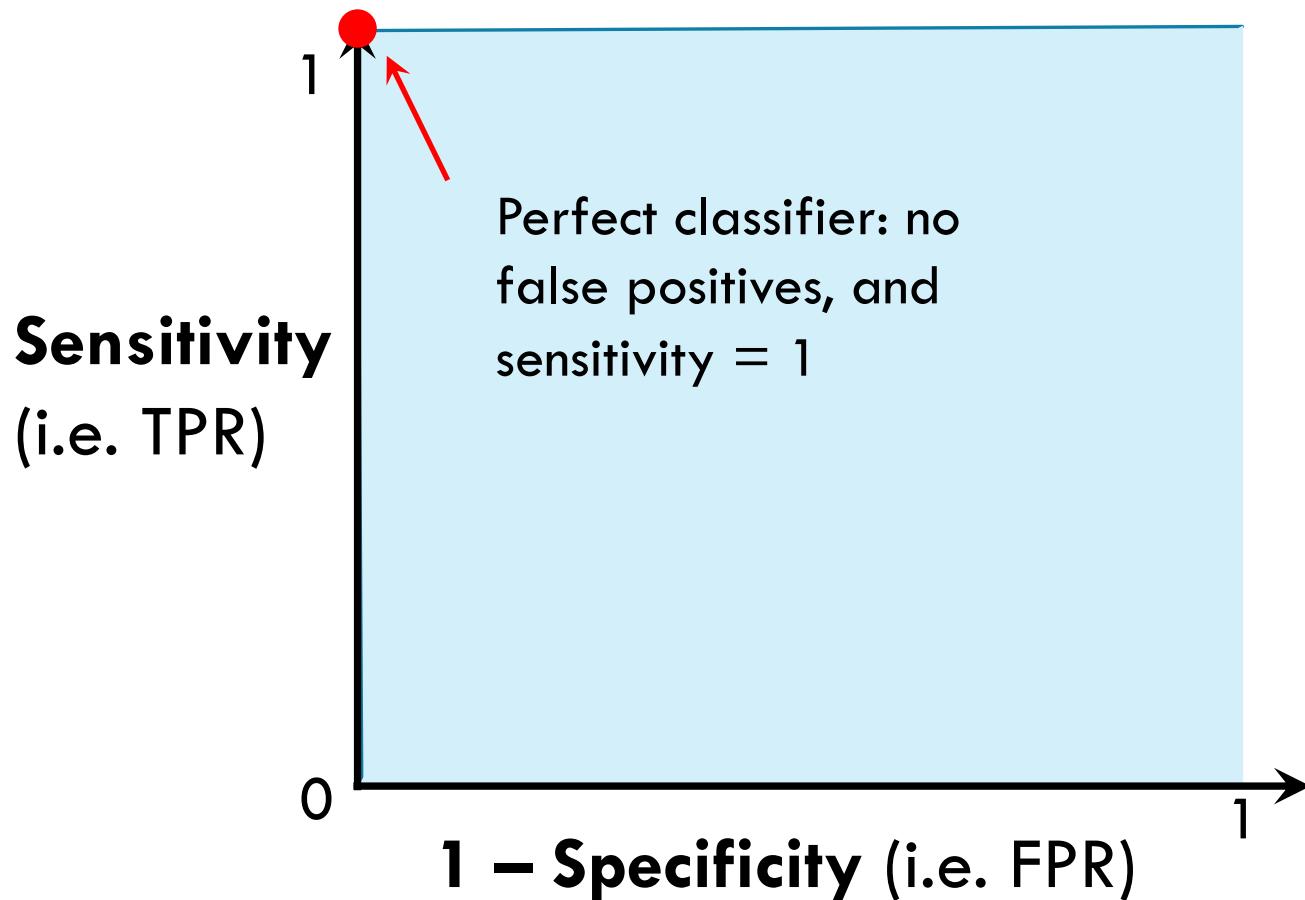
AREA UNDER RECEIVER-OPERATING CHARACTERISTIC (AUROC, OR AUC)

AUROC is defined as the **area under the ROC curve**

AUROC is between 0 and 1, with higher values meaning better detection

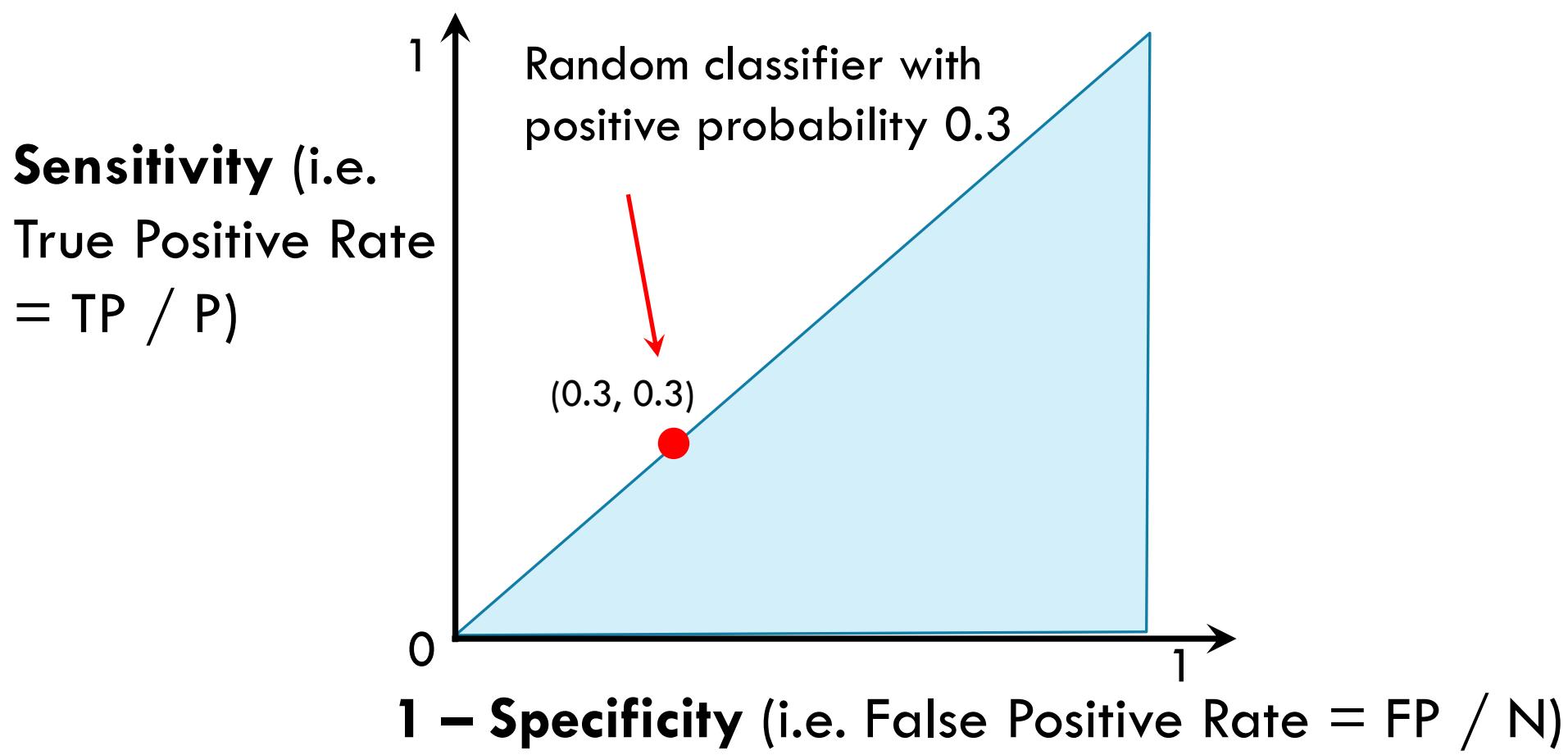


AUROC OF PERFECT CLASSIFIER = 1

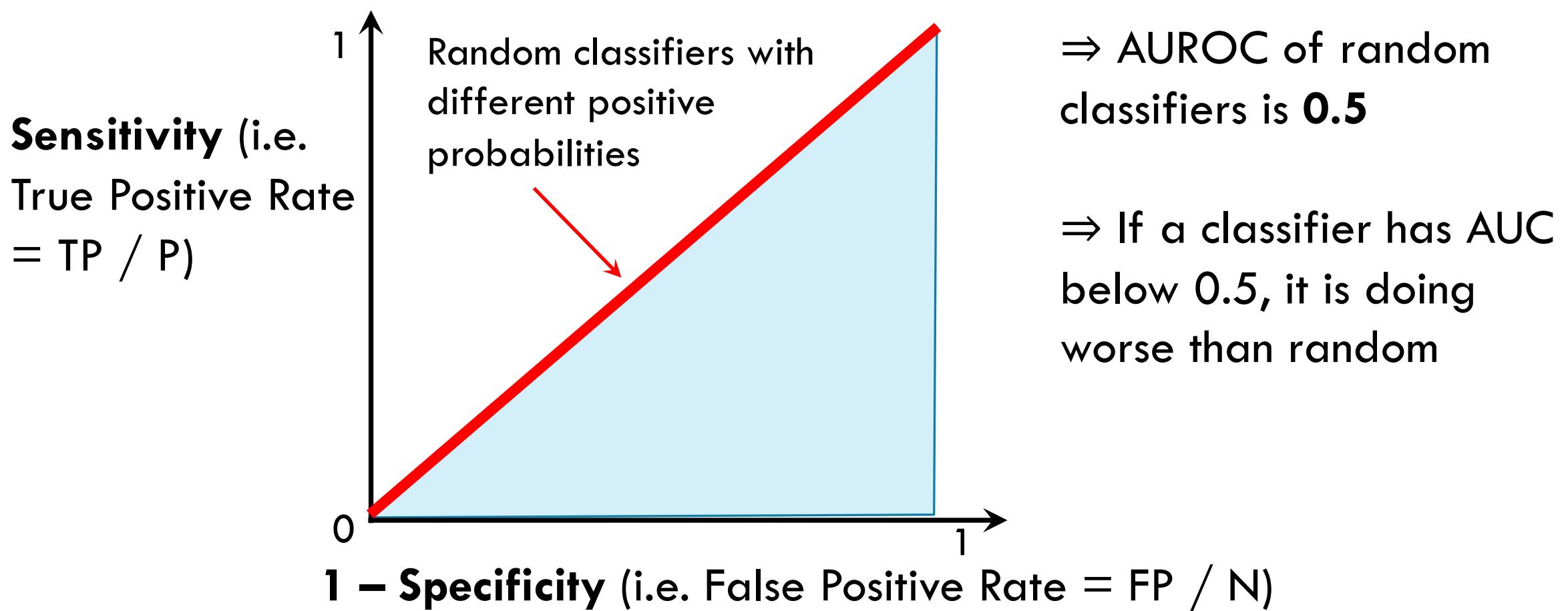


Maximum AUROC is 1, indicating perfect classifier

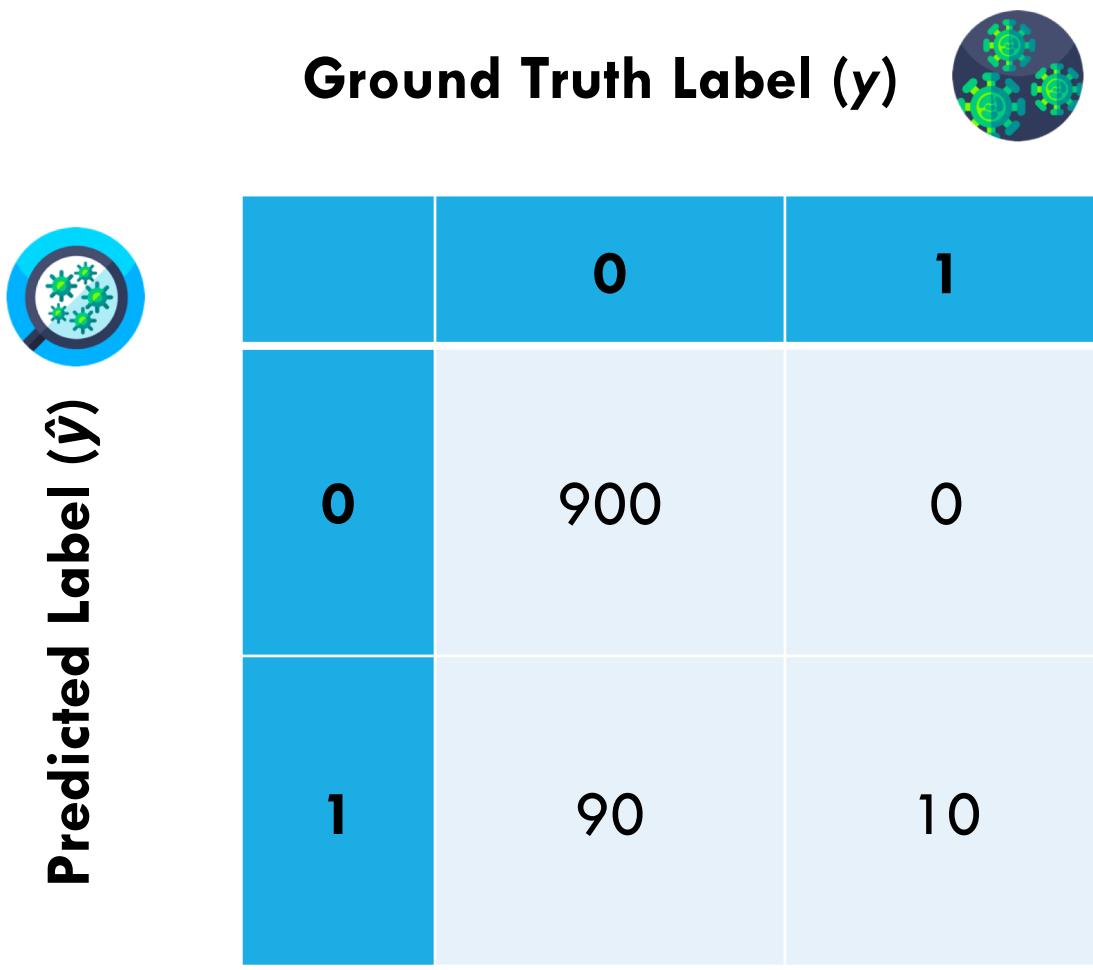
AUROC OF RANDOM CLASSIFIERS



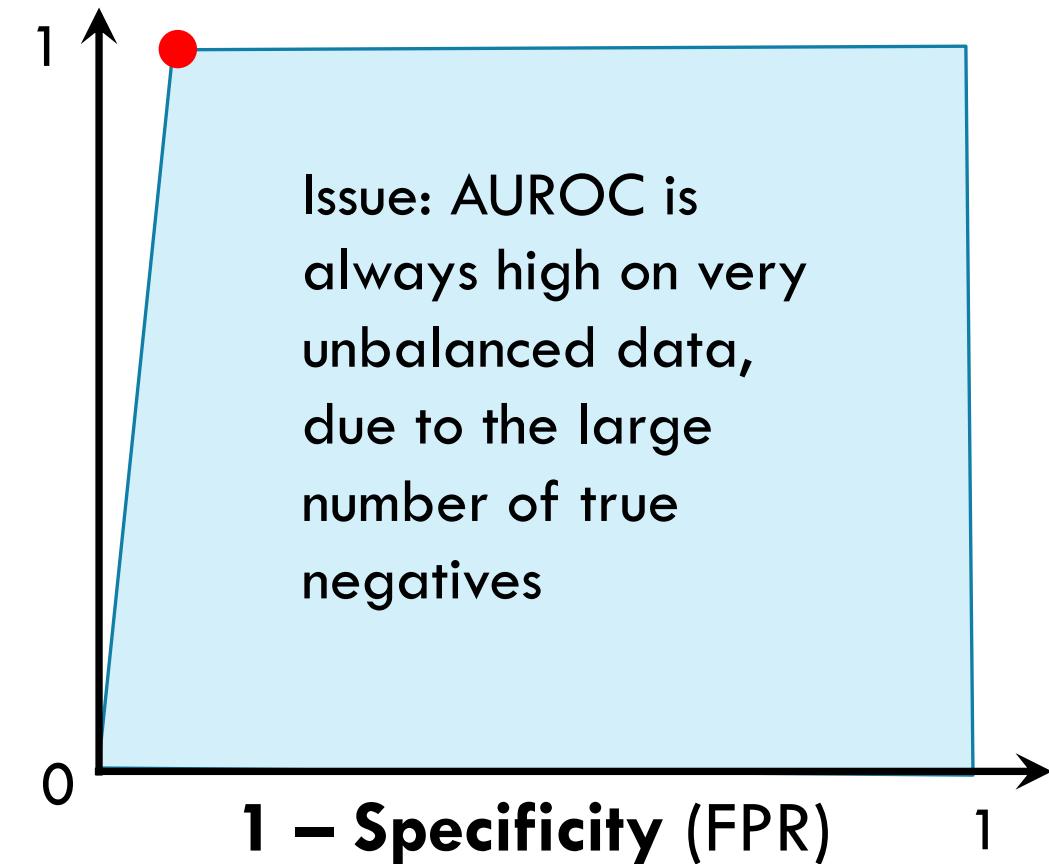
AUROC OF RANDOM CLASSIFIERS



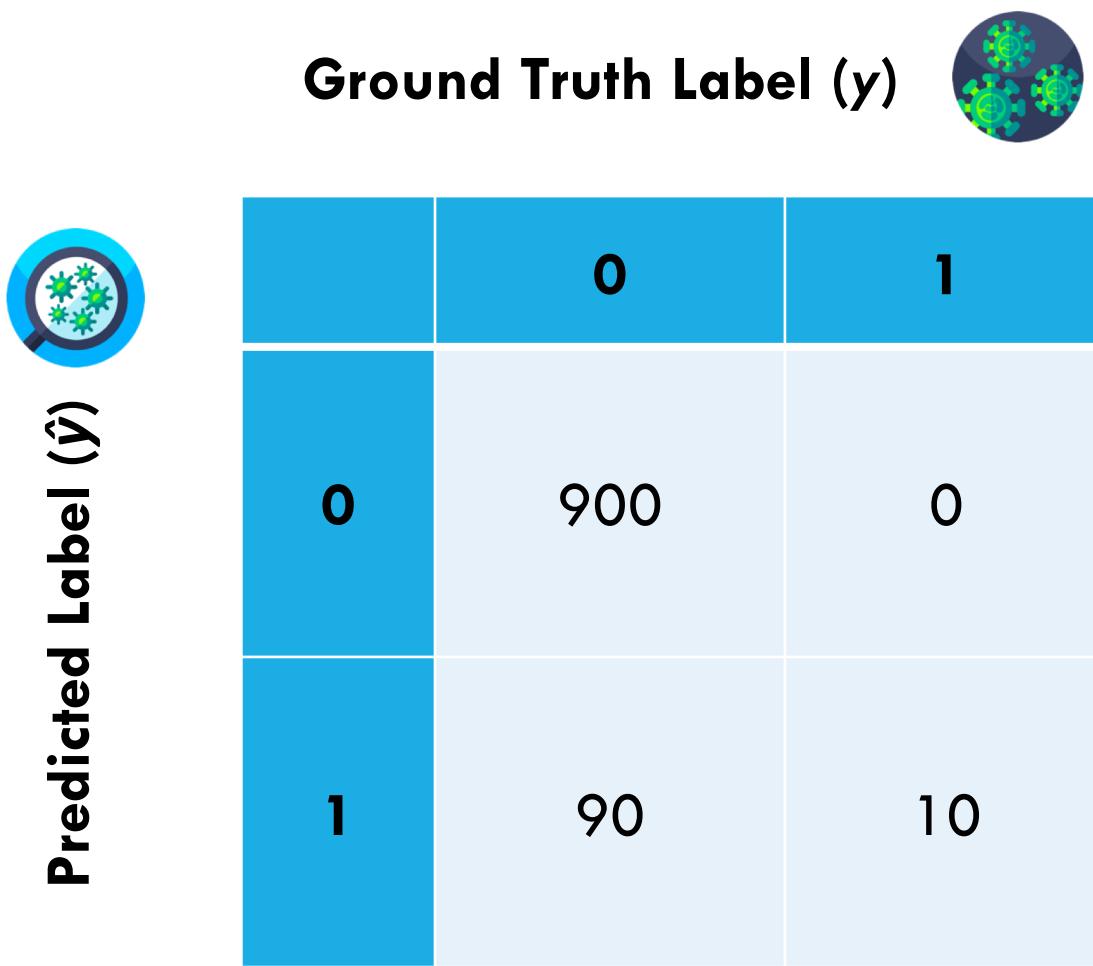
PROBLEM: AUROC ON HIGHLY UNBALANCED DATA



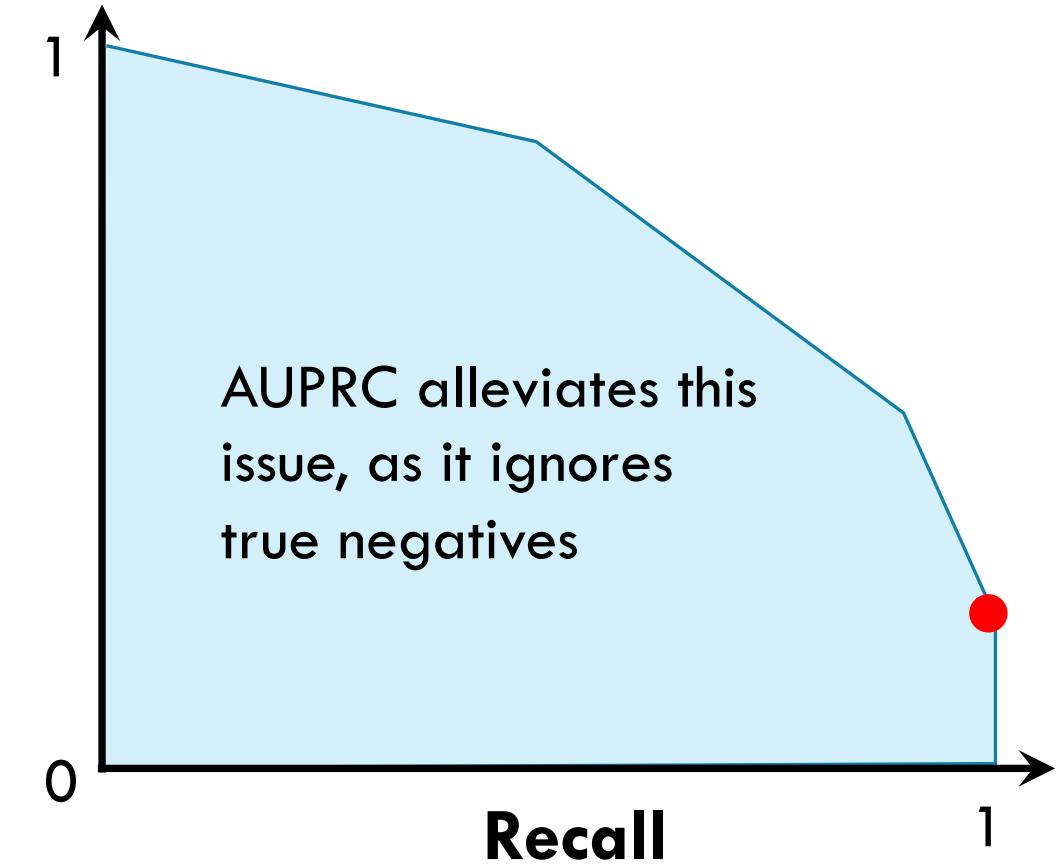
Sensitivity (TPR)



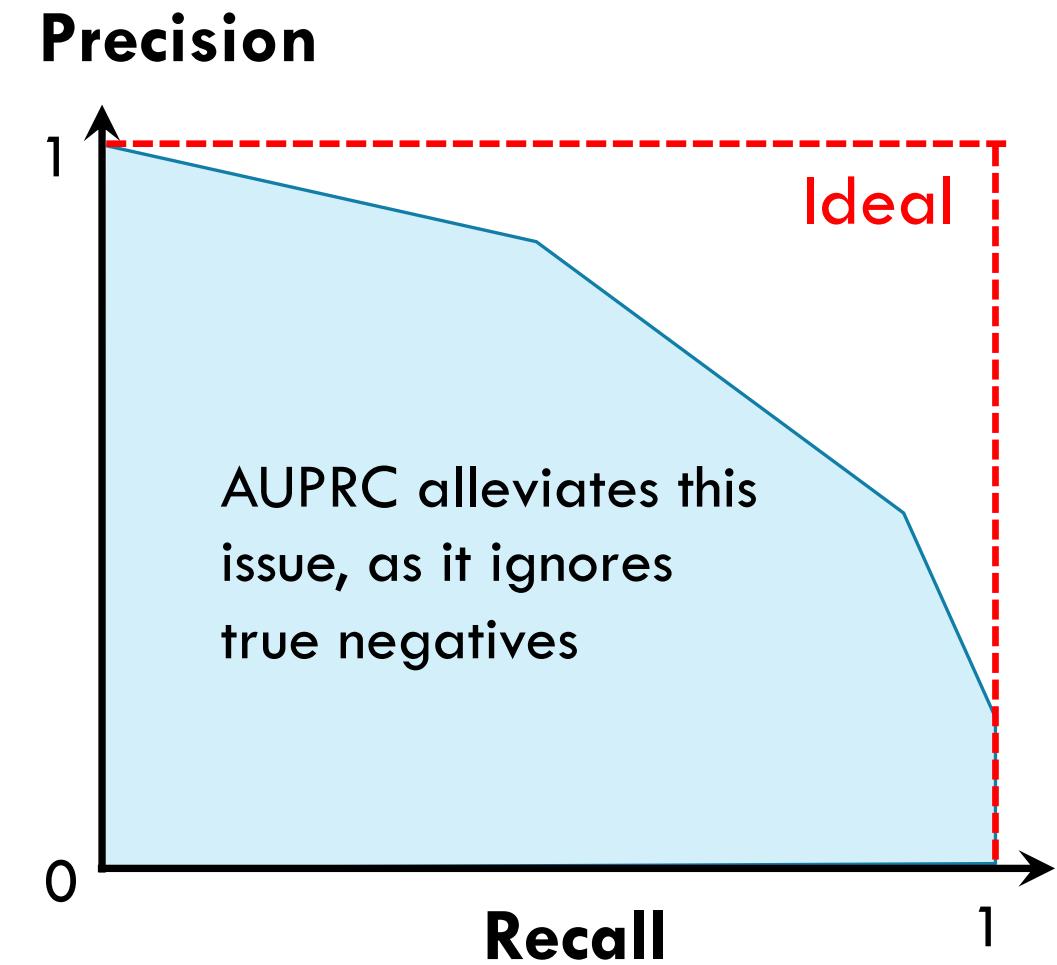
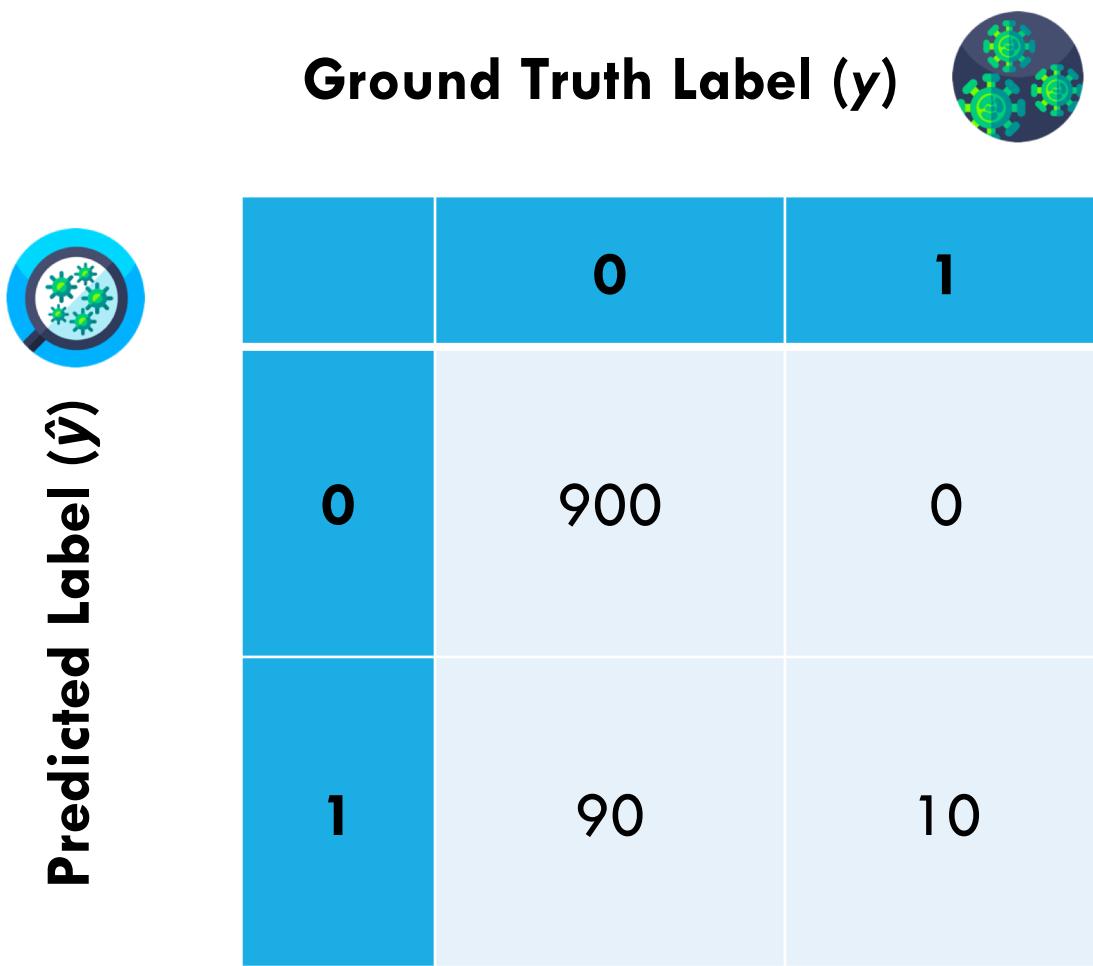
AREA UNDER PRECISION-RECALL CURVE (AUPRC)



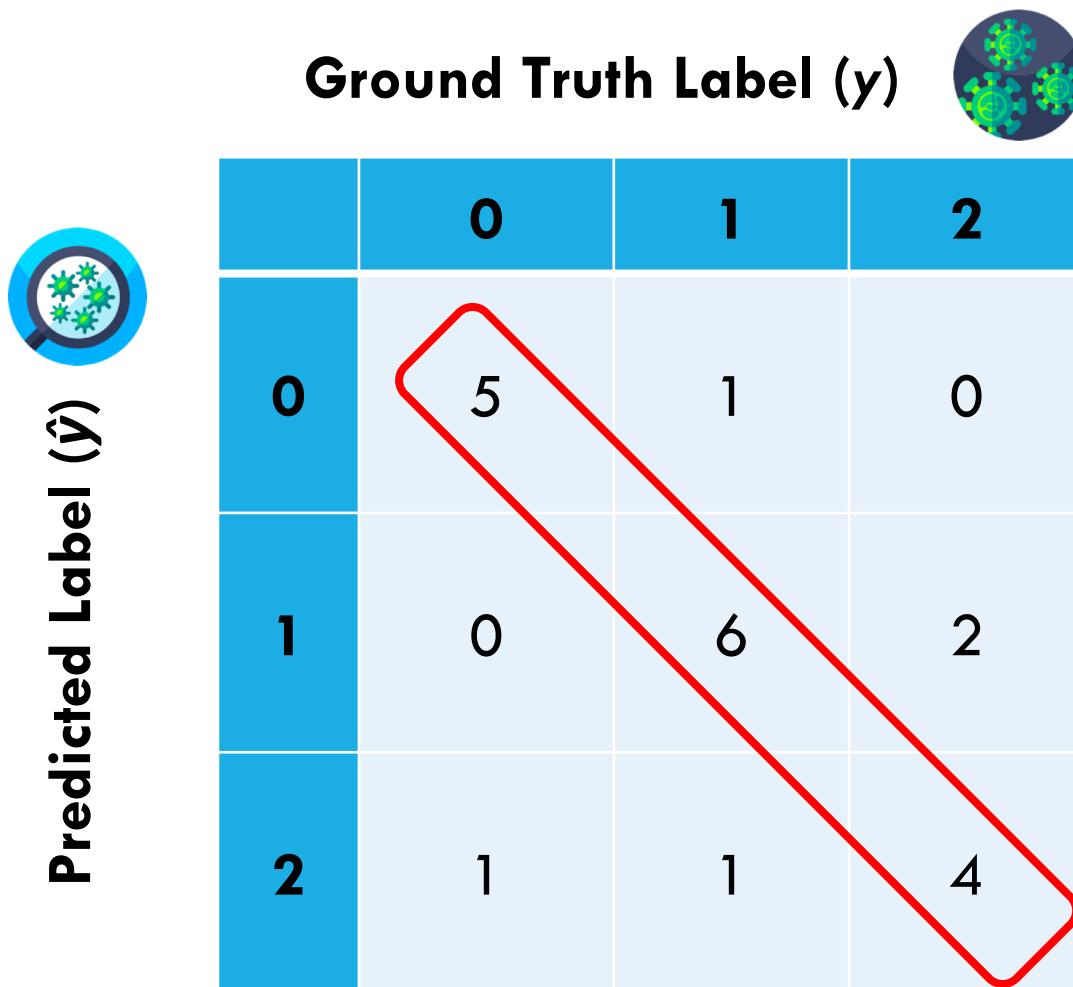
Precision



AREA UNDER PRECISION-RECALL CURVE (AUPRC)



MULTI-CLASS CLASSIFICATION EVALUATION

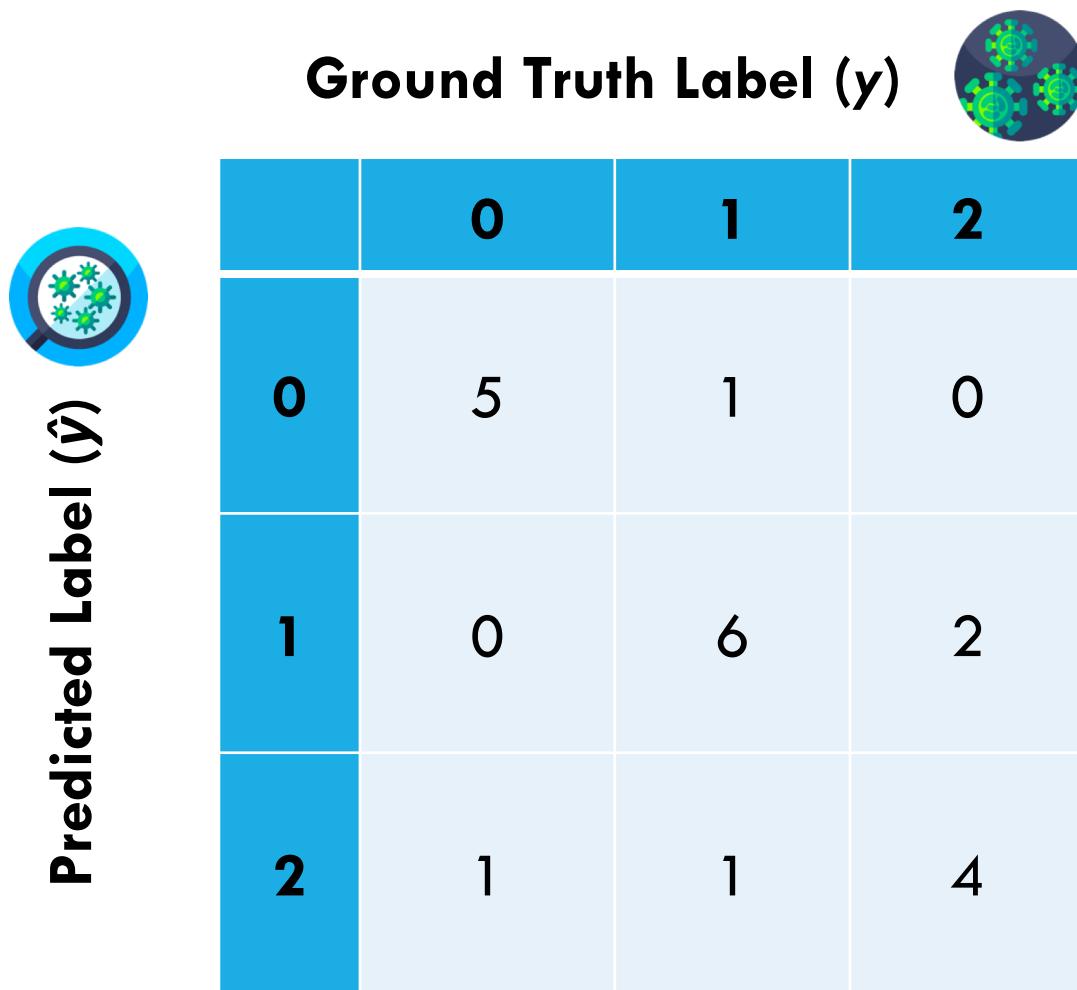


Accuracy

$$= (5 + 6 + 4) / 20$$

Total no. of samples

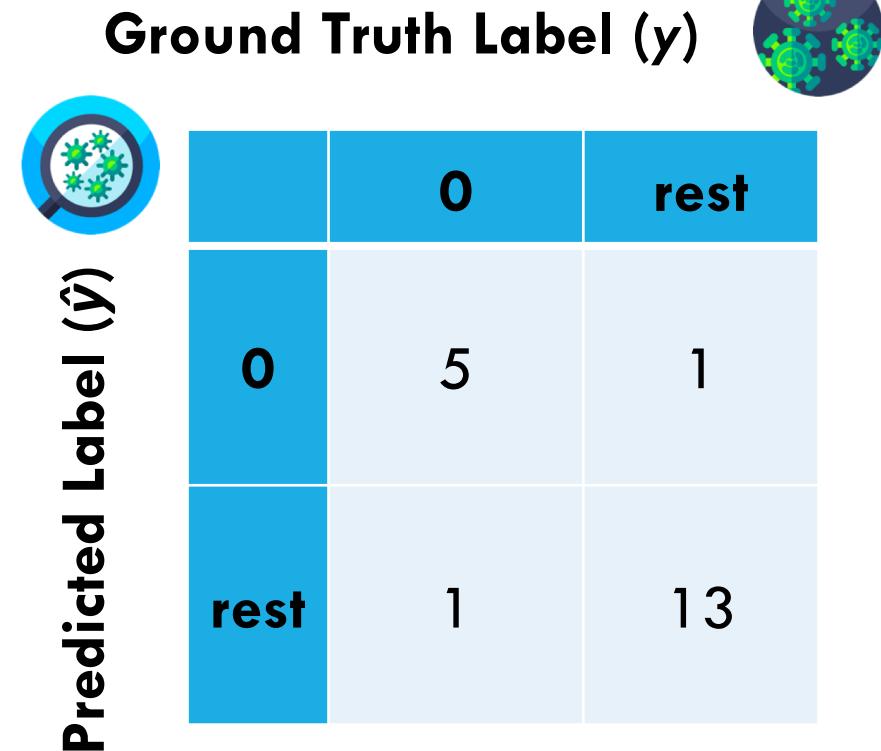
ONE VS REST CONFUSION MATRICES



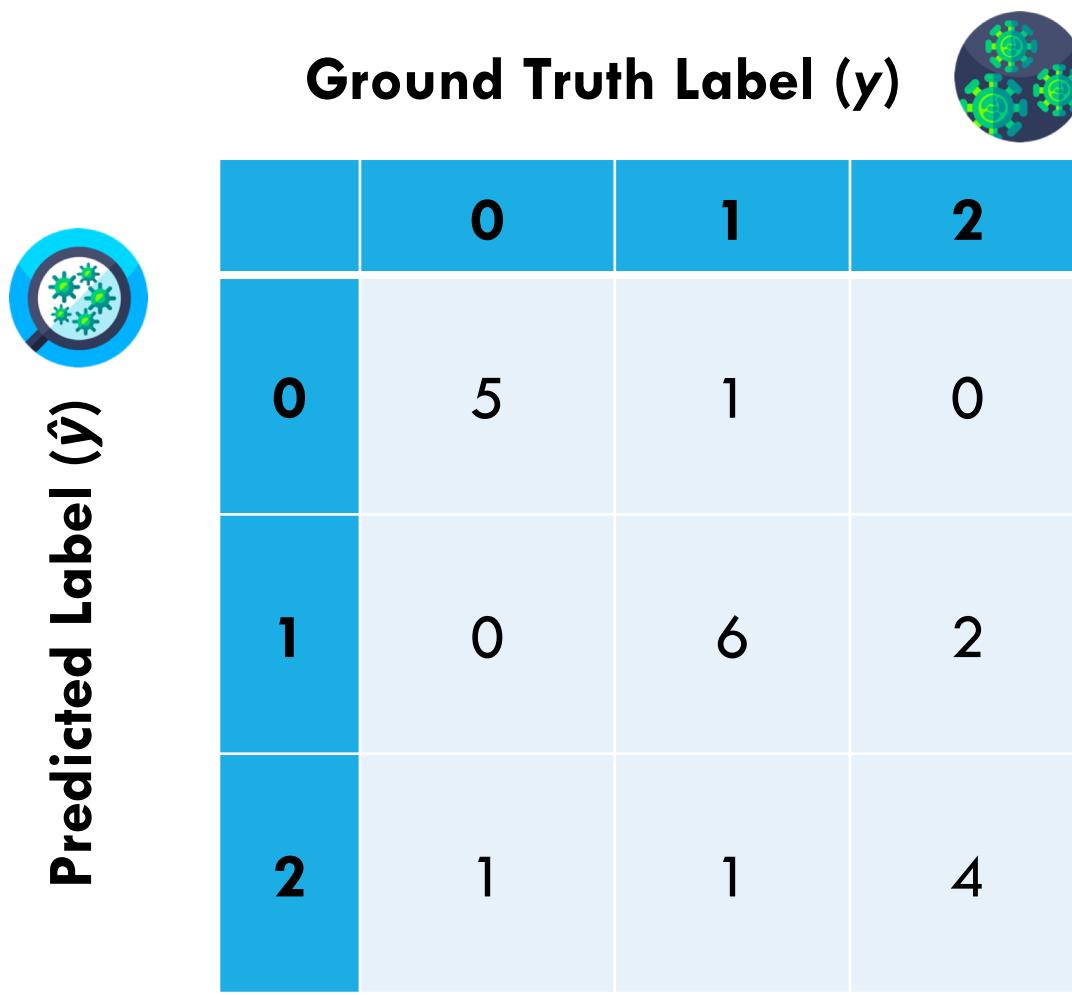
0 vs. rest



0 vs. Rest



ONE VS REST CONFUSION MATRICES



0 vs. rest



	0	rest
0	5	1
rest	1	13

1 vs. rest



	1	rest
1	6	2
rest	2	10

2 vs. rest



	2	rest
2	4	2
rest	2	12

MICRO PRECISION / RECALL / F1

		Ground Truth Label (y)		
		0	1	2
Predicted Label (\hat{y})	0	5	1	0
	1	0	6	2
	2	1	1	4

0 vs. rest



	0	rest
0	TP ₀	FP ₀
rest	FN ₀	TN ₀

Micro-averaging:
Average over the
confusion matrix values

1 vs. rest



	1	rest
1	TP ₁	FP ₁
rest	FN ₁	TN ₁

2 vs. rest



	2	rest
2	TP ₂	FP ₂
rest	FN ₂	TN ₂

MICRO PRECISION / RECALL / F1

		Ground Truth Label (y)		
		0	1	2
Predicted Label (\hat{y})	0	5	1	0
	1	0	6	2
	2	1	1	4

0 vs. rest

	0	rest
0	TP ₀	FP ₀
rest	FN ₀	TN ₀

1 vs. rest

	1	rest
1	TP ₁	FP ₁
rest	FN ₁	TN ₁

2 vs. rest

	2	rest
2	TP ₂	FP ₂
rest	FN ₂	TN ₂

Averaging

	-	-
-	TP _{ave}	FP _{ave}
-	FN _{ave}	TN _{ave}

Micro-averaging:
Average over the
confusion matrix values

MICRO PRECISION / RECALL / F1

		Ground Truth Label (y)		
		0	1	2
Predicted Label (\hat{y})	0	5	1	0
	1	0	6	2
	2	1	1	4

0 vs. rest
→

	0	rest
0	TP ₀	FP ₀
rest	FN ₀	TN ₀

1 vs. rest
→

	1	rest
1	TP ₁	FP ₁
rest	FN ₁	TN ₁

2 vs. rest
→

	2	rest
2	TP ₂	FP ₂
rest	FN ₂	TN ₂

Averaging

	-	-
-	TP _{ave}	FP _{ave}
-	FN _{ave}	TN _{ave}

Micro-averaging:
Average over the
confusion matrix values

→
Precision
Recall
F1 score

MACRO PRECISION / RECALL / F1

Macro-averaging:
Average over the
performance metric values

Ground Truth Label (y)				
Predicted Label (y)	0	1	2	
0	5	1	0	
1	0	6	2	
2	1	1	4	

0 vs. rest

1 vs. rest

2 vs. rest

	0	rest
0	TP ₀	FP ₀
rest	FN ₀	TN ₀

	1	rest
1	TP ₁	FP ₁
rest	FN ₁	TN ₁

	2	rest
2	TP ₂	FP ₂
rest	FN ₂	TN ₂

Precision₀
Recall₀

Precision₁
Recall₁

Precision₂
Recall₂

F1-Score₀

F1-Score₁

F1-Score₂

Averaging

F1-Score

MICRO VS MACRO METRICS: SUMMARY

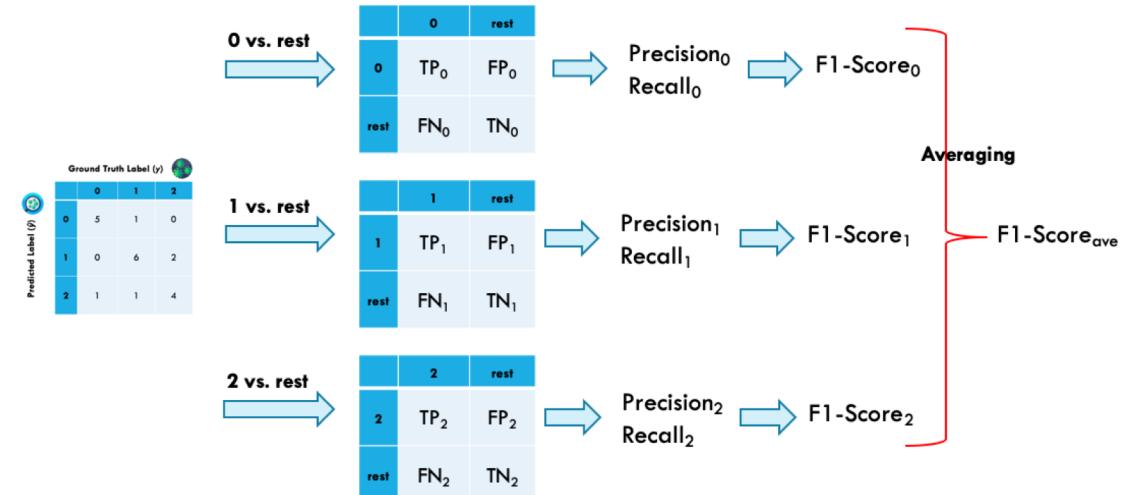
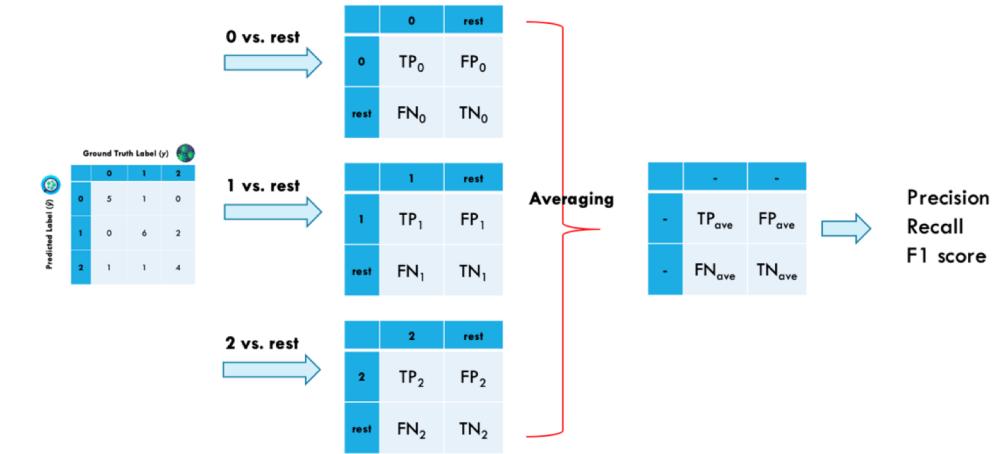
Both based on **one vs rest** confusion matrices

Micro: averaging takes place over *confusion matrix values* (i.e. TP, TN, FP, FN)

Macro: averaging takes place over *performance metric values* (i.e. Precision / Recall / F1)

Differences in behavior: macro-averaging treats all classes equally, while micro-averaging favors bigger classes

So, e.g. if we want to prioritize all classes equally, we should use macro-averaging



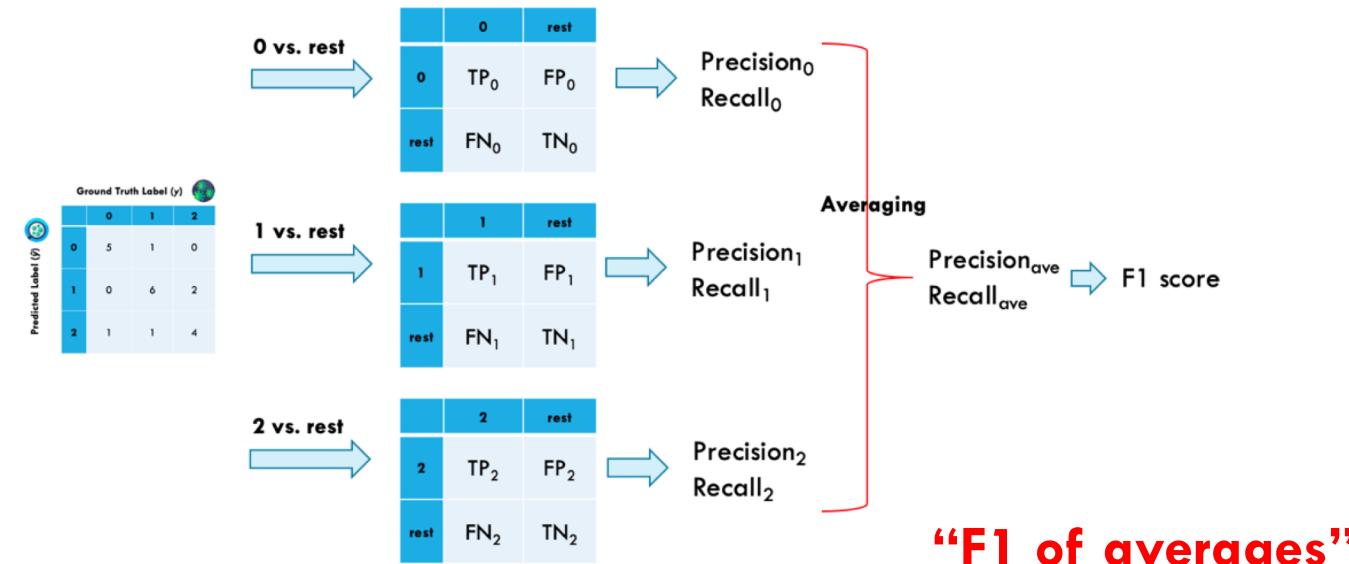
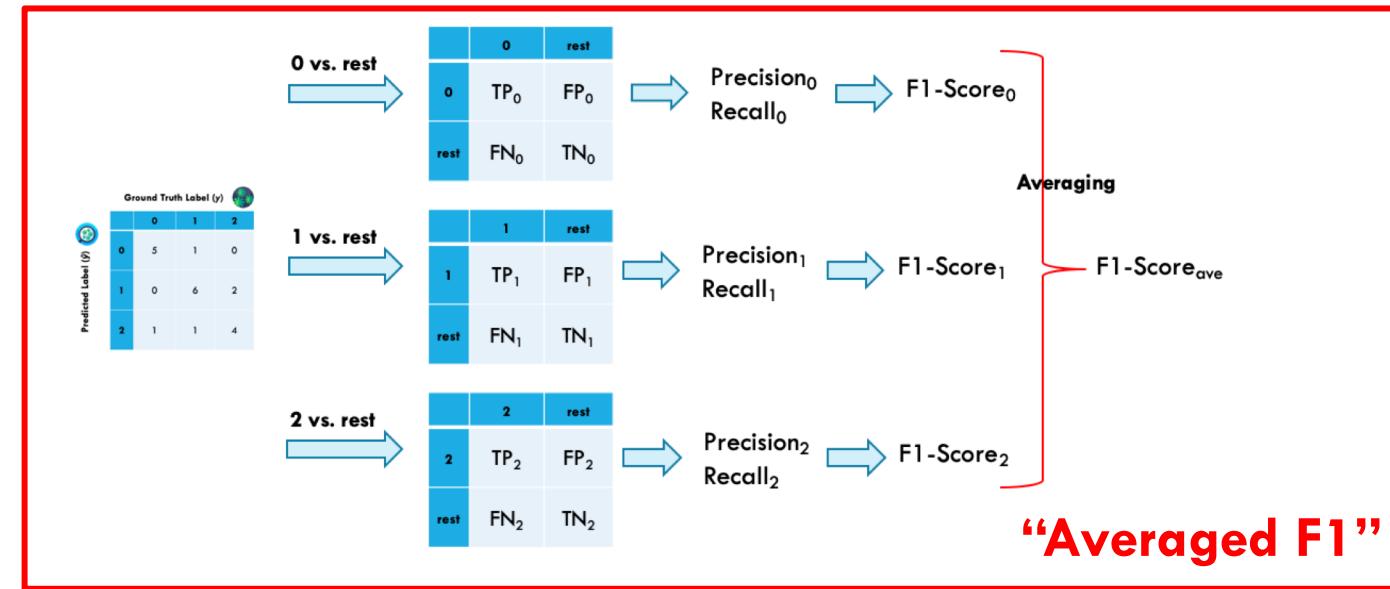
AMBIGUITY: TWO VERSIONS OF MACRO-F1

Optional

Averaged F1: Compute class specific F1 score; then average them to get macro-F1 score.

F1 of averages: Compute class-specific Precision / Recall, then average them to get Macro-Precision / Recall. Then macro-F1 is harmonic mean of Macro-Precision / Recall.

“History is written by the victors”:
sklearn.metrics.f1_score uses
Averaged F1



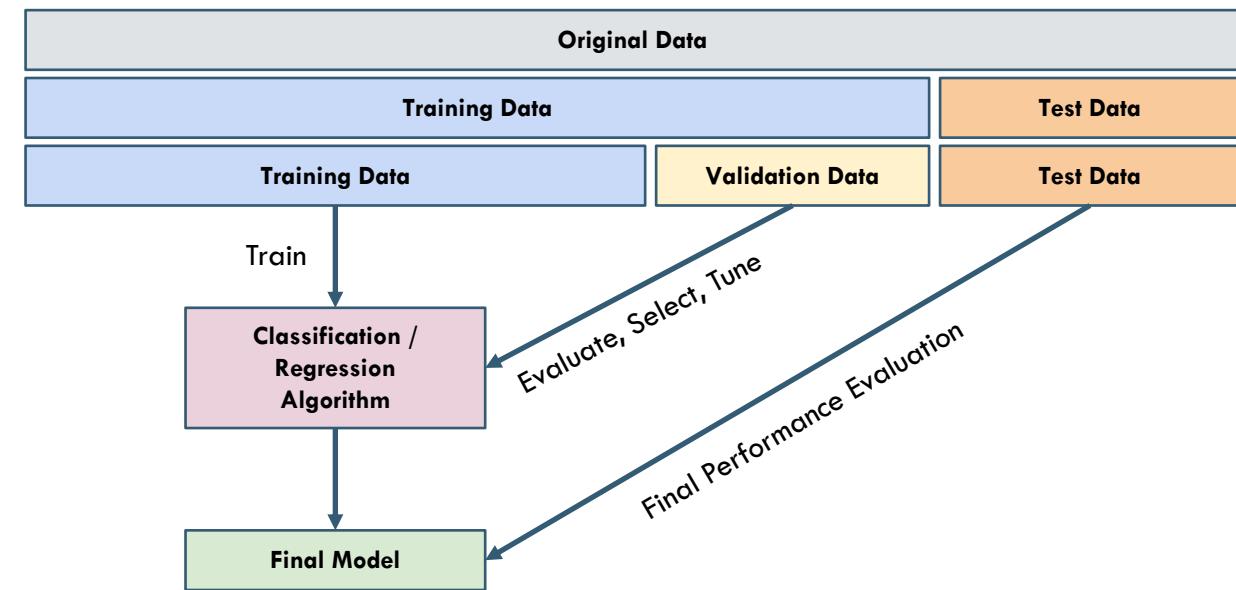
DATA SPLITTING: TRAIN, VALIDATION, TEST

Typical splitting procedure: split the data into train, validation, and test

Use test data at the end to measure performance of the final model

Use validation data for choosing which model to use or choosing between hyperparameters

(Proportions can vary, but a common choice is around 60% : 20% : 20%)

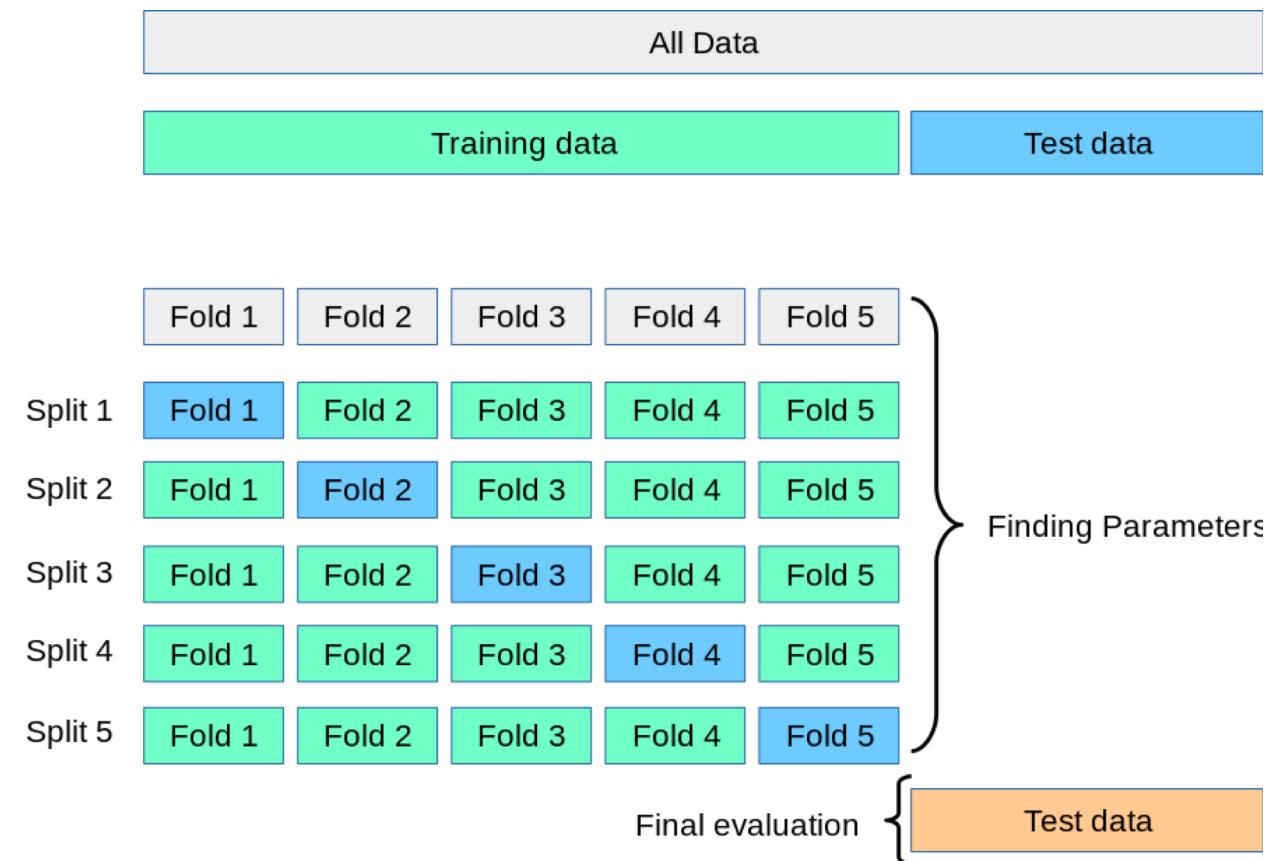


K-FOLD CROSS VALIDATION

Approach for hyperparameter / model selection: partition data into *folds*, and repeatedly train the data on **all folds except 1**, while evaluating the model on the **held-out (validation) fold**.

(Very similar to using a validation set, except that we average the results over the 5 choices of validation fold)

Finally use the test data at the end to measure performance



AVOIDING INFORMATION LEAKAGE

Important guideline:

- Do not normalize before splitting into training and test data
- Normalize training and test data but *only based on the statistics computed from training data*

```
# Split input data into training and test set
X_train, X_test = split(X, 0.2)

# Fit StandardScaler (i.e., calculate mean and variance)
scaler = preprocessing.StandardScaler().fit(X_train) # CORRECT!
#scaler = preprocessing.StandardScaler().fit(X)        # WRONG!!!

# Fit both training and test data
X_train_transformed = scaler.transform(X_train)
X_test_transformed = scaler.transform(X_test)
```

Intuition: test performance is only a valid estimate of an algorithm's performance if the algorithm's behavior *is not influenced by the test data*

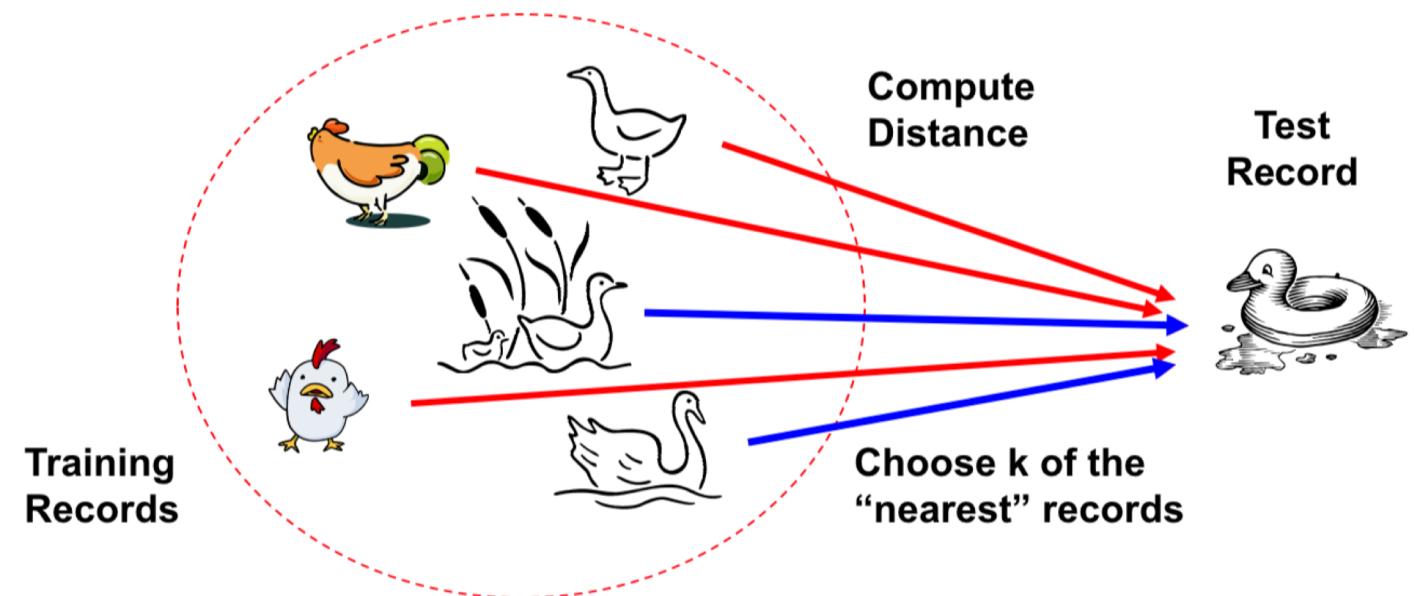
The preprocessor (the StandardScaler in the code above) is part of our algorithm, and is learning some parameters from the data (i.e. mean/variance of each feature). If we fit the preprocessor using X, this makes it influenced by the test data.

OVERVIEW

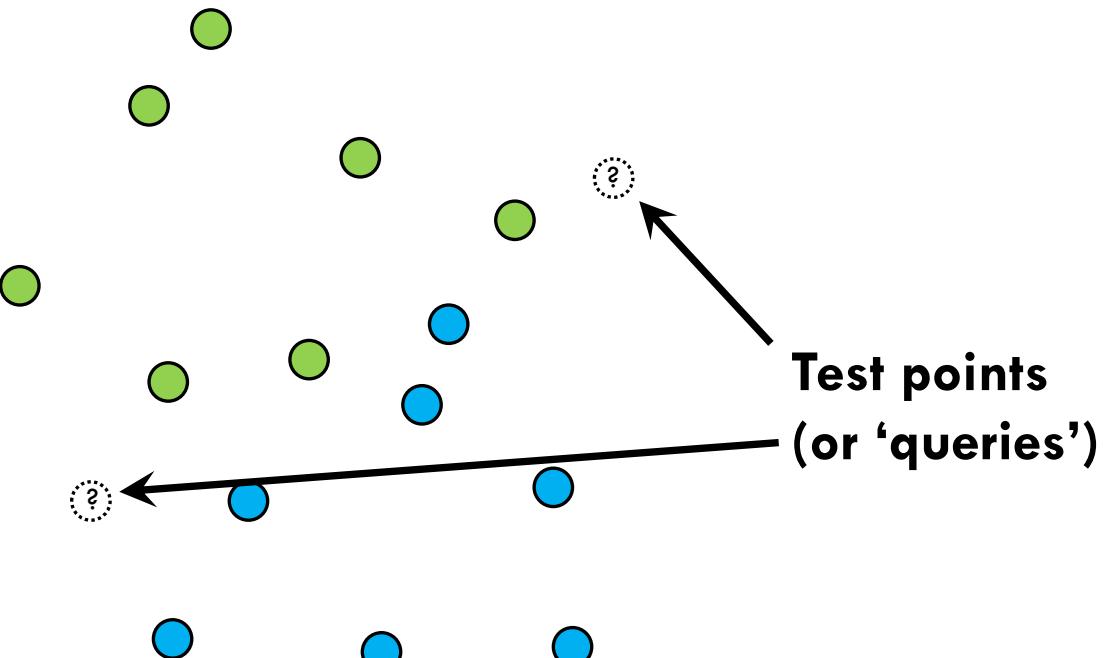
1. Problem Setup
2. Evaluating Classifiers
- 3. Nearest Neighbor Methods**
4. Tree-Based Methods
5. Ensemble Methods
6. Logistic Regression
7. Deep Learning

K NEAREST NEIGHBORS ALGORITHM

Principle: classify each test object (or ‘query’) based on its **nearest neighbors**, i.e. the training objects most similar to it

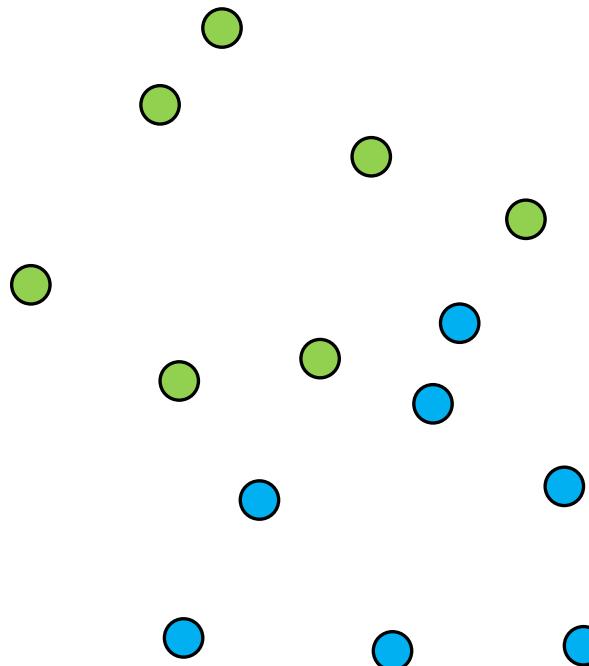


K NEAREST NEIGHBORS ALGORITHM



Training Data
(Blue = 0, Green = 1)

K NEAREST NEIGHBORS ALGORITHM

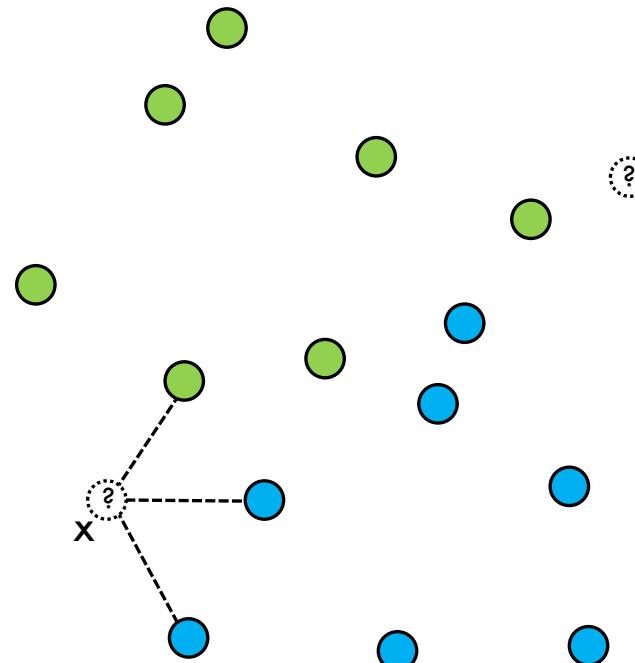


Training Data
(Blue = 0, Green = 1)

K NEAREST NEIGHBORS ALGORITHM

For each test point x :

- Compute its distances to all training instances: d_1, \dots, d_n
- **Majority Vote:** Output most frequent class among the k nearest neighbors of x

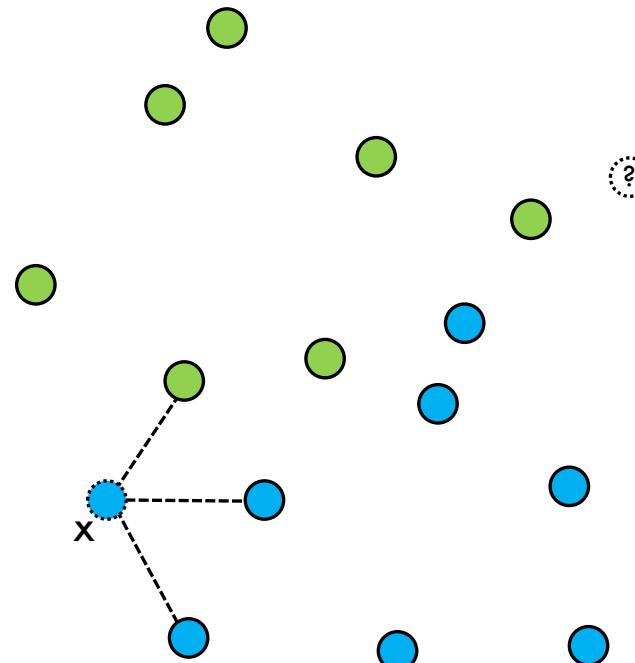


$$k = 3$$

K NEAREST NEIGHBORS ALGORITHM

For each test point x :

- Compute its distances to all training instances: d_1, \dots, d_n
- **Majority Vote:** Output most frequent class among the k nearest neighbors of x

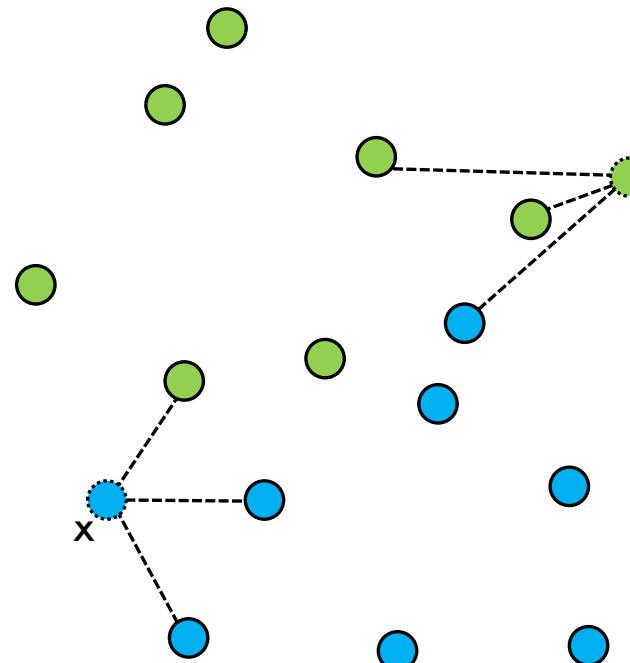


$$k = 3$$

K NEAREST NEIGHBORS ALGORITHM

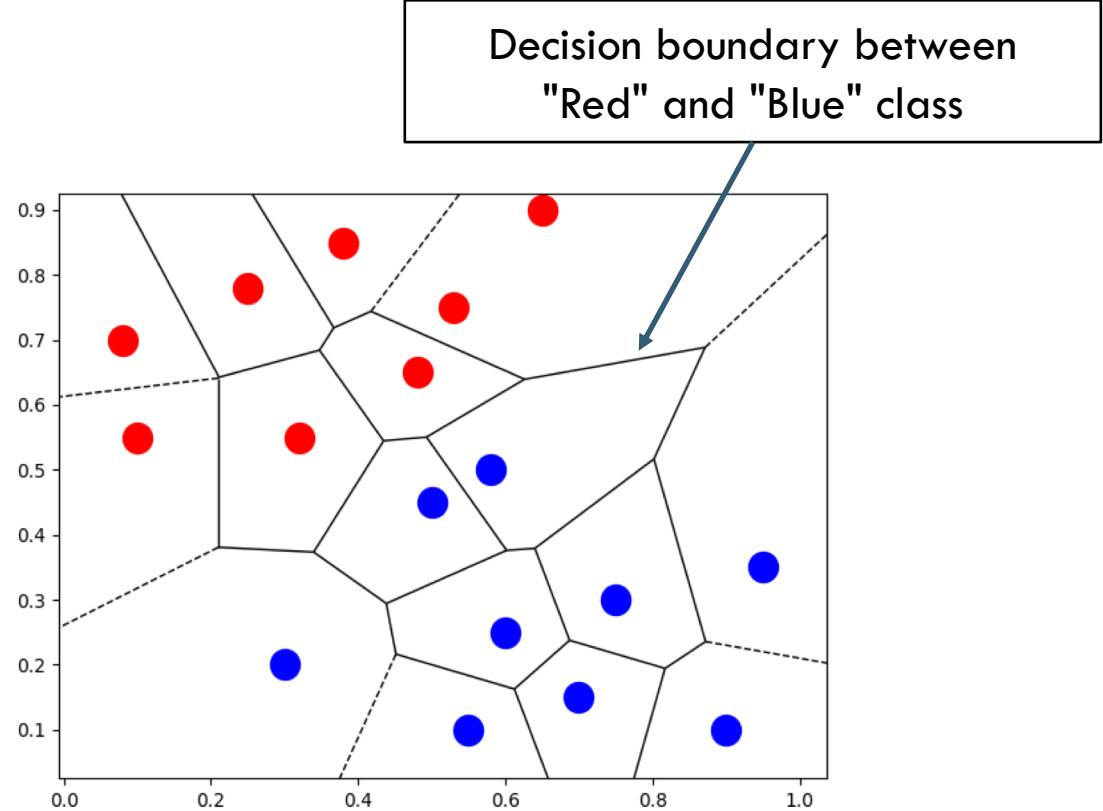
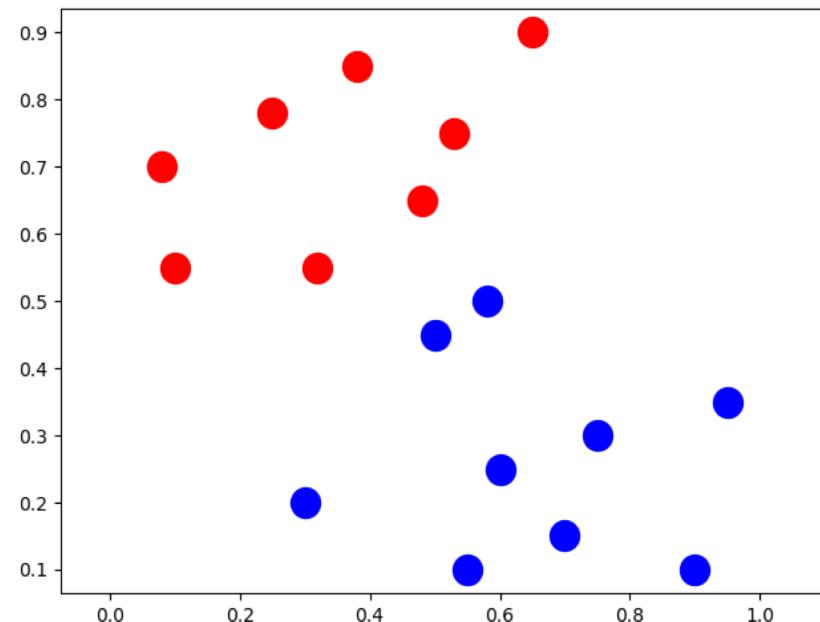
For each test point x :

- Compute its distances to all training instances: d_1, \dots, d_n
- **Majority Vote:** Output most frequent class among the k nearest neighbors of x



$$k = 3$$

1-NEAREST NEIGHBOR: VORONOI DIAGRAM

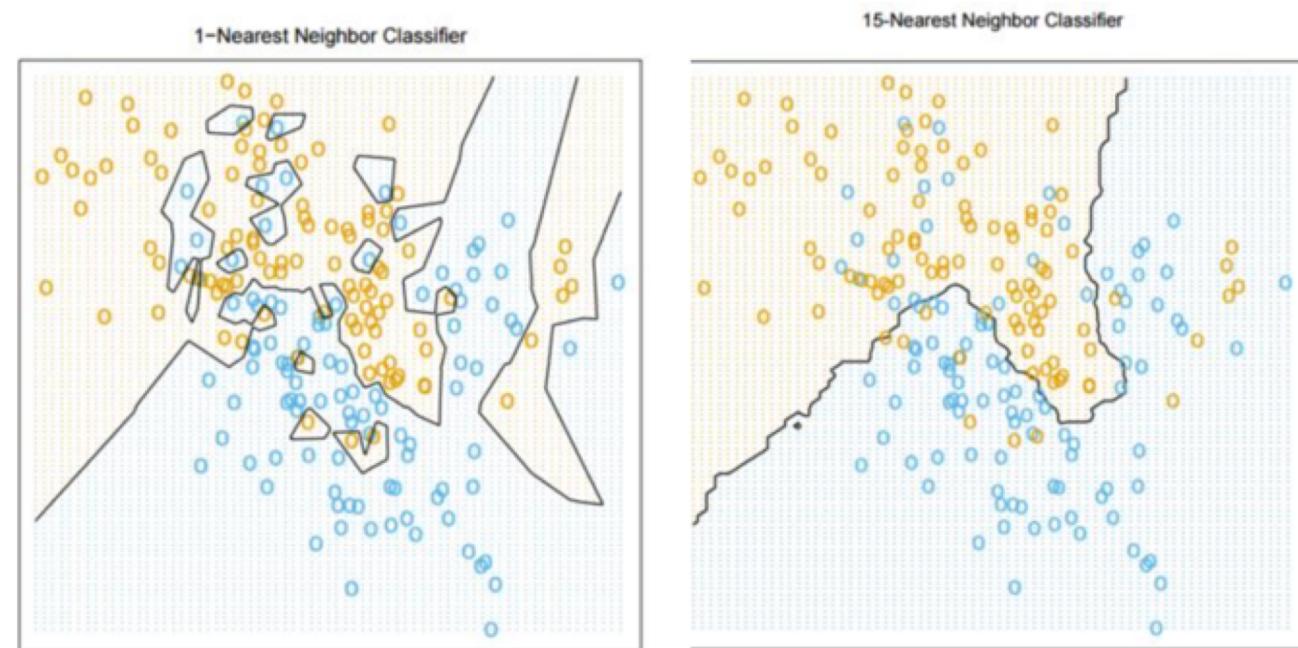


SELECTING K (UNDERFITTING VS OVERFITTING)

Selecting k is a trade-off:

- Smaller k: more flexible model, tends to overfit
- Larger k: less flexible model, tends to underfit

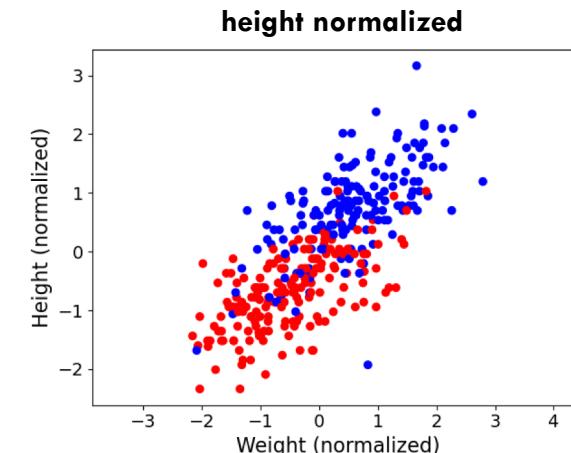
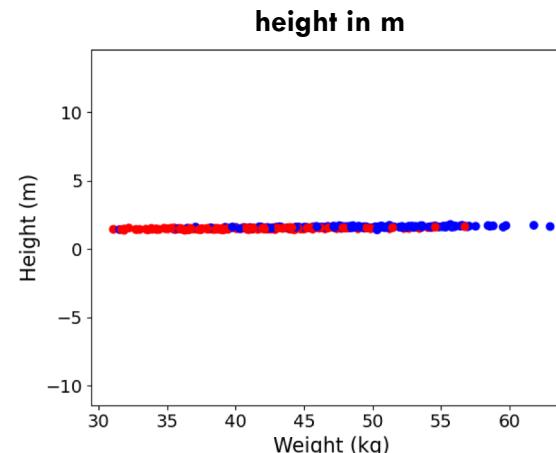
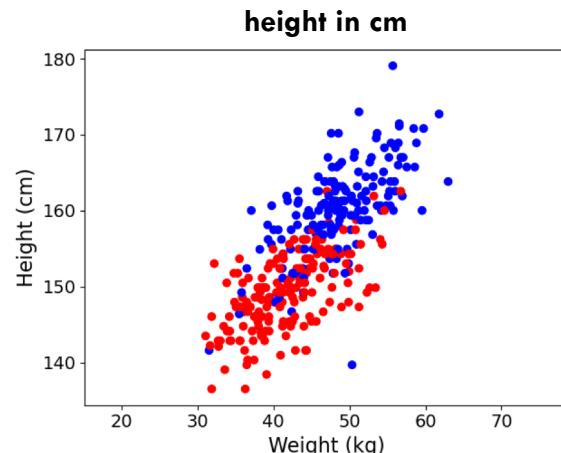
We can select k based on accuracy on a **validation set** or **cross-validation**



CAVEAT 1: SENSITIVE TO CHOICE OF SCALE

Scaling the variables (e.g. cm vs m) will change the distances, and thus the k-nearest neighbor results

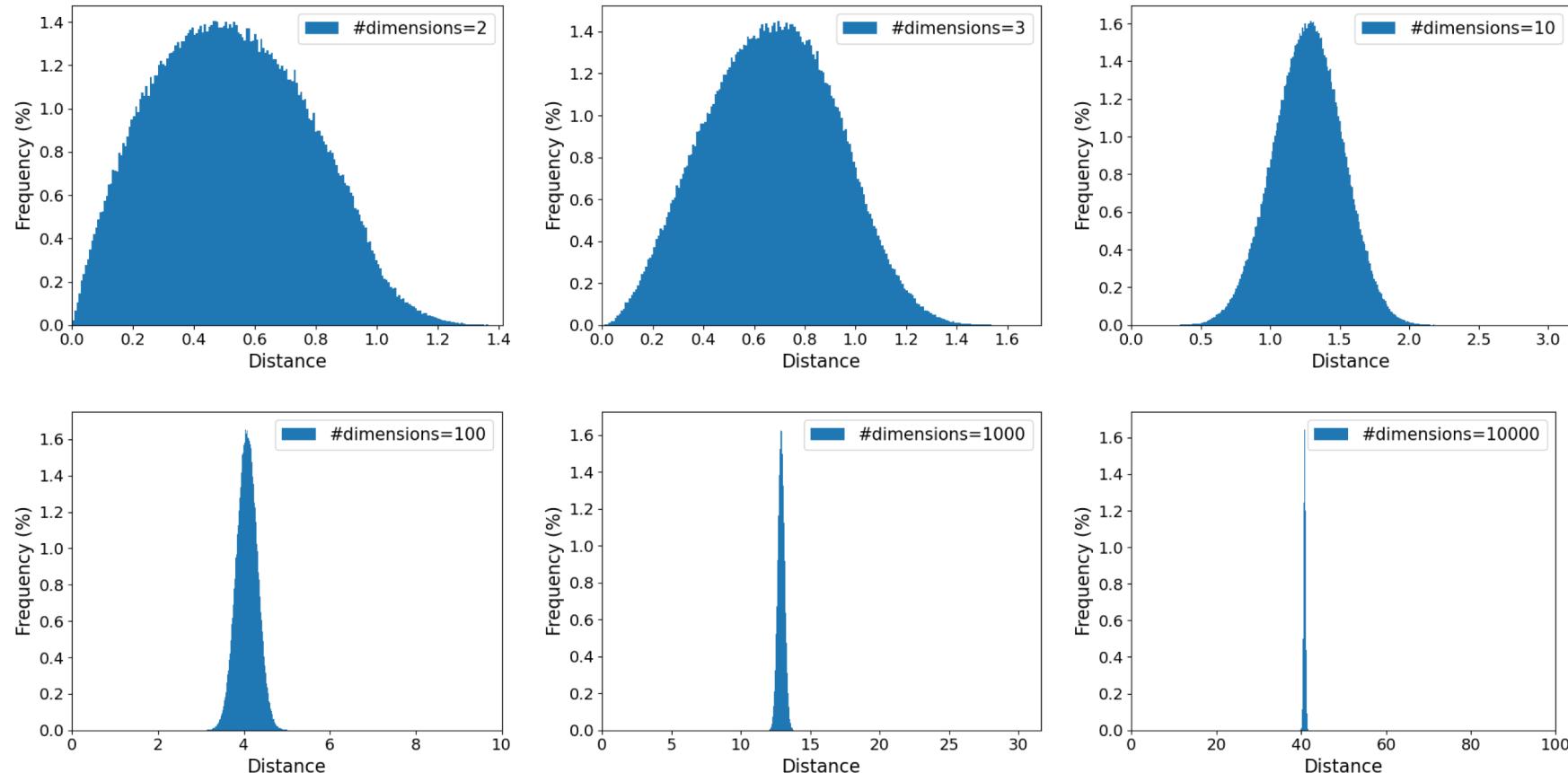
Example: measuring height in meters makes the variation in height basically negligible; normalization can fix this problem



CAVEAT 2: CURSE OF DIMENSIONALITY

Intuition: As dimension size increases, the distances between most pairs of points tend to become *large* and *increasingly similar*

Hence, both distances and nearest neighbors tend to become *uninformative*.

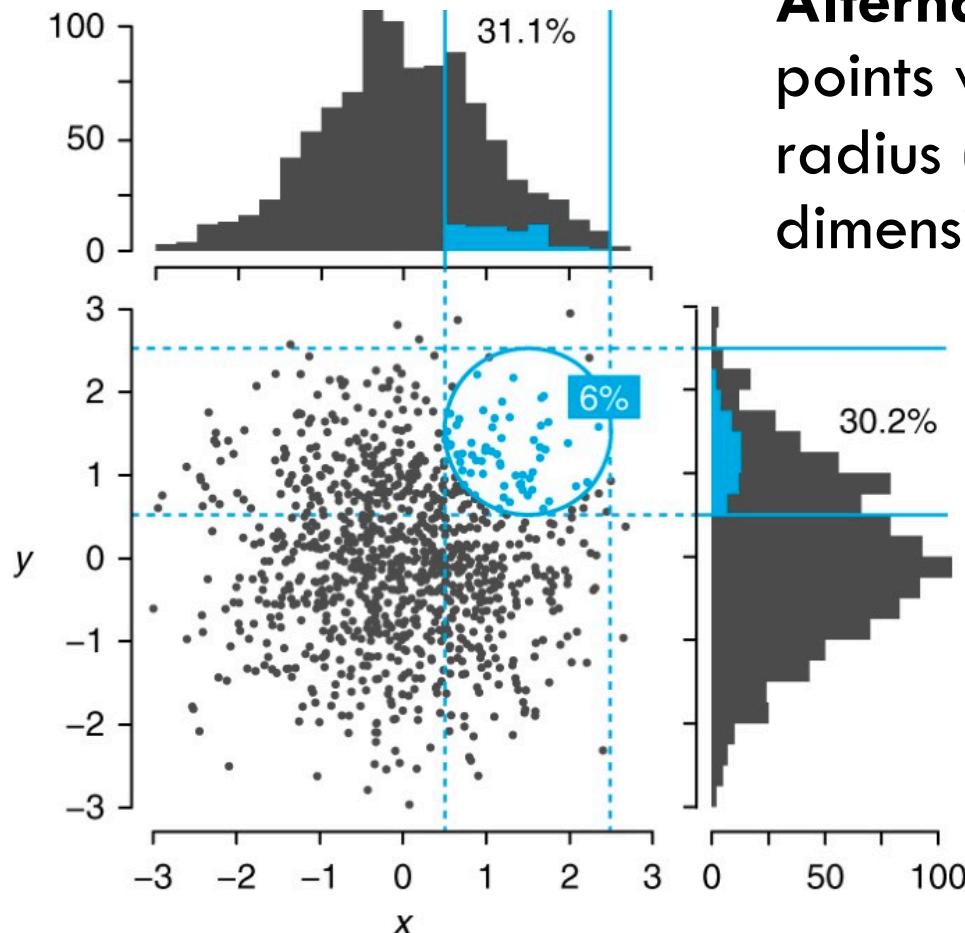


CURSE OF DIMENSIONALITY

Intuition: As dimension size increases, the distances between most pairs of points tend to become *large* and *increasingly similar*

Hence, both distances and nearest neighbors tend to become uninformative.

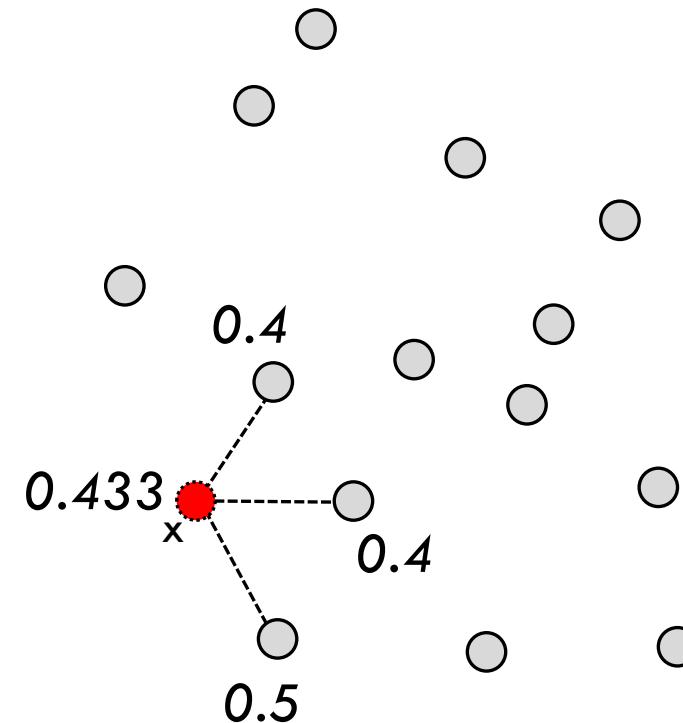
Alternate view: number of points within a ball of fixed radius (e.g. 1) decays as the dimensionality increases



K NEAREST NEIGHBORS REGRESSION

For each test point x :

- Compute its distances to all training instances: d_1, \dots, d_n
- **Aggregate:** Output the average (e.g. mean, median) of the response variable among the k -nearest neighbors of x

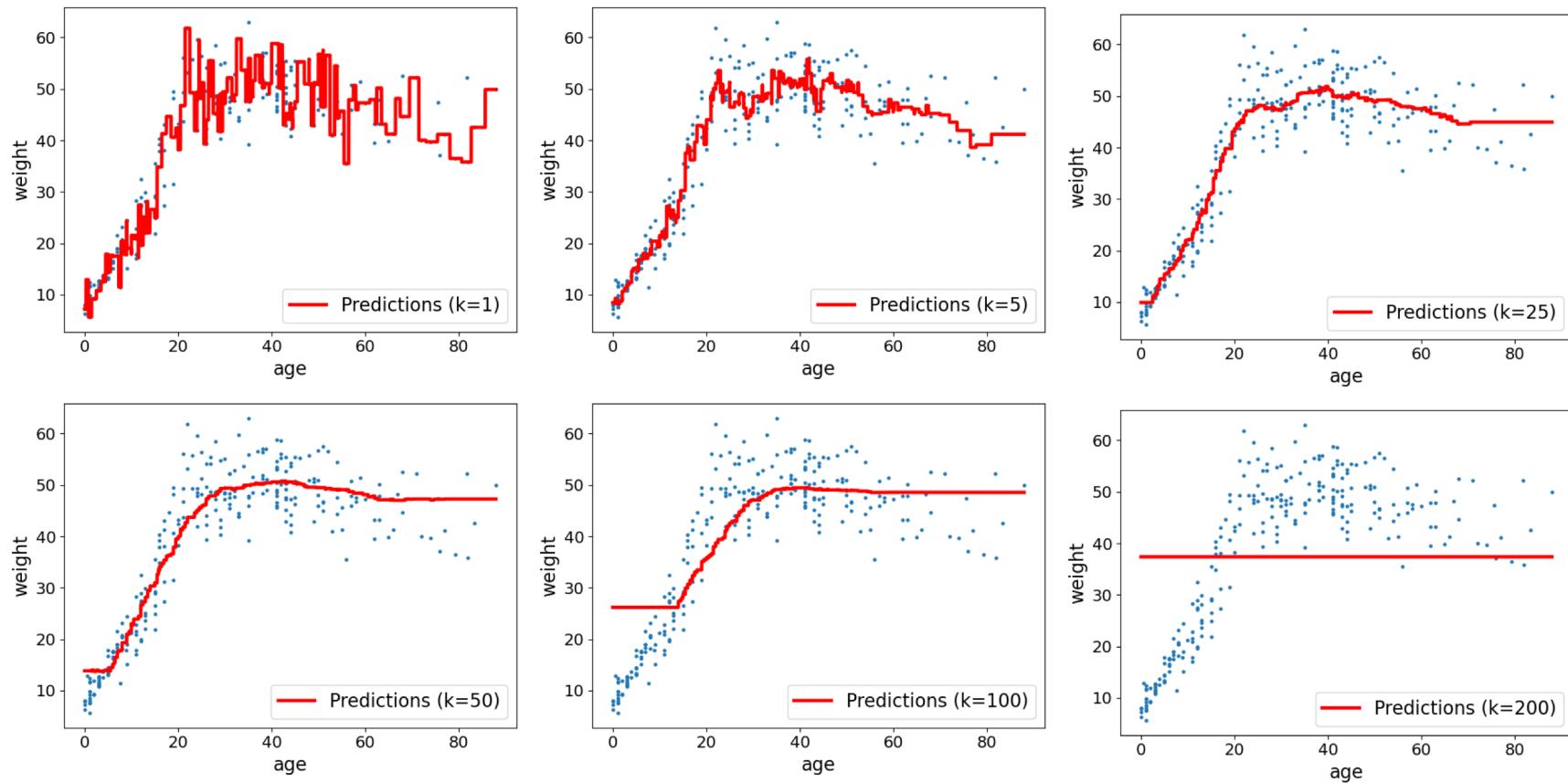
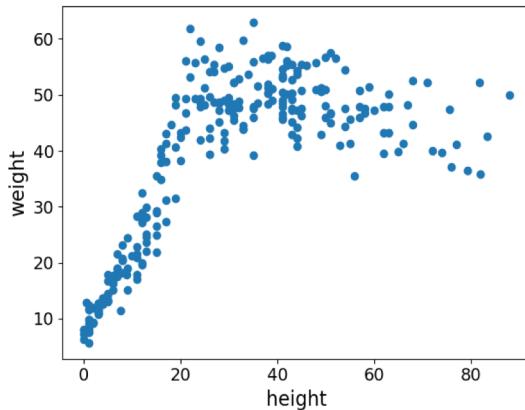


$k = 3$

KNN FOR REGRESSION: EXAMPLE

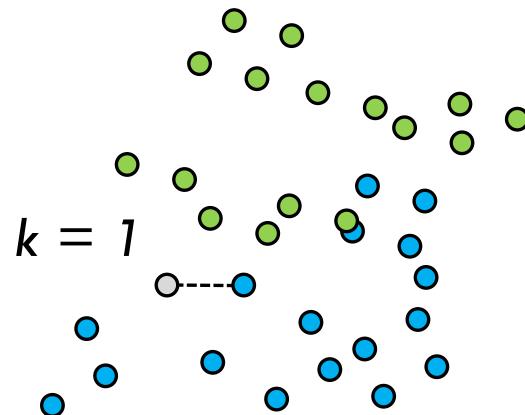
Example use-case:
predict weight from
height

Method: k nearest
neighbours for
 $k = 1, 5, 25, 50,$
 $100, 200$



THEORETICAL GUARANTEES

(Cover, Hart) As $n \rightarrow \infty$, the error of 1-nearest neighbours is at most 2 times the error of the optimal¹ classifier.



1: The optimal (“Bayes optimal”) classifier is the classifier that (unrealistically) predicts based on perfect knowledge of the **true distribution** $P(y | x)$. Specifically, the Bayes optimal classifier is defined as $y^* = \operatorname{argmax}_y P(y | x)$. In practice, the true distribution cannot be known, so such optimal classifiers are not used in practice; they are just seen as an upper bound for any classifier’s performance.

SURPRISINGLY COMPETITIVE PERFORMANCE

Recommendation Systems

Are We Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches

Maurizio Ferrari Dacrema
Politecnico di Milano, Italy
maurizio.ferrari@polimi.it

Paolo Cremonesi
Politecnico di Milano, Italy
paolo.cremonesi@polimi.it

Dietmar Jannach
University of Klagenfurt, Austria
dietmar.jannach@aau.at

“Only 7 of them could be reproduced ... 6 of them can often be outperformed with comparably simply heuristic methods, e.g. based on nearest neighbor or graph-based techniques.”

(Dacrema et al., 2019)

Time Series Classification

“...multiple rigorous independent studies show that for the core problem of time series classification, Nearest Neighbor DTW is very hard to beat.”

(Mueen & Keogh, 2016, “Extracting Optimal Performance from Dynamic Time Warping”)

PROS AND CONS

Pros:

- Very simple & interpretable algorithm
- Generic: can run as long as a distance exists
- No training time
- Can produce arbitrarily shaped decision boundaries

Cons:

- Finding neighbors at test time can be slow: $O(m \times n \times d)$ time
 - (Efficiency can often be improved using fast nearest neighbor search, e.g. k-d trees for lower dimensionality data (e.g. up to 20 or so), or approximate similarity search for higher dimensionality)
- Curse of dimensionality (in high dimensions)
- Need to store & query all the training data (may be high memory usage)

test samples
train samples
dimensions

EXAMPLE: DIALECTS OVER THE U.S.

