MMA/MMAI 869
Machine Learning and AI

Performance Metrics

Stephen Thomas

Updated: Oct 29, 2022

Outline

- How do we quantify the quality of a model's predictions?

truth	predicted
Yes	Yes
Yes	No
No	No
No	No
Yes	Yes
No	Yes
Yes	No
.....	...

Accuracy: 0.300
Precision: 0.800
Recall: 0.667
F1: 0.727
Specificity: 0.750
…

Reminder: Machine Learning Terminology

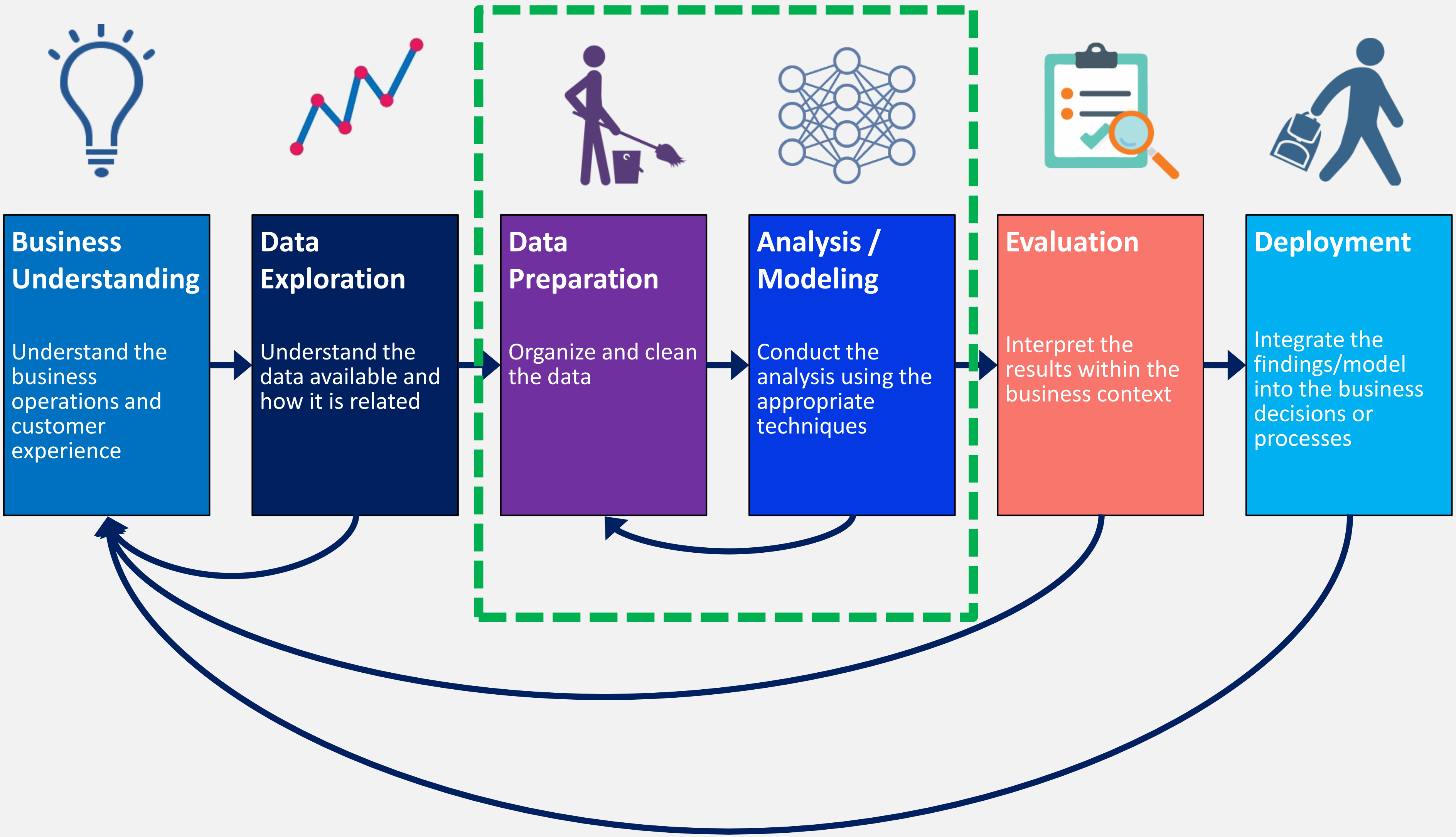
Features
(inputs, independent variables, X, columns)

Target or **Label**
(response, output, dependent variable, Y)

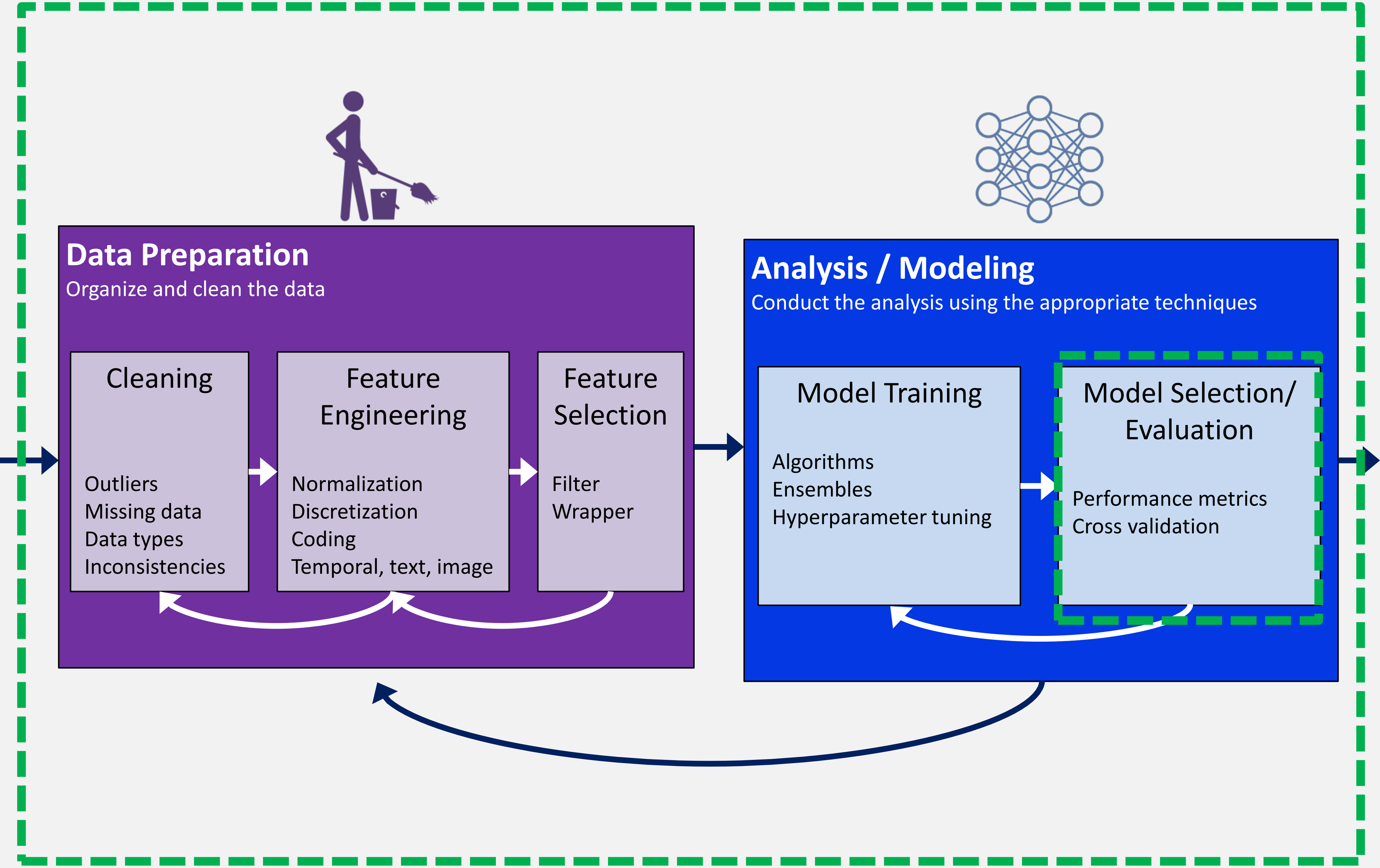
Instances
(rows, cases, records)

Age	Income	Married	Citizenship	Default
55	36,765	True	Canada	True
66	87,983	True	Canada	True
21	24,354	False	USA	False
24	56,654	True	Canada	False
34	98,324	False	UK	False
36	132,229	False	Germany	True
28	35,000	True	Canada	False
49	50,334	True	Canada	False

The Analytics Process: CRISP-DM



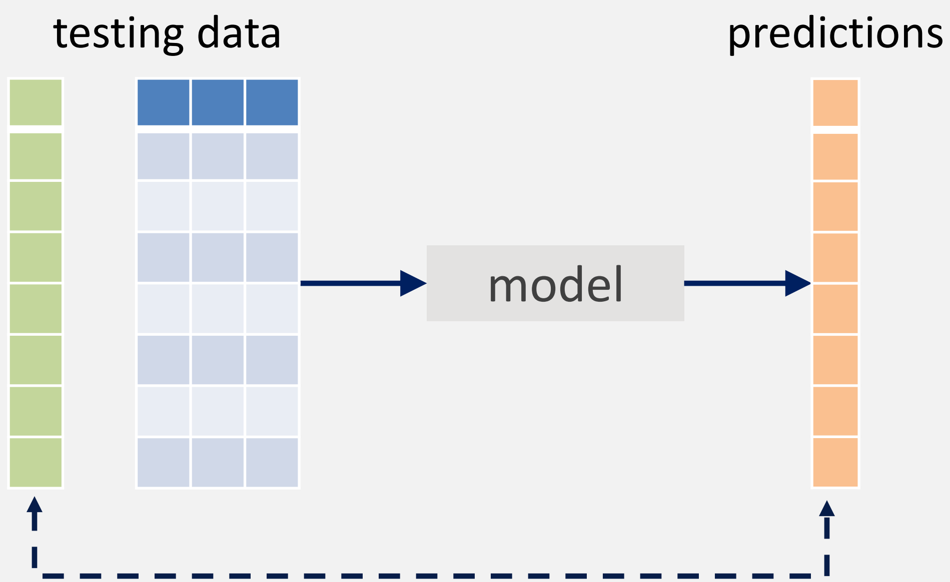
More Detail


Data Preparation
Organize and clean the data
Cleaning
Outliers
Missing data
Data types
Inconsistencies
Feature Engineering
Normalization
Discretization
Coding
Temporal, text, image
Feature Selection
Filter
Wrapper
Analysis / Modeling
Conduct the analysis using the appropriate techniques
Model Training
Algorithms
Ensembles
Hyperparameter tuning
Model Selection/ Evaluation
Performance metrics
Cross validation

METRICS FOR PERFORMANCE EVALUATION

Performance Metrics

- How good are a model's predictions?



Compare using metric:

Regression
- Mean Squared Error
- Mean Absolute Error
- Root MSE

Classification
- Accuracy/Error
- Precision, Recall
- F1 score
- Sensitivity, Specificity
- ROC Curve and AUC
- Log Loss

Recommendation
- Mean Average Precision @ K
- Coverage
- Personalization
- Intra-list similarity

Classification Report

- Scikit-learn has a function *classification_report()*:

	precision	recall	f1-score	support
0	0.97	0.94	0.96	80
1	0.78	0.90	0.84	20
accuracy			0.93	100
macro avg	0.88	0.92	0.90	100
weighted avg	0.94	0.93	0.93	100

- What do all these numbers mean? Which is most important?

Multi-class Metrics

- Basically the same, just more of them

	precision	recall	f1-score	support
class 0	0.50	1.00	0.67	1
class 1	0.00	0.00	0.00	1
class 2	1.00	0.67	0.80	3
accuracy			0.60	5
macro avg	0.50	0.56	0.49	5
weighted avg	0.70	0.60	0.61	5

Classification Metrics

A model's predictions might be right or wrong, in two different ways each.

Truth Target	Predicted Target	
Yes	Yes	TP (true positive)
Yes	No	FN (false negative) Type II error
No	No	TN (true negative)
No	Yes	FP (false positive) Type I error
Yes	Yes	…
Yes	No	
No	No	

compare

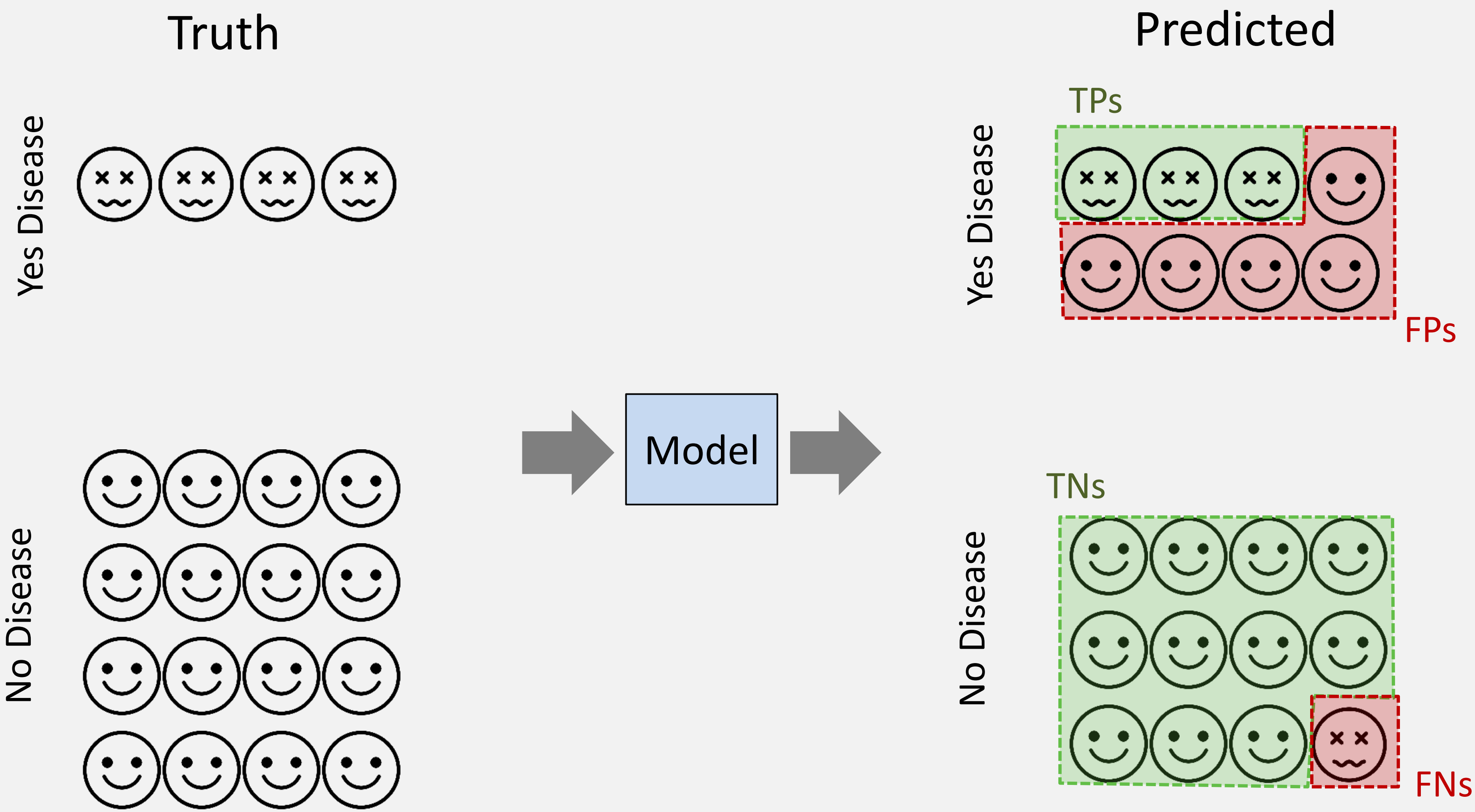

CONGRATS YOU'RE PREGNANT
YOU'RE NOT PREGNANT
TYPE I ERROR
TYPE II ERROR

Exercise

Mark each of the following as TP, TN, FP, or FN

Truth Target	**Predicted Target**
Yes	Yes
No	Yes
Yes	No
No	No
Yes	Yes
Yes	No
No	No

A Medical Example



Confusion Matrix

- ***Confusion matrix***: a tabulation of the predictions against the truth

Truth Target	Predicted Target	
Yes	Yes	TP
No	Yes	FP
Yes	No	FN
No	No	TN
Yes	Yes	TP
Yes	No	FN
No	No	TN

		Predicted Yes	Predicted No	
Truth	Yes	2	2	4
	No	1	2	3
		3	4	7

2 TP (true positive)
2 FN (false negative)
1 FP (false positive)
2 TN (true negative)

Accuracy

- ***Accuracy:*** percentage of instances that are classified correctly
- *1-Accuracy = Error*

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

		Predicted: Yes	Predicted: No	
Truth	Yes	15	5	20
	No	5	75	80
		20	80	100

$\text{Accuracy} = ?$

		Predicted: Yes	Predicted: No	
Truth	Yes	10	10	20
	No	30	50	80
		40	60	100

$\text{Accuracy} = ?$

Accuracy is Usually a Bad Choice

- Most datasets are imbalanced
- Which model is better?

Model 1

Predicted

		Yes	No	
Truth	Yes	0	10	10
	No	0	990	990
		0	1000	1000

$$\text{Accuracy} = \frac{990}{1000} = .99$$

Model 2

Predicted

		Yes	No	
Truth	Yes	9	1	10
	No	9	981	990
		18	982	1000

$$\text{Accuracy} = \frac{990}{1000} = .99$$

Accuracy is Usually a Bad Choice

- Assumes the costs of FPs vs FNs are equal
- Which model is better for detecting a deadly disease?

Model 1

		Predicted Yes	**Predicted** No	
Truth	Yes	80	0	100
Truth	No	20	100	100
		100	100	200

$$\text{Accuracy} = \frac{180}{200} = .90$$

Model 2

		Predicted Yes	**Predicted** No	
Truth	Yes	80	20	100
Truth	No	0	100	100
		80	120	200

$$\text{Accuracy} = \frac{180}{200} = .90$$

Precision

- ***Precision:*** % of "yes" predictions that are correct
  - Does model make *precise* "yes" predictions?

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

		Predicted		
		Yes	No	
Truth	Yes	15	5	20
	No	0	80	80
		15	85	100

$\text{Precision} = ?$

		Predicted		
		Yes	No	
Truth	Yes	15	5	20
	No	5	75	80
		20	80	100

$\text{Precision} = ?$

Recall

- ***Recall:*** % of truth "yes" were correctly predicted as "yes"
  - Does model detect the most disease carriers?

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

Predicted

		Yes	No	
Truth	Yes	20	0	20
	No	5	75	80
		25	75	100

$\text{Recall} = ?$

Predicted

		Yes	No	
Truth	Yes	15	5	20
	No	0	80	80
		15	85	100

$\text{Recall} = ?$

Example: Duck Hunt

- High precision = ???
- High recall = ???
- High precision and recall = ??

NPV: "Precision of no"

- ***Negative Predictive Value:*** % of "no" predictions that are correct

$$\text{NPV} = \frac{TN}{TN + FN}$$

		Predicted: Yes	Predicted: No	
Truth	Yes	15	5	20
Truth	No	0	80	80
		15	85	100

$$\text{NPV} = ?$$

		Predicted: Yes	Predicted: No	
Truth	Yes	15	5	20
Truth	No	5	75	80
		20	80	100

$$\text{NPV} = ?$$

TNR: "Recall of no"

- ***True Negative Rate:*** % of truth "no" were correctly predicted as "no"
  - aka *Specificity*

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

Truth		Predicted: Yes	Predicted: No	
	Yes	20	0	20
	No	5	75	80
		25	75	100

$$\text{TNR} = ?$$

Truth		Predicted: Yes	Predicted: No	
	Yes	15	5	20
	No	0	80	80
		15	85	100

$$\text{TNR} = ?$$

F1 Score

- ***F1 Score:*** Harmonic mean between precision and recall

$$F_1 = \frac{2 \ * \text{Precision} \ * \text{Recall}}{\text{Precision} + \text{Recall}}$$

		Predicted		
		Yes	No	
Truth	Yes	18	2	20
	No	2	78	80
		20	80	100

$F_1 = ?$

		Predicted		
		Yes	No	
Truth	Yes	10	10	20
	No	2	78	80
		12	88	100

$F_1 = ?$

Exercise

		Predicted		
		Yes	No	
Truth	Yes	300	100	400
	No	20	580	600
		320	680	1000

- Accuracy = ?
- Precision = ?
- Recall = ?
- NPV (Precision of No) = ?
- TNR (Recall of Yes) = ?

Per-class, Macro, and Weighted

		Predicted		
		Yes	No	
Truth	Yes	18	2	20
	No	5	75	80
		23	77	100

$$\text{Precision} = \frac{18}{23} = 0.78$$

$$\text{Recall} = \frac{18}{20} = 0.90$$

$$F_1 = \frac{2 * 0.78 * 0.90}{0.78 + 0.90} = 0.84$$

	Precision	Recall	F1 Score	Support
Yes	0.78	0.90	0.84	20
No	0.97	0.94	0.96	80
Macro Avg	0.88	0.92	0.90	100
Weighted Avg	0.94	0.93	0.93	100

NPV

TNR

Decision Threshold

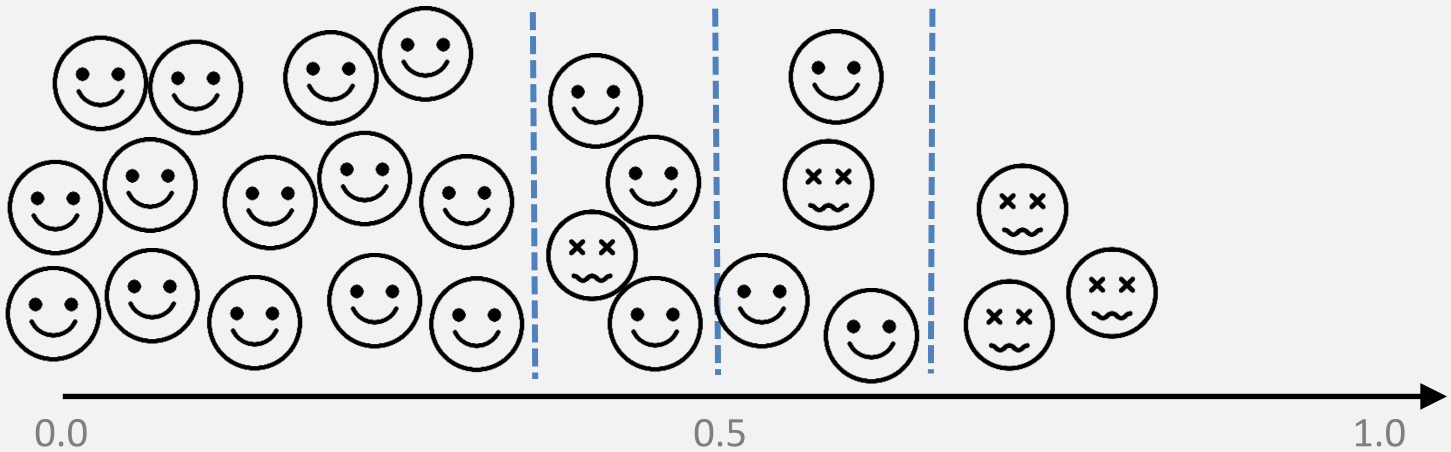
- Most classifiers actually predict a probability between [0, 1]
- You turn into a "yes" or "no" decision by using a threshold
  - E.g., threshold = 0.5

Truth target	Predicted Prob.	Predicted target
Yes	0.67	Yes
Yes	0.21	No
No	0.11	No
No	0.01	No
Yes	0.98	Yes
Yes	0.78	Yes
Yes	0.45	No

- By altering the threshold value up or down, you can trade-off precision and recall

Visual Example

Higher Recall, Lower Precision ⟵ ⟶ Lower Recall, Higher Precision



Threshold = 0.40

Truth		Predicted Yes	Predicted No	
	Yes	5	0	5
	No	6	14	20
		11	14	25

Precision = 45%
Recall = 100%

Threshold = 0.50

Truth		Predicted Yes	Predicted No	
	Yes	4	1	5
	No	3	17	20
		7	18	25

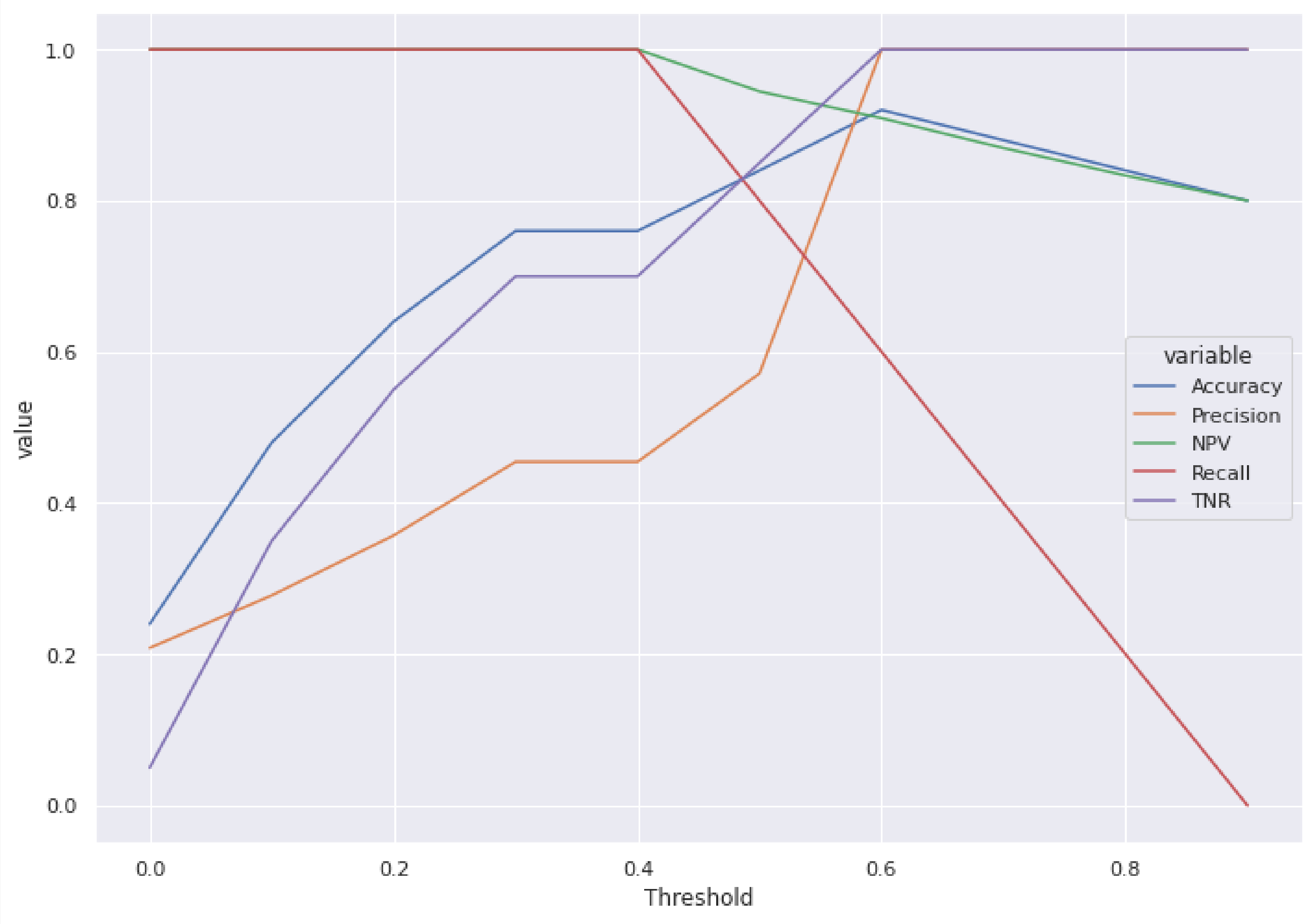
Precision = 57%
Recall = 80%

Threshold = 0.60

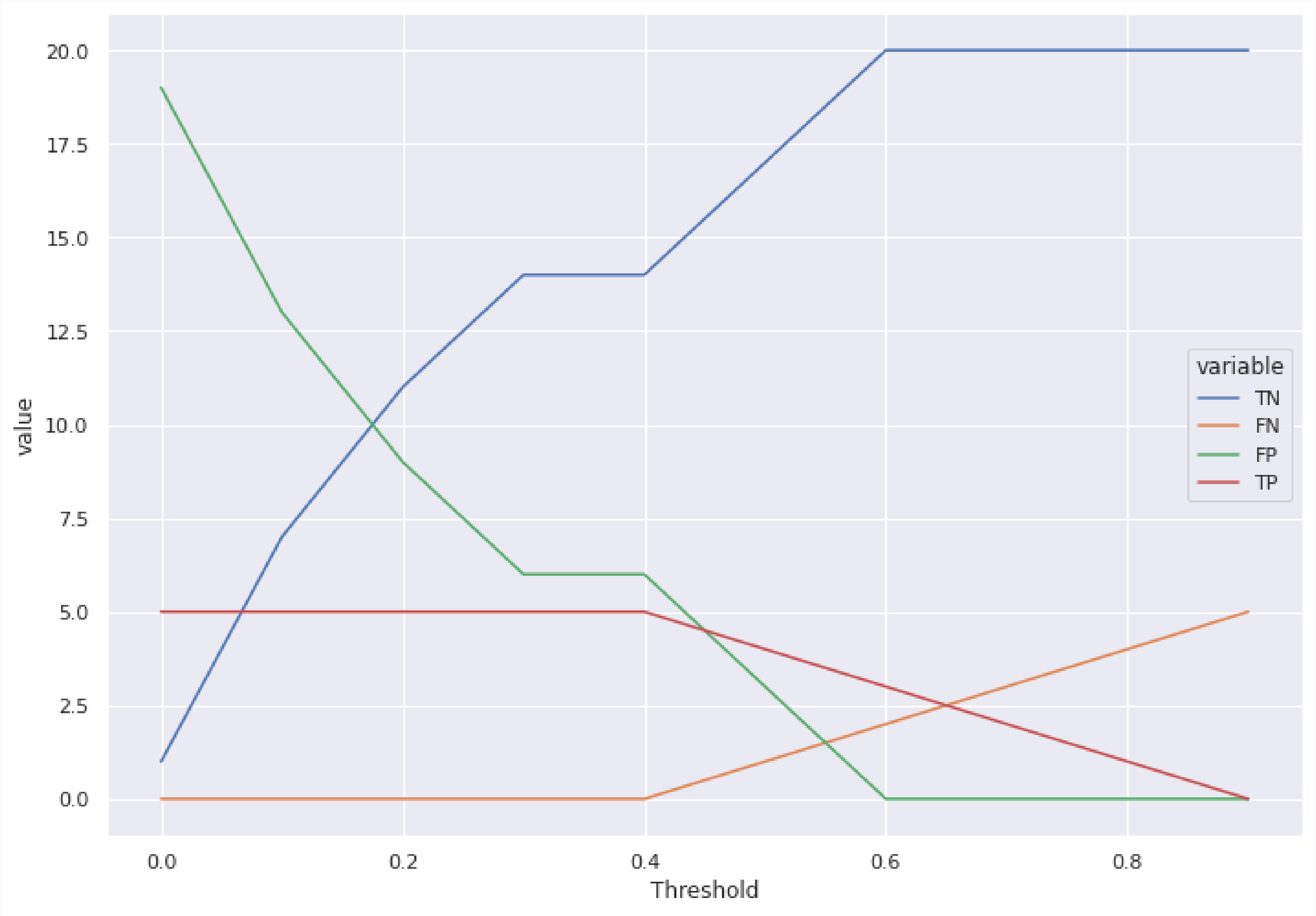
Truth		Predicted Yes	Predicted No	
	Yes	3	2	5
	No	0	20	20
		3	22	25

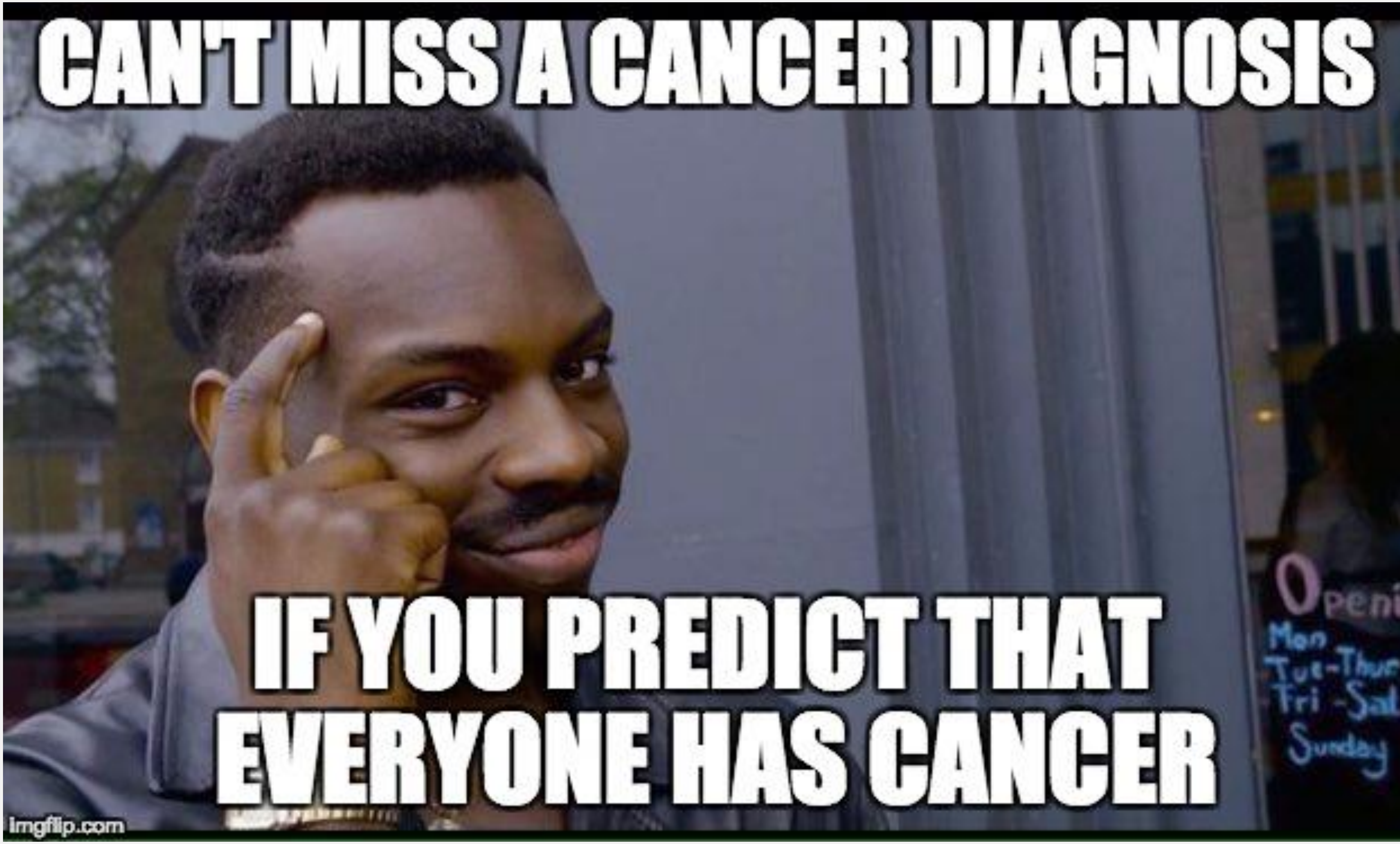
Precision = 100%
Recall = 60%

Visual Example



Visual Example




CAN'T MISS A CANCER DIAGNOSIS
IF YOU PREDICT THAT EVERYONE HAS CANCER
imgflip.com

Decision Threshold Trade-off

- Example: Marketing campaign
  - Selling cars with profit of $5K/each
  - Want to snail-mail brochures to potential customers ($5/each)
  - Model predicts which customers will respond to offer
    - Yes = will buy car
    - No = will not buy car
  - If model predicts yes, then you will send brochure to customer
  - Which threshold is better?

Threshold = 0.30

		Predicted Yes	Predicted No	
Truth	Yes	176	24	200
	No	153	647	800
		329	671	1000

Precision = 53%
Recall = 88%

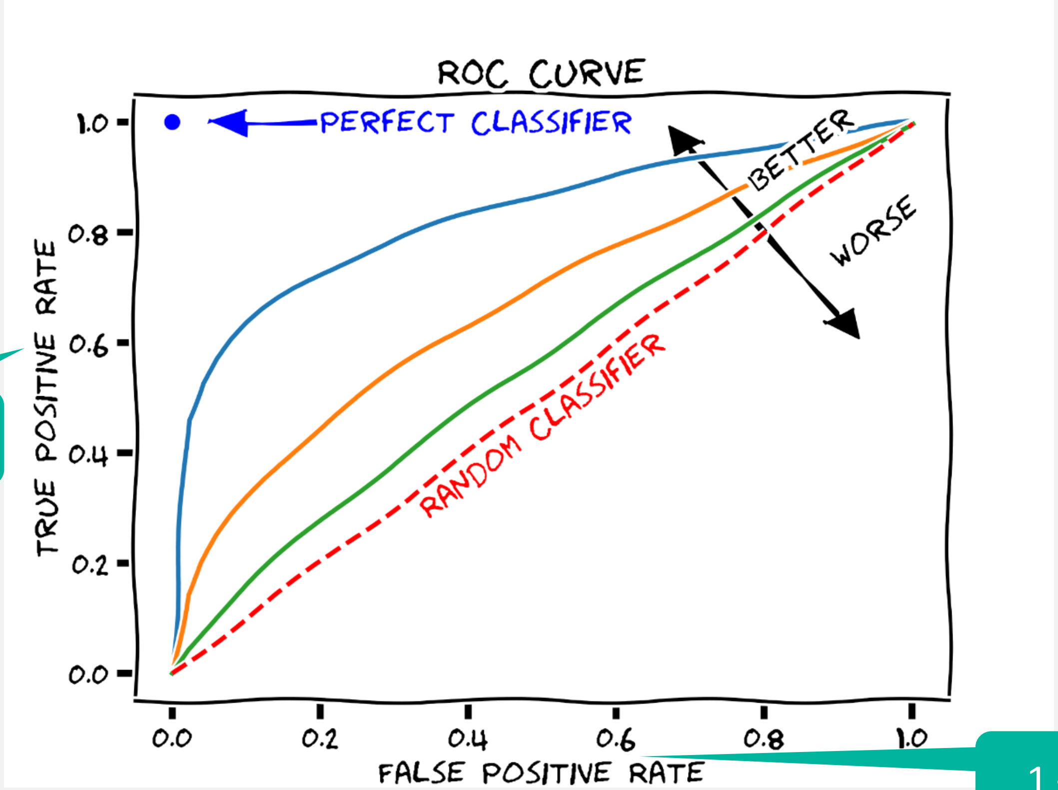
Threshold = 0.70

		Predicted Yes	Predicted No	
Truth	Yes	89	111	200
	No	19	781	800
		108	892	1000

Precision = 82%
Recall = 45%

ROC Curve

- ***ROC Curve***: a graphical plot that simultaneously displays two metrics (i.e., TPR and FPR) for many different threshold values



Recall of 1

1 – TNR (Recall of 0)

Rather than a single number, it's a graphic!

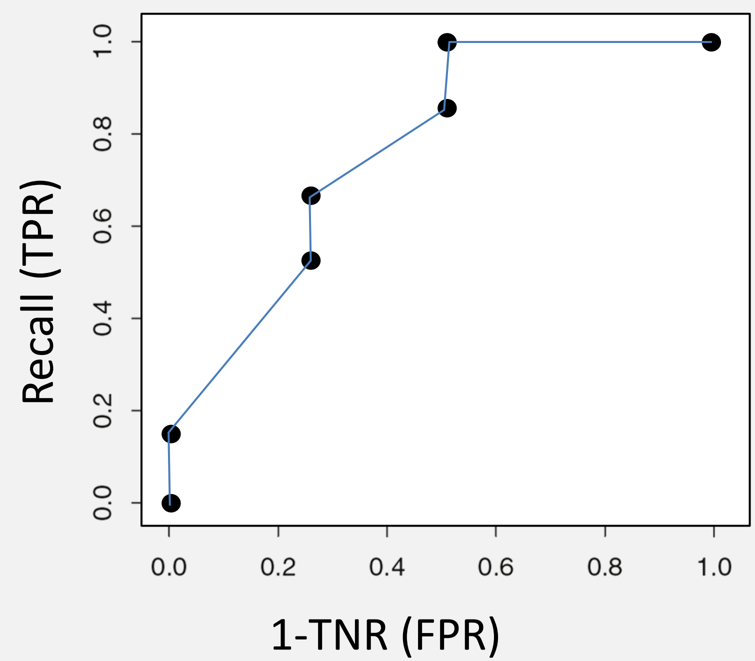
- Shows recall (aka TPR) against the FPR (1 – TNR)
  - Each point is calculated by choosing a different threshold

Drawing the ROC Curve

Model 1

Thresholds

Truth target	Prob.	0.0	0.2	0.4	0.5	0.6	0.8	1.0
Yes	0.67	Yes	Yes	Yes	Yes	Yes	No	No
Yes	0.21	Yes	Yes	No	No	No	No	No
No	0.11	Yes	No	No	No	No	No	No
No	0.01	Yes	No	No	No	No	No	No
Yes	0.98	Yes	Yes	Yes	Yes	Yes	Yes	No
No	0.78	Yes	Yes	Yes	Yes	Yes	No	No
Yes	0.45	Yes	Yes	Yes	No	No	No	No
Yes	0.53	Yes	Yes	Yes	Yes	No	No	No
No	0.40	Yes	Yes	Yes	No	No	No	No
Yes	0.60	Yes	Yes	Yes	Yes	Yes	No	No
	Recall:	100%	100%	83%	67%	50%	17%	0%
	1-TNR :	100%	50%	50%	25%	25%	0%	0%

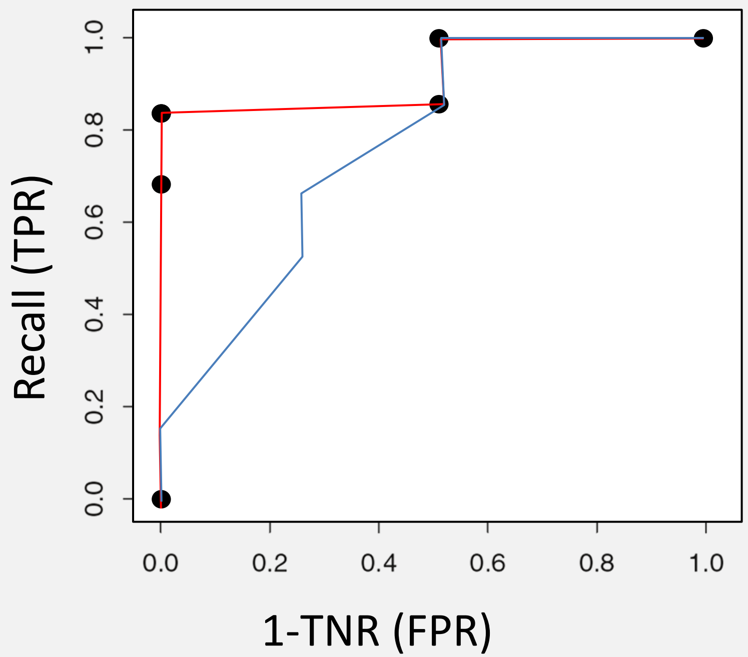


Drawing the ROC Curve

Model 2

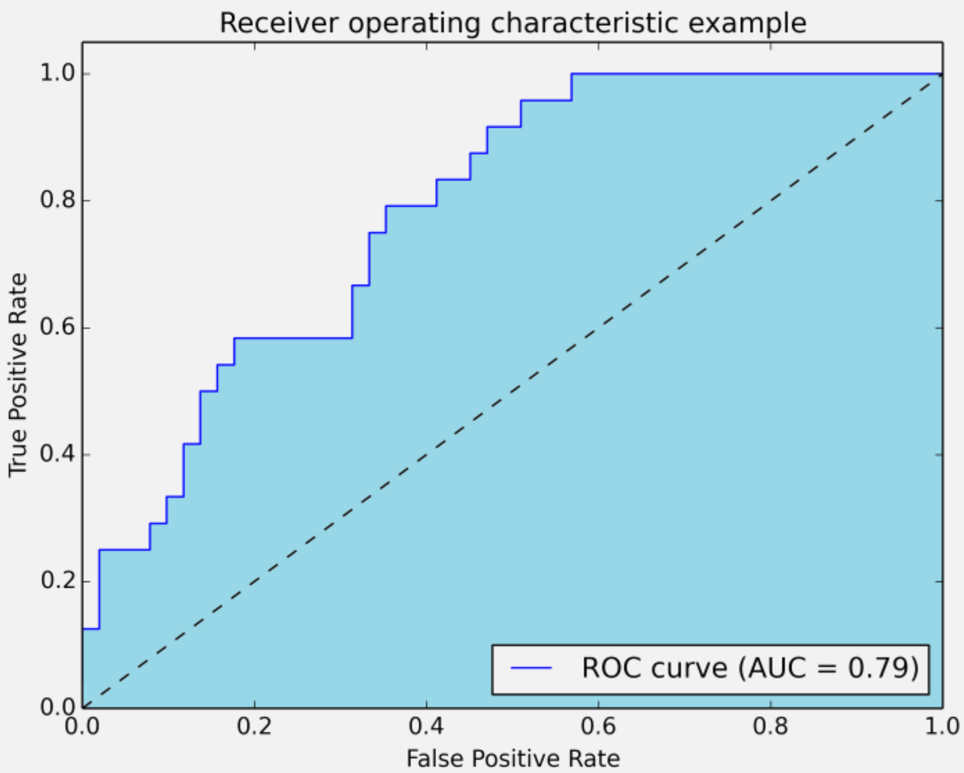
Thresholds

			Truth target	Prob.	0.0	0.2	0.4	0.5	0.6	0.8	1.0
			Yes	0.96	Yes	Yes	Yes	Yes	Yes	Yes	No
			Yes	0.80	Yes	Yes	Yes	Yes	Yes	Yes	No
			No	0.11	Yes	No	No	No	No	No	No
			No	0.05	Yes	No	No	No	No	No	No
			Yes	0.98	Yes	Yes	Yes	Yes	Yes	Yes	No
			No	0.40	Yes	Yes	Yes	No	No	No	No
			Yes	0.30	Yes	Yes	No	No	No	No	No
			Yes	0.89	Yes	Yes	Yes	Yes	Yes	Yes	No
			No	0.40	Yes	Yes	Yes	No	No	No	No
			Yes	0.75	Yes	Yes	Yes	Yes	Yes	No	No
				Recall:	100%	100%	83%	83%	83%	67%	0%
				1-TNR :	100%	50%	50%	0%	0%	0%	0%



Area Under Curve (AUC)

- ROC plots are cool, but a single number is *really* cool
  - Easier to compare two classifiers
  - Just like the F1 Score combines Precision and Recall
- *AUC* is the area under the ROC curve
  - Remember calculus, anyone?



1.0: perfect prediction
0.9: excellent prediction
0.8: good prediction
0.7: mediocre prediction
0.6: poor prediction
0.5: random prediction
< 0.5: something wrong!

WARNINGS ABOUT TERMINOLOGY

Terminology is Hard

		Predicted		
		Yes	No	
Actual	Yes	18	2	20
	No	5	75	80
		23	77	100

Predicted Condition

Positive, Condition Positive

Negative, Condition Negative

TP, hit

Actual Condition, True Condition

FN, type II error, miss

TN, Correct rejection

FP, type I error, false alarm

Total Population

	Precision	Recall	F1 Score	Support
Yes	0.78	0.90	0.84	20
No	0.97	0.94	0.96	80
Macro Avg	0.88	0.92	0.90	100
Weighted Avg	0.94	0.93	0.93	100

NPV

FOR

FDR

Precision, PPV

FPR

Specificity, TNR

Recall, Sensitivity, hit rate, TPR

FNR

Wikipedia's Page

		Predicted condition		Sources: [13][14][15][16][17][18][19][20] view · talk · edit	
	Total population $= P + N$	**Positive (PP)**	**Negative (PN)**	Informedness, bookmaker informedness (BM) $= TPR + TNR - 1$	Prevalence threshold (PT) $= \frac{\sqrt{TPR \times FPR} - FPR}{TPR - FPR}$
Actual condition	**Positive (P)**	**True positive (TP),** hit	**False negative (FN),** type II error, miss, underestimation	True positive rate (TPR), recall, sensitivity (SEN), probability of detection, hit rate, power $= \frac{TP}{P} = 1 - FNR$	False negative rate (FNR), miss rate $= \frac{FN}{P} = 1 - TPR$
	Negative (N)	**False positive (FP),** type I error, false alarm, overestimation	**True negative (TN),** correct rejection	False positive rate (FPR), probability of false alarm, fall-out $= \frac{FP}{N} = 1 - TNR$	True negative rate (TNR), specificity (SPC), selectivity $= \frac{TN}{N} = 1 - FPR$
	Prevalence $= \frac{P}{P + N}$	Positive predictive value (PPV), precision $= \frac{TP}{PP} = 1 - FDR$	False omission rate (FOR) $= \frac{FN}{PN} = 1 - NPV$	Positive likelihood ratio (LR+) $= \frac{TPR}{FPR}$	Negative likelihood ratio (LR−) $= \frac{FNR}{TNR}$
	Accuracy (ACC) $= \frac{TP + TN}{P + N}$	False discovery rate (FDR) $= \frac{FP}{PP} = 1 - PPV$	Negative predictive value (NPV) $= \frac{TN}{PN}$ $= 1 - FOR$	Markedness (MK), deltaP (Δp) $= PPV + NPV - 1$	Diagnostic odds ratio (DOR) $= \frac{LR+}{LR-}$
	Balanced accuracy (BA) $= \frac{TPR + TNR}{2}$	F_1 score $= \frac{2PPV \times TPR}{PPV + TPR} = \frac{2TP}{2TP + FP + FN}$	Fowlkes–Mallows index (FM) $= \sqrt{PPV \times TPR}$	Matthews correlation coefficient (MCC) $= \sqrt{TPR \times TNR \times PPV \times NPV} - \sqrt{FNR \times FPR \times FOR \times FDR}$	Threat score (TS), critical success index (CSI), Jaccard index $= \frac{TP}{TP + FN + FP}$

https://en.wikipedia.org/wiki/Confusion_matrix

Careful About Axes Orientation!

These slides

Truth \ Predicted	Yes	No
Yes	30	2
No	11	24

scikit-learn

Truth \ Predicted	No	Yes
No	24	11
Yes	2	30

Predicted \ Truth	Yes	No
Yes	30	11
No	2	24

Predicted \ Truth	No	Yes
No	24	2
Yes	11	30

MODEL FINANCIALS:
KCU EXERCISE

How Much Money Does a Model Make/Save?

- Current situation:
  - 2,000 customers
  - 5% churn per month
  - Each churn costs an average of $400
- Cost of current situation: 100*$400 = $40,000
- Churn prediction model:
  - If model predicts Yes, will offer 20% discount offer
    - Average cost of $50
    - Assume everyone accepts offer
  - Current model has below confusion matrix
- How much does the model save them per month?

		Predicted		
		Yes	No	
Truth	Yes	90	10	100
	No	285	1615	1900
		375	1625	

Confusion Matrix

$\odot$

		Predicted	
		Yes	No
Truth	Yes	$50	$400
	No	$50	-

Cost Matrix

Cost per transaction

=

		Predicted	
		Yes	No
Truth	Yes	$4,500	$4,000
	No	$14,250	

Total Cost

$22,750

KCU's Model to Predict Fraudulent Transactions

- Assume:
  - 10,000 transactions per month
  - 1% of transactions are fraudulent
  - Each fraudulent transaction costs $10,000
  - It costs $100 to investigate potentially fraudulent transactions
- What is the cost of fraudulent transactions now (without model)?
- KCU builds a model with below confusion matrix
  - If predicts Yes = Fraud, then investigate
  - If predicts No = Not Fraud, then don't investigate
- How much money does model save per month?

		Predicted	
		Yes	No
Truth	Yes	76	24
	No	1485	8415

Confusion Matrix

$\odot$

		Predicted	
		Yes	No
Truth	Yes		
	No		

Cost Matrix

Cost per transaction

=

		Predicted	
		Yes	No
Truth	Yes		
	No		

Total Cost

RESOURCES

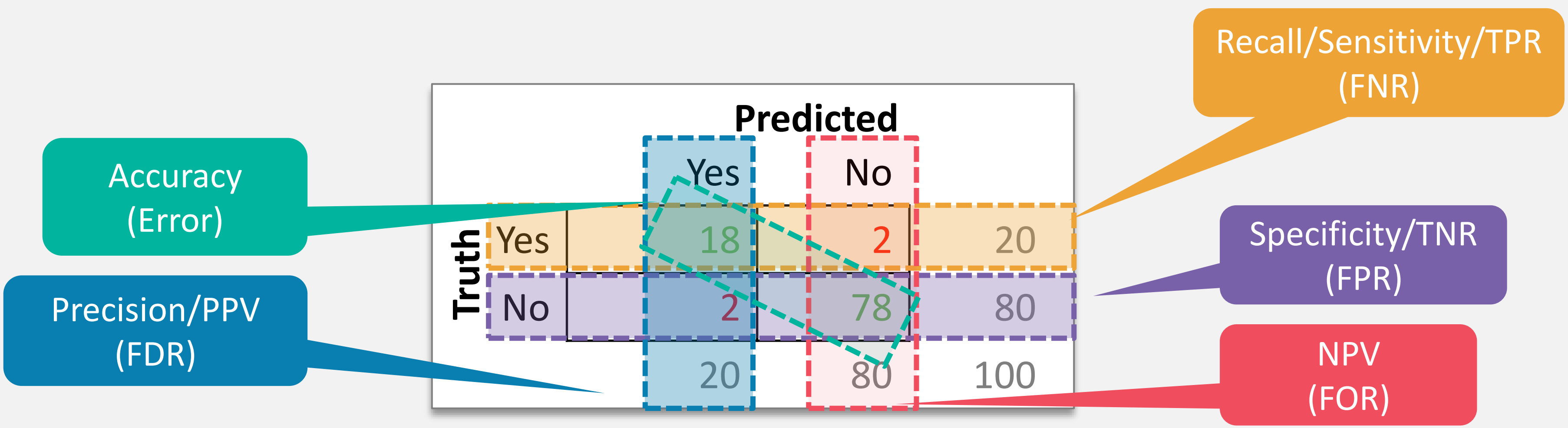
Resources

- Coding Tutorial Files by Uncle Steve
  - slides_performance.ipynb

SUMMARY

Uncle Steve's Table of Performance Metrics

Metric (Inverse)	Summary	Good when you want:
Accuracy (Error)	% of predictions that are correct	✓ Simplicity, have balanced data and equal costs between FP and FN
Precision/PPV (FDR)	% of "yes" predictions that are correct	✓ To measure how precise with "yes" the model is
NPV	% of "no" predictions that are correct	✓ To measure how precise with "no" the model is
Recall/Sensitivity/TPR (FNR)	% of "yes" cases that were predicted as "yes"	✓ To measure if the model misses any "yes"s
TNR/Specificity	% of "no" cases that were predicted as "no"	✓ To measure if the model misses any "no"s
F1 Score	Harmonic mean of precision and recall	✓ An overall measure of model performance
ROC Curve	Shows TPR vs FPR for all possible threshold values	✓ An overall measure of model performance
AUC	Measures area under ROC curve	✓ An overall measure of model performance
Log Loss	Like accuracy, but takes into account *how* right or wrong the predictions are	✓ To penalize models for being very wrong



APPENDIX

Log Loss

- Say you build two models to predict loan default
  - 1 = Yes = default
  - 0 = No = paid
- You test each model with a test instance with truth = 0 (paid)
  - Model 1 predicts .51 probability
  - Model 2 predicts .98 probability
  - Both are wrong, but which is worse?
- ***Log Loss*** is a metric that takes the probability into account, and how far it is from the truth value.
- Heavily penalizes predictions that are confident but incorrect.

$$\text{Log Loss} = -[y \ln(p) + (1-y)\ln(1-p)\ ]$$

y is actual

p is predicted

Model 1 $\text{Log Loss} = -[0\ln(.51) + (1-0)\ln(1-.51)\ ]$ = .71

Model 2 $\text{Log Loss} = -[0\ln(.98) + (1-0)\ln(1-.98)\ ]$ = 3.9

Log Loss Examples

Truth	Predicted		Log loss
No	0.01	$-[0\ln(.01) + (1-0)\ln(1-.01)\ ] =$	0.01
No	0.10	$-[0\ln(.10) + (1-0)\ln(1-.10)\ ] =$	0.11
No	0.30	…	0.36
No	0.50		0.69
No	0.80		1.61
No	0.90		2.30
No	1.00		35.0
Yes	0.01		4.60
Yes	0.10		2.30
Yes	0.30		1.20
Yes	0.50		0.69
Yes	0.80		0.22
Yes	0.90	$-[1\ln(.90) + (1-1)\ln(1-.90)\ ] =$	0.10
Yes	1.00	$-[1\ln(1.0) + (1-1)\ln(1-1.0)\ ] =$	0.00

Take average of all instance's log losses

Specificity

- ***Specificity**:* Percentage of truth "no"s that were correctly predicted
  - Is the disease diagnosis specific? Correctly reject healthy patients?
  - Aka: *true negative rate (TNR), Recall of "No" class*
  - *1 – specificity = False positive rate (FPR)*

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

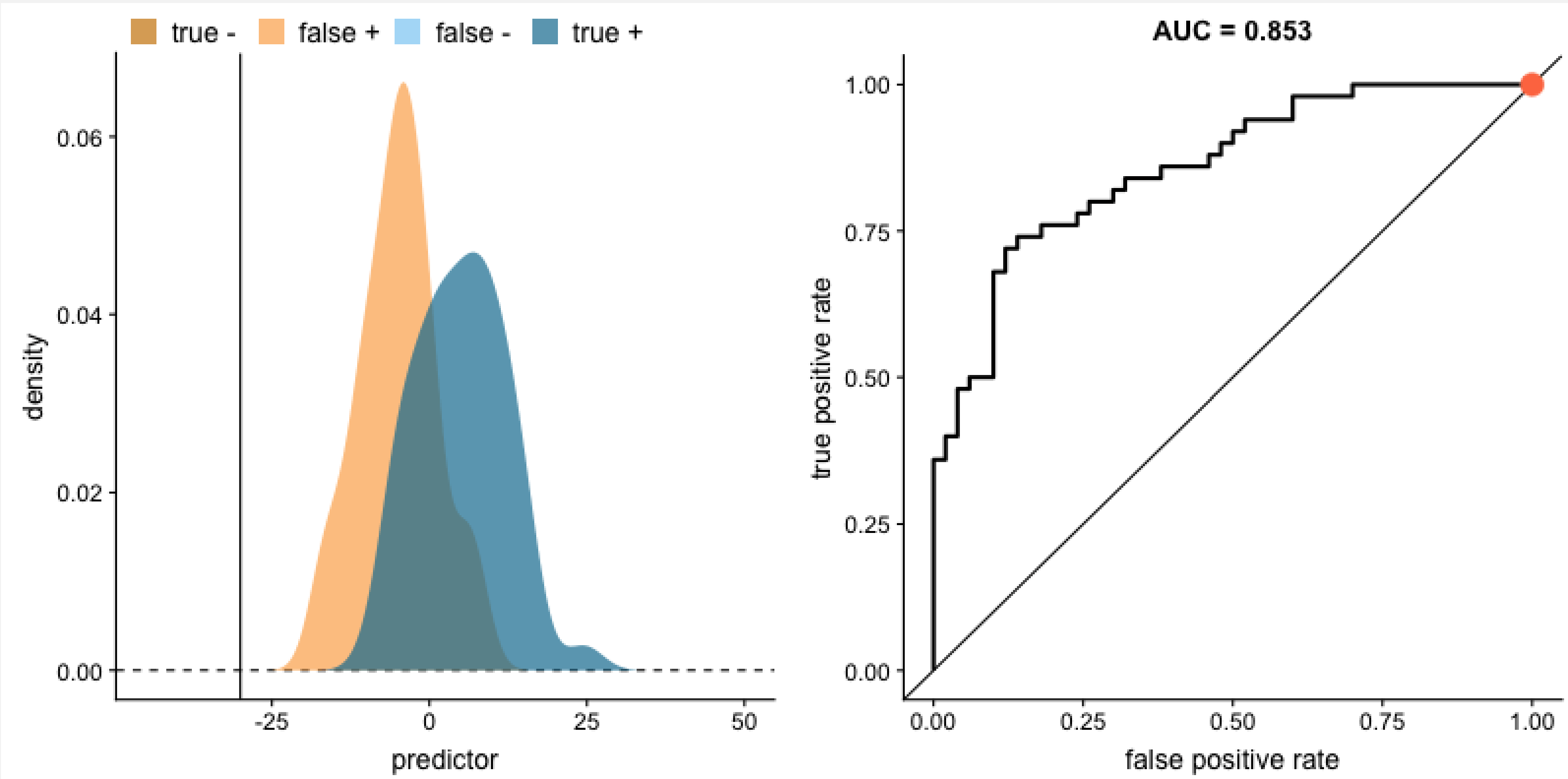
		Predicted Yes	Predicted No	
Truth	Yes	15	5	20
Truth	No	2	78	80
		17	83	100

$$\text{Specificity} = ?$$

		Predicted Yes	Predicted No	
Truth	Yes	20	0	20
Truth	No	40	40	80
		60	40	100

$$\text{Specificity} = ?$$

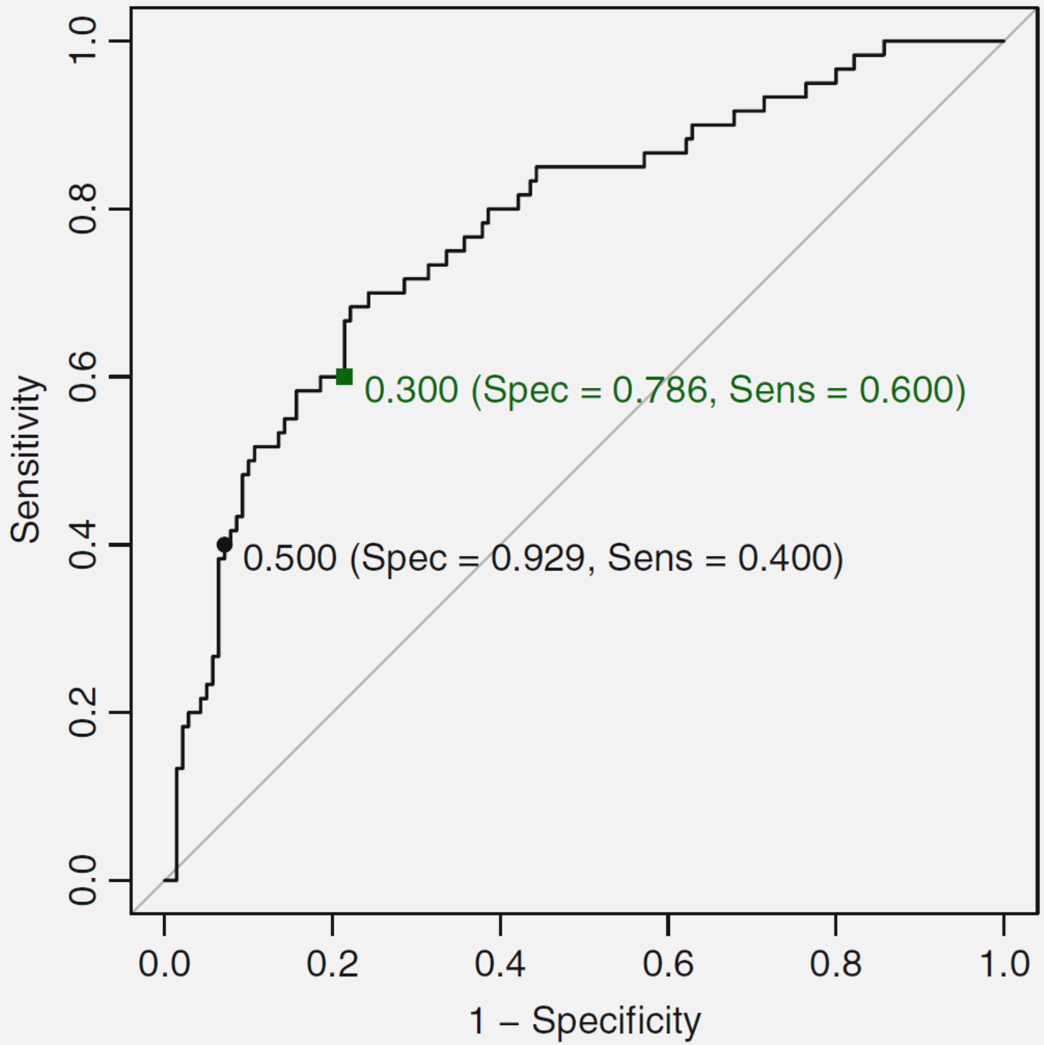
Drawing the ROC Curve



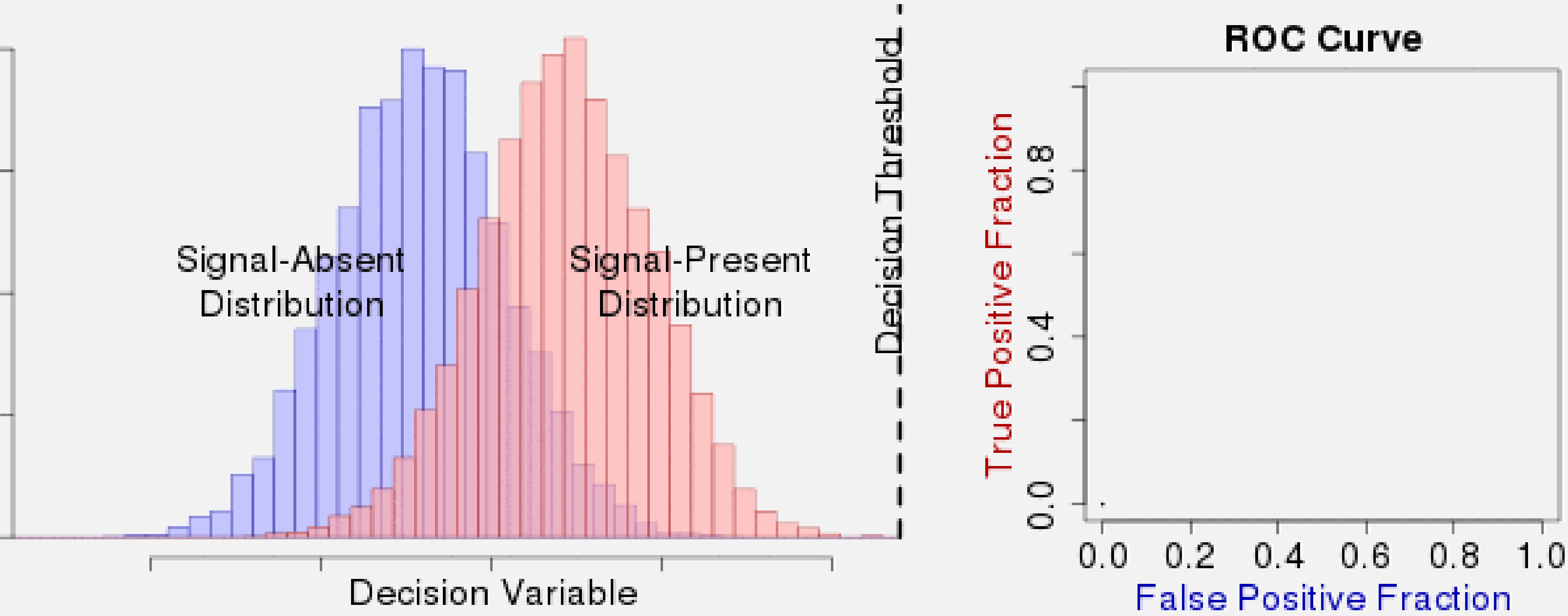
Source: https://github.com/dariyasydykova/open_projects

Drawing the ROC Curve

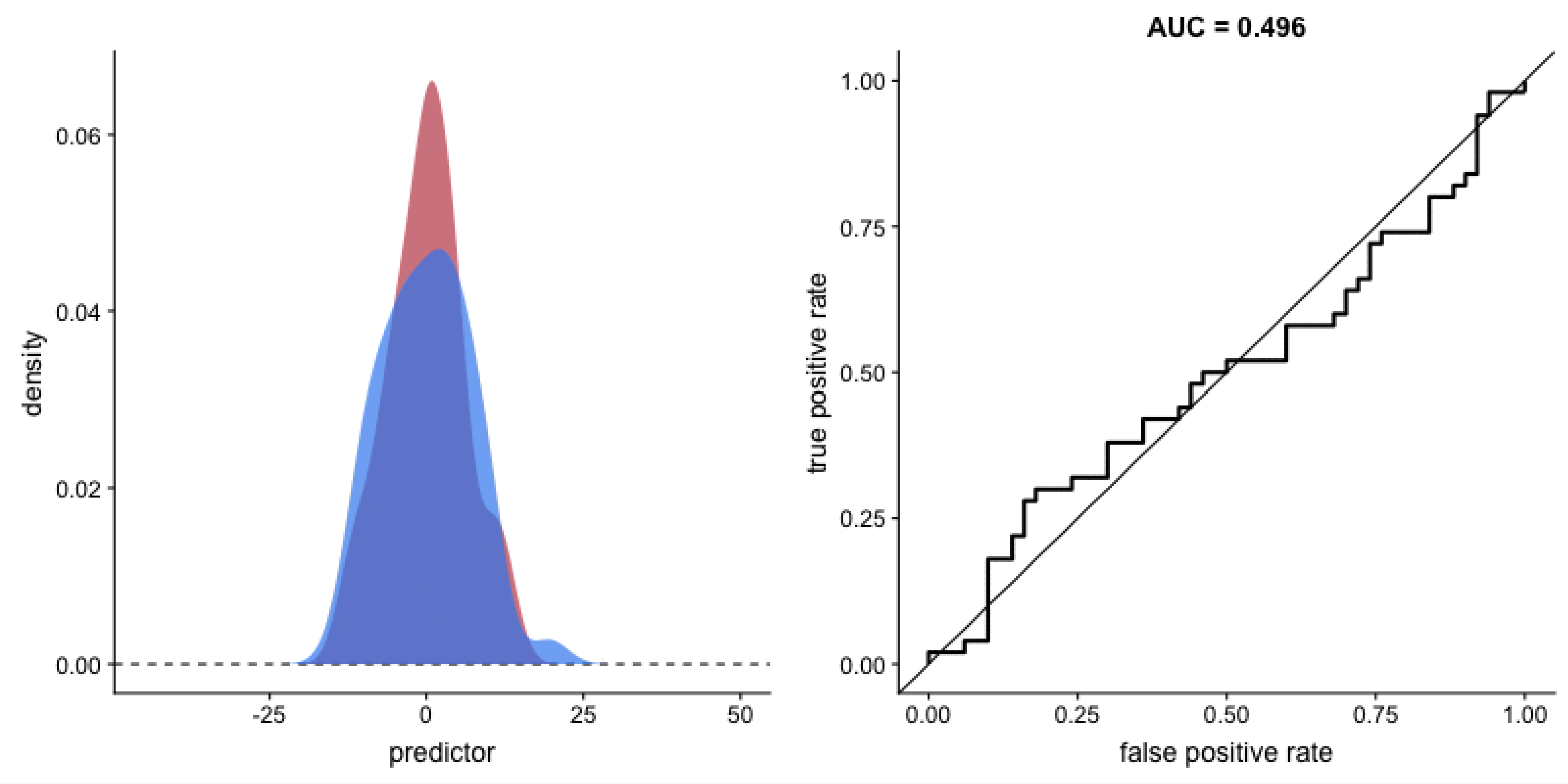
- Set threshold to 1.00
  - Calculate TPR, FPR
  - Plot point
- Set threshold to 0.99
  - Calculate TPR, FPR
  - Plot point
- …
- Set threshold to 0.01
  - Calculate TPR, FPR
  - Plot point
- Set threshold to 0.00
  - Calculate TPR, FPR
  - Plot point



Drawing the ROC Curve: Animation

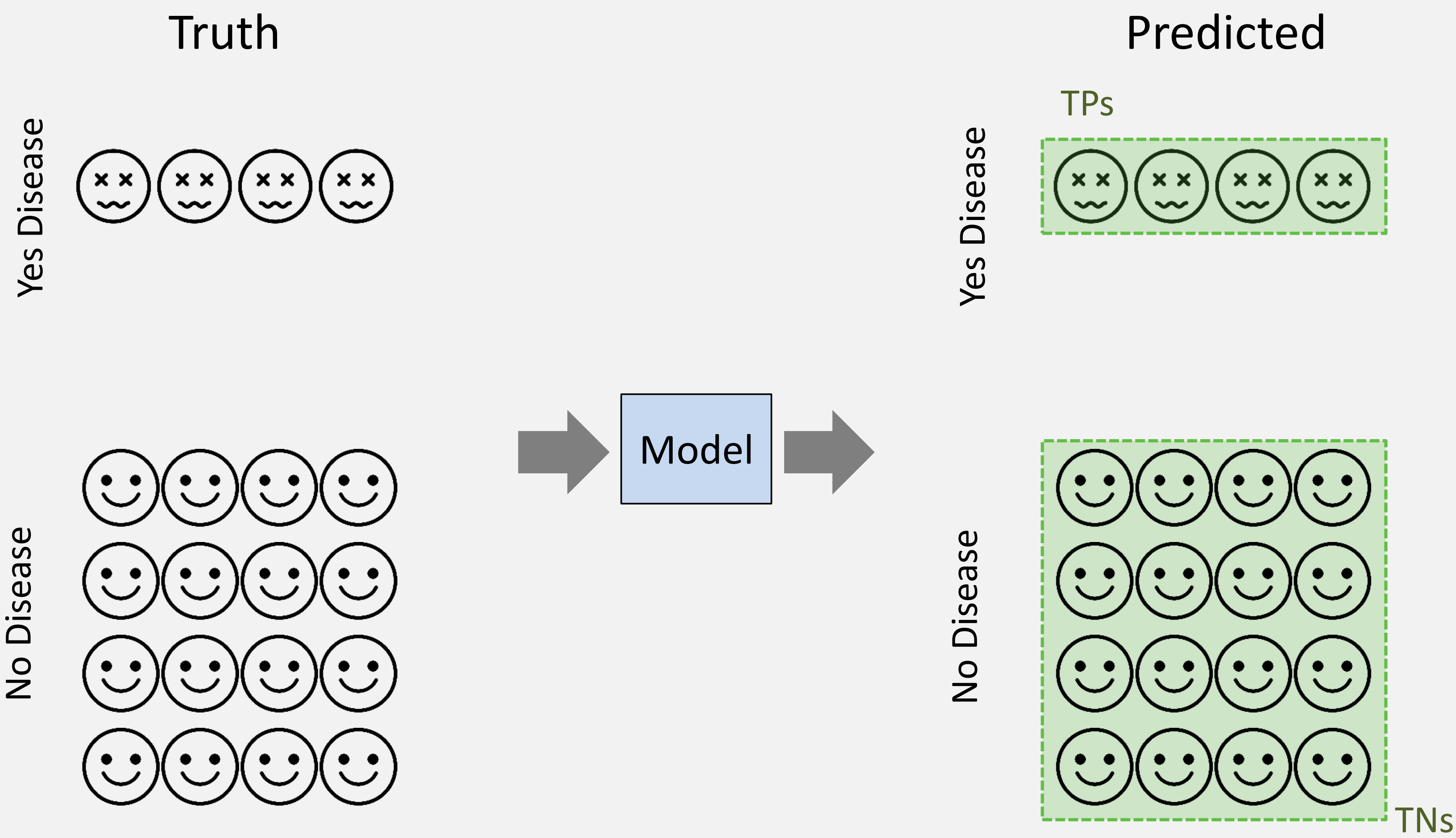


ROC



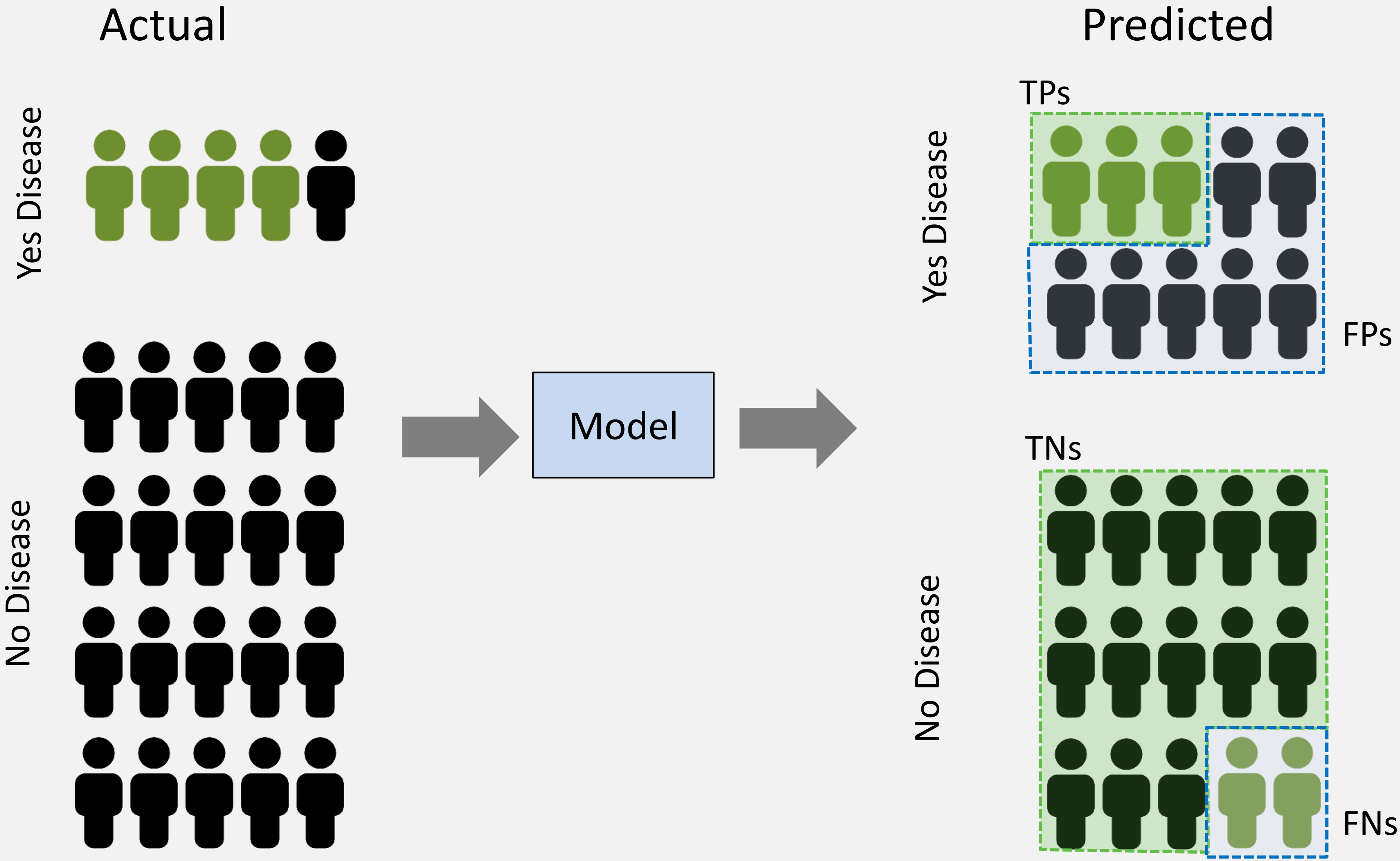
Source: https://github.com/dariyasydykova/open_projects

The Perfect Model/Test



The Perfect Model® does not exist.

Another Example



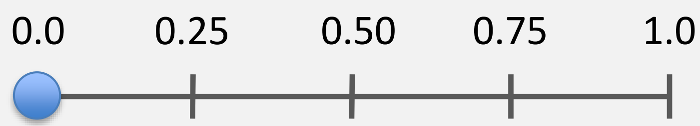
Tradeoff Between Sensitivity and Specificity

- It's easy to build a classifier with perfect sensitivity!
  - Threshold = 0.0 →Predict "yes" always
- It's easy to build a classifier with perfect specificity!
  - Threshold = 1.0 →Predict "no" always
- In both cases, there's a cost
  - Higher sensitivity = Lower specificity
  - Lower sensitivity = Higher specificity
- The best combination depends on the application/domain

	0.0	0.25	0.50	0.75	1.0
Sensitivity	**1.000**	0.833	0.500	0.167	0.000
Specificity	0.000	0.500	0.750	0.750	**1.000**
Accuracy	0.600	**0.700**	0.600	0.400	0.400
Error	0.400	**0.300**	0.400	0.600	0.600
Precision	0.600	0.714	**0.750**	0.500	-
Recall	**1.000**	0.833	0.500	0.167	0.000
F1	0.750	**0.769**	0.600	0.250	-

Same example dataset as before

Example (1/5)



Threshold = 0.0

			Truth target	Predicted Prob.	Predicted target
			Yes	0.67	Yes
			Yes	0.21	Yes
			No	0.11	Yes
			No	0.01	Yes
			Yes	0.98	Yes
			No	0.78	Yes
			Yes	0.45	Yes
			Yes	0.40	Yes
			No	0.40	Yes
			Yes	0.60	Yes

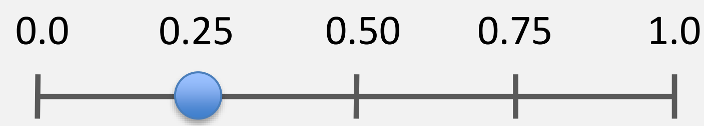
Predicted

Actual	Yes (Predicted)	No (Predicted)	
Yes	6	$\boxed{0}$ FNs	6
No	FPs $\boxed{4}$	0	4
	10	0	10

Accuracy = 60%
Sensitivity = 100%
Specificity = 0%

Precision = 60%
Recall = 100%
F1-Score = 75%

Example (2/5)



Threshold = 0.25

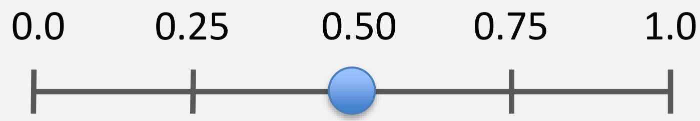
Truth target	Predicted Prob.	Predicted target
Yes	0.67	Yes
Yes	0.21	No
No	0.11	No
No	0.01	No
Yes	0.98	Yes
No	0.78	Yes
Yes	0.45	Yes
Yes	0.40	Yes
No	0.40	Yes
Yes	0.60	Yes

		Predicted		
		Yes	No	
Actual	Yes	5	1 FNs	6
	No	FPs 2	2	4
		7	3	10

Accuracy = 70%
Sensitivity = 83%
Specificity = 50%

Precision = 71%
Recall = 83%
F1-Score = 77%

Example (3/5)



Threshold = 0.50

			Truth target	Predicted Prob.	Predicted target
			Yes	0.67	Yes
			Yes	0.21	No
			No	0.11	No
			No	0.01	No
			Yes	0.98	Yes
			No	0.78	Yes
			Yes	0.45	No
			Yes	0.40	No
			No	0.40	No
			Yes	0.60	Yes

Predicted

Actual	Yes	No	
Yes	3	3 FNs	6
No	FPs 1	3	4
	4	6	10

Accuracy	=	60%	Precision	=	75%
Sensitivity	=	50%	Recall	=	50%
Specificity	=	75%	F1-Score	=	60%

Example (4/5)

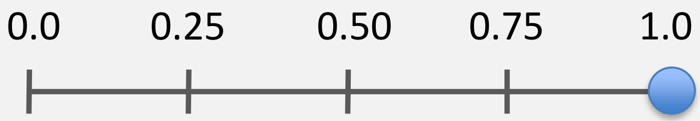
Threshold = 0.75

Truth target	Predicted Prob.	Predicted target
Yes	0.67	No
Yes	0.21	No
No	0.11	No
No	0.01	No
Yes	0.98	Yes
No	0.78	Yes
Yes	0.45	No
Yes	0.40	No
No	0.40	No
Yes	0.60	No

		Predicted		
		Yes	No	
Actual	Yes	1	5 FNs	6
	No	FPs 1	3	4
		2	8	10

Accuracy	= 40%		Precision	=	50%
Sensitivity	= 17%		Recall	=	17%
Specificity	= 75%		F1-Score	=	25%

Example (5/5)



Threshold = 1.0

Truth target	Predicted Prob.	Predicted target
Yes	0.67	No
Yes	0.21	No
No	0.11	No
No	0.01	No
Yes	0.98	No
No	0.78	No
Yes	0.45	No
Yes	0.40	No
No	0.40	No
Yes	0.60	No

		Predicted		
		Yes	No	
Actual	Yes	0	6 FNs	6
	No	FPs 0	4	4
		0	10	10

Accuracy	=	40%	Precision	=	NA
Sensitivity	=	0%	Recall	=	0%
Specificity	=	100%	F1-Score	=	NA

Example: VAERS Case Study

- Vaccine Adverse Event Reporting System: A US dataset of adverse events that occur after administration of vaccines
- ***Case study***:
  - **Situation**: 4% of patients that get H1N1 vaccine get severe anaphylaxis reaction → may cause death
  - **Objective**: Want to build prediction model to predict which patients will have bad reaction
  - **Data**: Collected 6K VAERS reports involving H1N1 vaccine, each labeled as either positive or negative for anaphylaxis
    - Number of positive reports = 237 (4%)

Example: VAERS Case Study

Tried many classification algorithms

Classifiers	Testing set			Validation set		
	macro-R	macro-P	macro-F*	macro-R	macro-P	macro-F*
NB	0.794 (10)	0.753 (4)	0.773	0.671 (11)	0.697 (4)	0.684
ME	0.701 (6)	0.863 (9)	0.773	0.577 (7.5)	0.775 (10.5)	0.661
DT	0.609 (2)	0.891 (12)	0.724	0.544 (1.5)	0.767 (9)	0.637
RPCT	0.642 (4)	0.872 (10)	0.740	0.544 (1.5)	0.732 (6.5)	0.624
BT	0.881 (13)	0.639 (2)	0.741	0.789 (12)	0.648 (2)	0.711
w-SVM	0.871 (12)	0.619 (1)	0.724	0.809 (13)	0.619 (1)	0.701
s-SVM	0.642 (4)	0.855 (7.5)	0.734	0.555 (4)	0.795 (12)	0.654
SB	0.708 (8)	0.836 (6)	0.767	0.574 (5)	0.677 (3)	0.622
MARS	0.707 (7)	0.809 (5)	0.755	0.576 (6)	0.733 (8)	0.645
RDA	0.831 (11)	0.649 (3)	0.729	0.640 (10)	0.702 (5)	0.669
RF	0.642 (4)	0.855 (7.5)	0.734	0.589 (9)	0.817 (13)	0.684
GAM	0.751 (9)	0.885 (11)	0.813	0.577 (7.5)	0.775 (10.5)	0.661
w-kNN	0.576 (1)	0.892 (13)	0.700	0.554 (3)	0.732 (6.5)	0.631

Example: VAERS Case Study

Performance of best classifier

		Predicted Yes	Predicted No	
Actual	Yes	212	25	237
	No	89	5674	5763
		301	5699	6000

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{5886}{6000} = .98$$

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{212}{301} = .70$$

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{212}{237} = .89$$

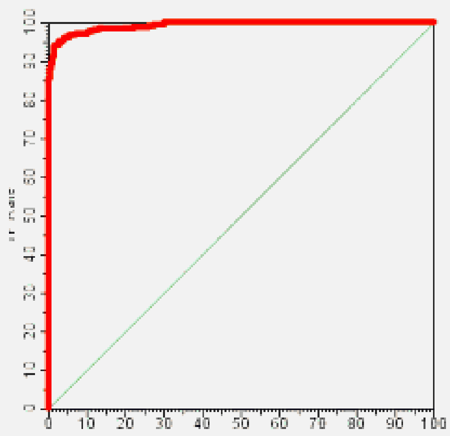
Example: VAERS Case Study

Discussion:

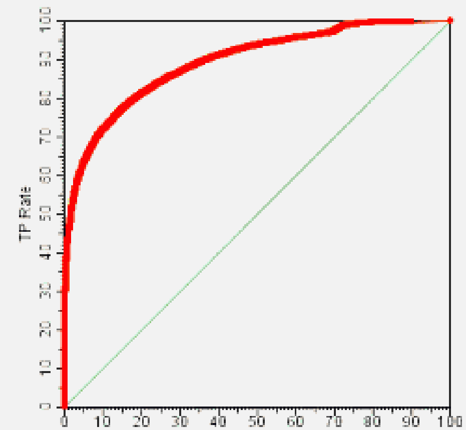
- If model were deployed on the 6K reports:
  - Of the 237 reports that are actually positive:
    - 212 (89%) correctly classified as positive
    - 25 (11%) incorrectly classified as negative
    - ***Is that good enough?***
  - Of the 301 reports classified as positive:
    - 212 (70%) would actually be positive
    - 89 (30%) would actually be negative
    - ***Is that good enough?***

ROC Curve Interpretation

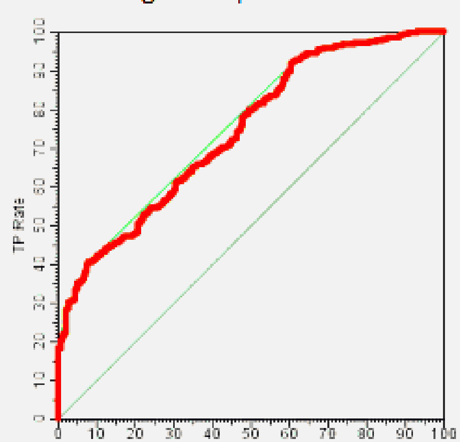
- Lines "up and to the left" are better
- Straight diagonal lines are no good → random guessing



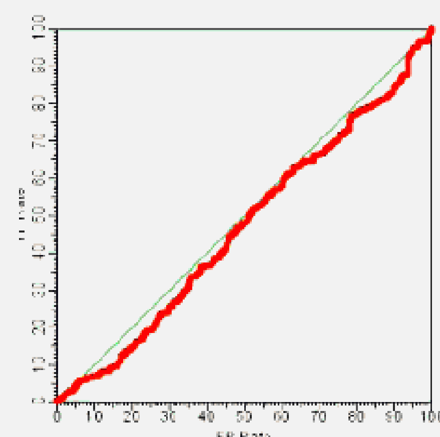
good separation



reasonable



poor separation



random separation