

Machine Learning for Finance

Lecture 10

FNT633

Forecasting **Part 3**

Performance metrics

Prepared by
Dr. Diaa Salama

- Forecasting
- Performance metrics (Confusion Matrix)

What Kind of Problem do I Need to Solve? How do I Solve it?

The Problem to Solve	The Category of Techniques	Covered in this Course
I want to group items by similarity. I want to find structure (commonalities) in the data	Clustering	K-means clustering
I want to discover relationships between actions or items	Association Rules	Apriori
I want to determine the relationship between the outcome and the input variables	Regression	Linear Regression Logistic Regression
I want to assign (known) labels to objects	Classification	Naïve Bayes Decision Trees
I want to find the structure in a temporal process I want to forecast the behavior of a temporal process	Time Series Analysis	ACF, PACF, ARIMA
I want to analyze my text data	Text Analysis	Regular expressions, Document representation (Bag of Words).

Agenda

- Forecasting (Seasonal Variations in Data)
- Confusion Matrixs
- Performance metrics

Week

11

Seasonal Variations in Data

SEASONAL VARIATIONS IN DATA

The multiplicative seasonal model can adjust trend data for seasonal variations in demand (jet skis, snow mobiles)

SEASONAL VARIATIONS IN DATA

Steps in the process:

1. Find **average** historical demand **for each season**
2. Compute the **average** demand **over all seasons**
3. Compute a **seasonal index** for each season
4. Estimate next year's total demand
5. Divide this estimate of total demand by the number of seasons, then multiply it by the seasonal index for that season

Seasonal Variations in Data

SEASONAL INDEX EXAMPLE

Month	Demand			Average 2010-2012	Average Monthly	Seasonal Index
	2010	2011	2012			
Jan	80	85	105	90	94	
Feb	70	85	85	80	94	
Mar	80	93	82	85	94	
Apr	90	95	115	100	94	
May	113	125	131	123	94	
Jun	110	115	120	115	94	
Jul	100	102	113	105	94	
Aug	88	102	110	100	94	
Sept	85	90	95	90	94	
Oct	77	78	85	80	94	
Nov	75	72	83	80	94	
Dec	82	78	80	80	94	

Seasonal Variations in Data

SEASONAL INDEX EXAMPLE

Seasonal Index	Average Monthly	Average 2010-2012	2012	Demand 2011	2010	Month
Jan	80	85	105	90	94	0.957

$$\text{Seasonal index} = \frac{\text{Average 2010-2012 monthly demand}}{\text{Average monthly demand}}$$

$$= 90/94 = .957$$

Jul	100	102	115	105	94
Aug	88	102	110	100	94
Sept	85	90	95	90	94
Oct	77	78	85	80	94
Nov	75	72	83	80	94
Dec	82	78	80	80	94

Seasonal Variations in Data

SEASONAL INDEX EXAMPLE

Month	Demand			Average 2010-2012	Average Monthly	Seasonal Index
	2010	2011	2012			
Jan	80	85	105	90	94	0.957
Feb	70	85	85	80	94	0.851
Mar	80	93	82	85	94	0.904
Apr	90	95	115	100	94	1.064
May	113	125	131	123	94	1.309
Jun	110	115	120	115	94	1.223
Jul	100	102	113	105	94	1.117
Aug	88	102	110	100	94	1.064
Sept	85	90	95	90	94	0.957
Oct	77	78	85	80	94	0.851
Nov	75	72	83	80	94	0.851
Dec	82	78	80	80	94	0.851

Seasonal Variations in Data

SEASONAL INDEX EXAMPLE

Month	2010	Demand 2011	2012	Average 2010-2012	Average	Seasonal
Monthly Index						
Jan	8					0.957
Feb	7					0.851
Mar	8					0.904
Apr	9					1.064
May	11					1.309
Jun	11					1.223
Jul	10					1.117
Aug	8					1.064
Sept	8					0.957
Oct	7					0.851
Nov	7					0.851
Dec	82	78	80	80	94	0.851

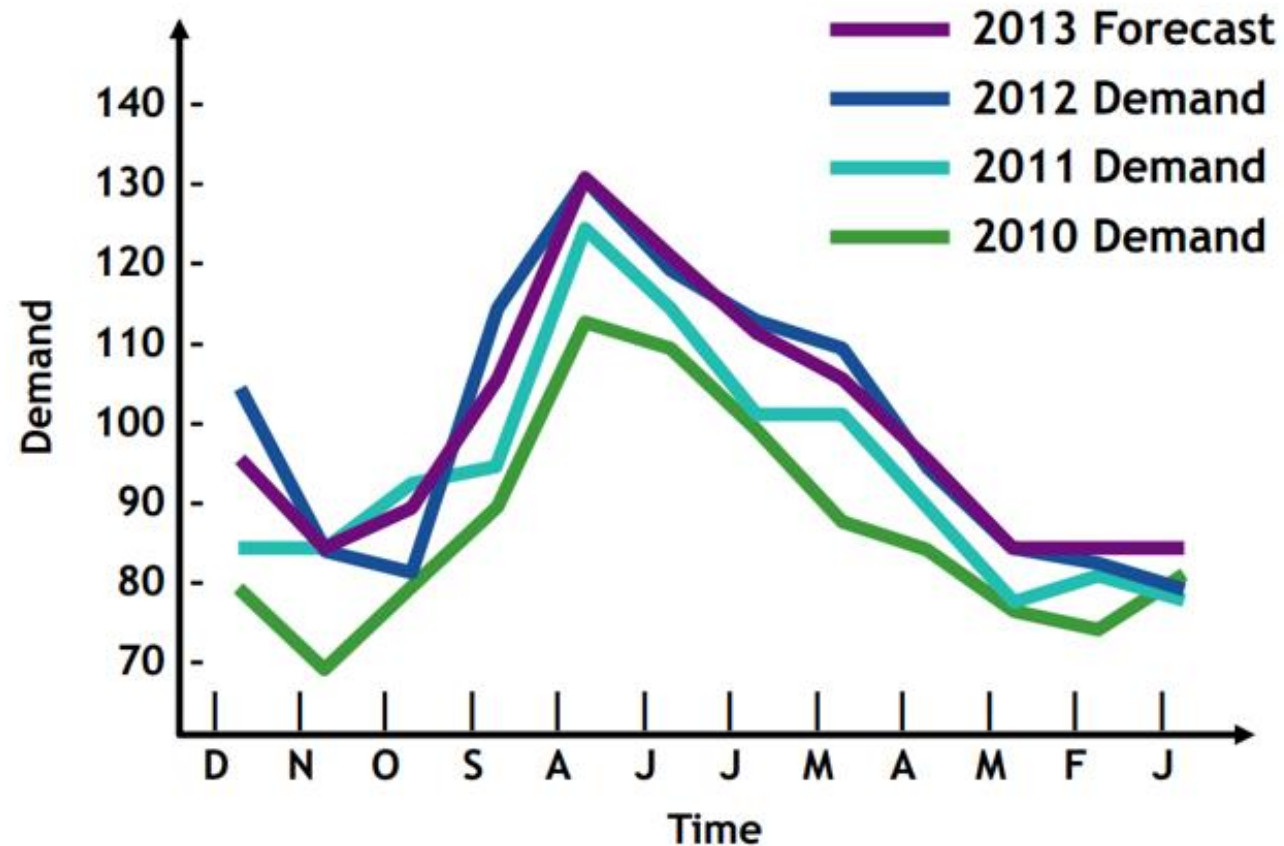
Forecast for 2013

Expected annual demand = 1,200

$$\text{Jan } \frac{1,200}{12} \times .957 = 96$$

$$\text{Feb } \frac{1,200}{12} \times .851 = 85$$

SEASONAL INDEX EXAMPLE



The Confusion Matrix

The Confusion Matrix

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
...	...	...	...	...

- Imagine we have this medical data

The Confusion Matrix

- Suppose we 've some clinical measurement



Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
...	...	...	...	...

a

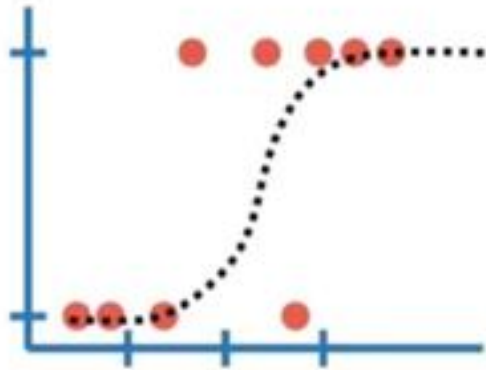
The Confusion Matrix

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
...	...	...	...	...

-and we want to apply a machine learning method to them to predict whether or not someone will develop heart disease.

The Confusion Matrix

To do this, we could use
Logistic Regression...



...or **K-Nearest
Neighbors...**



...or a **Random
Forest...**



...or some other method.
There are tons to choose from.

How do we decide which one
works best with our data?

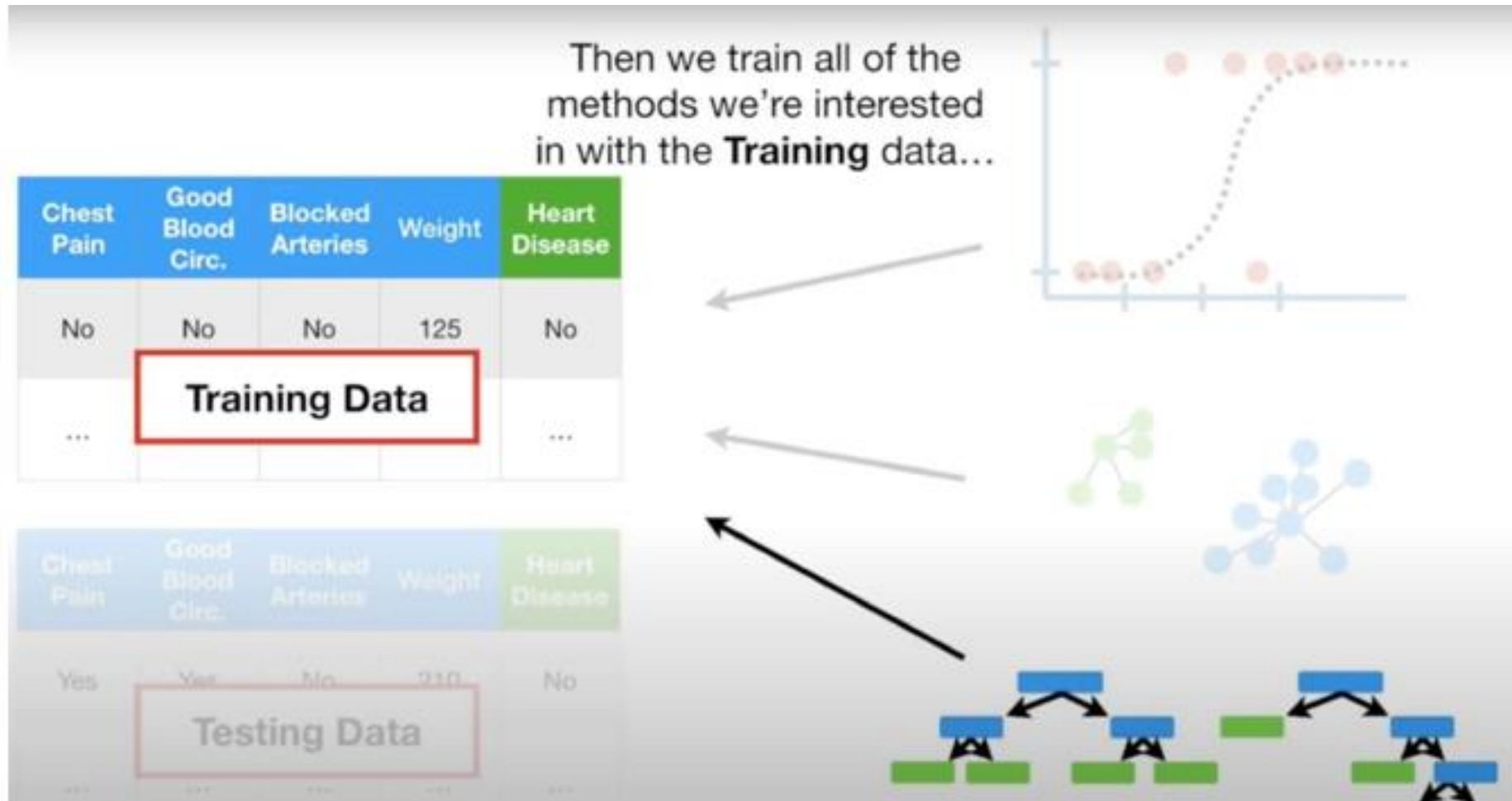
The Confusion Matrix

- We start by dividing the data into Training and Testing sets...

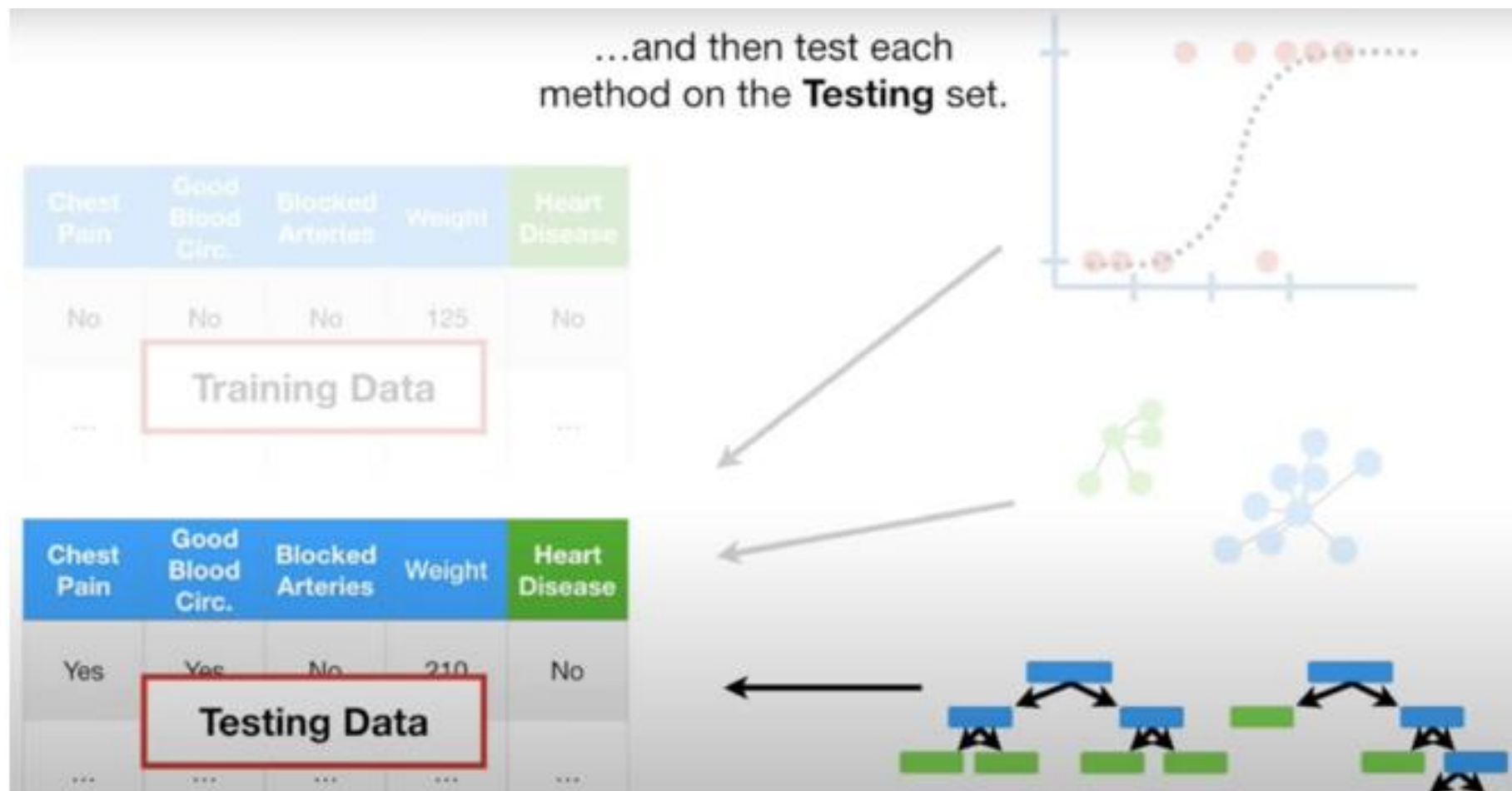
Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
...	Training Data			...

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	Yes	No	210	No
...	...	...	...	...

The Confusion Matrix



The Confusion Matrix



- Now we need to summarize how each method performed on the Testing data

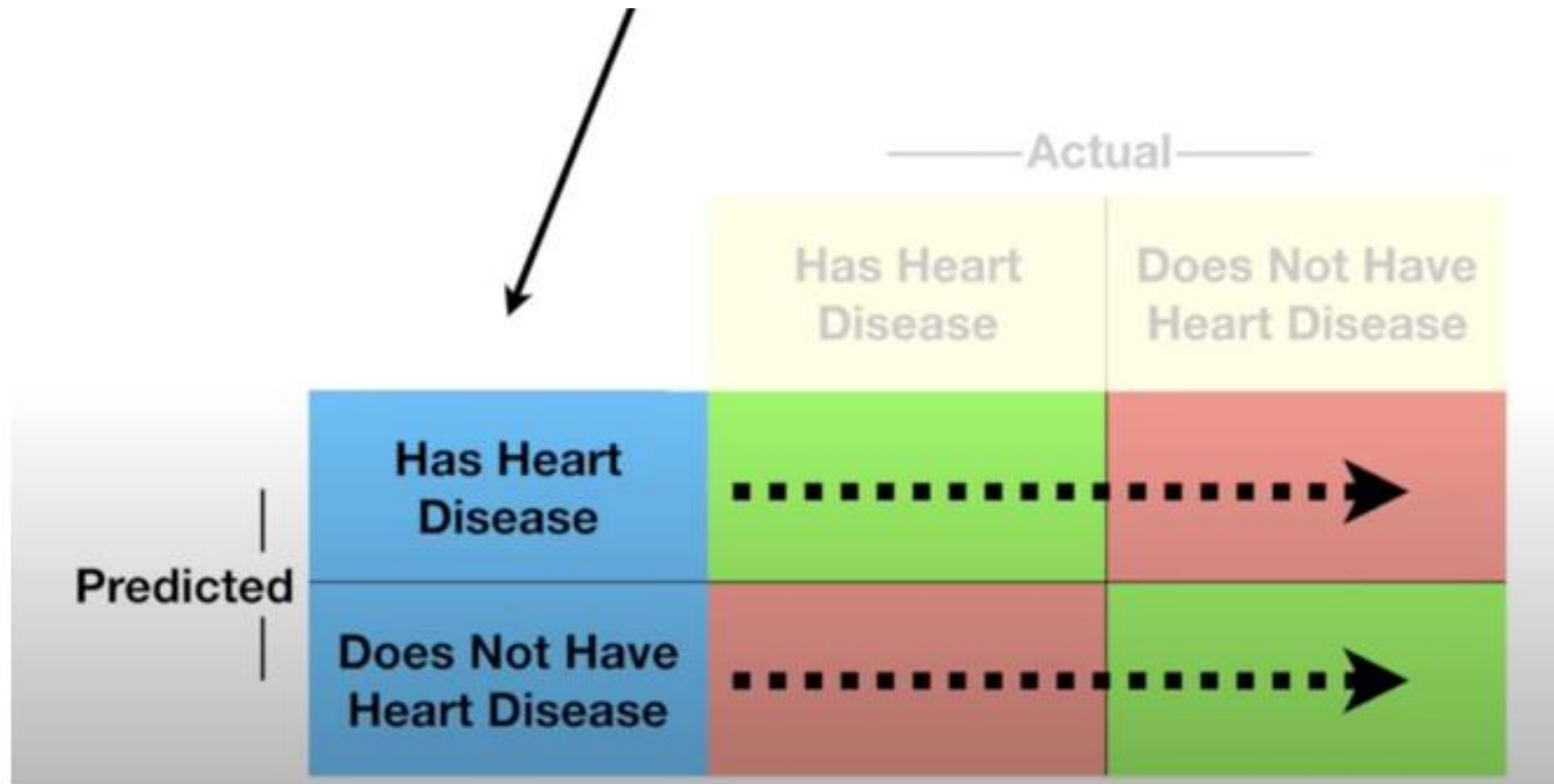
The Confusion Matrix

- One way to do this is by creating a Confusion Matrix for each method.

		Actual	
		Has Heart Disease	Does Not Have Heart Disease
Predicted	Has Heart Disease		
	Does Not Have Heart Disease		

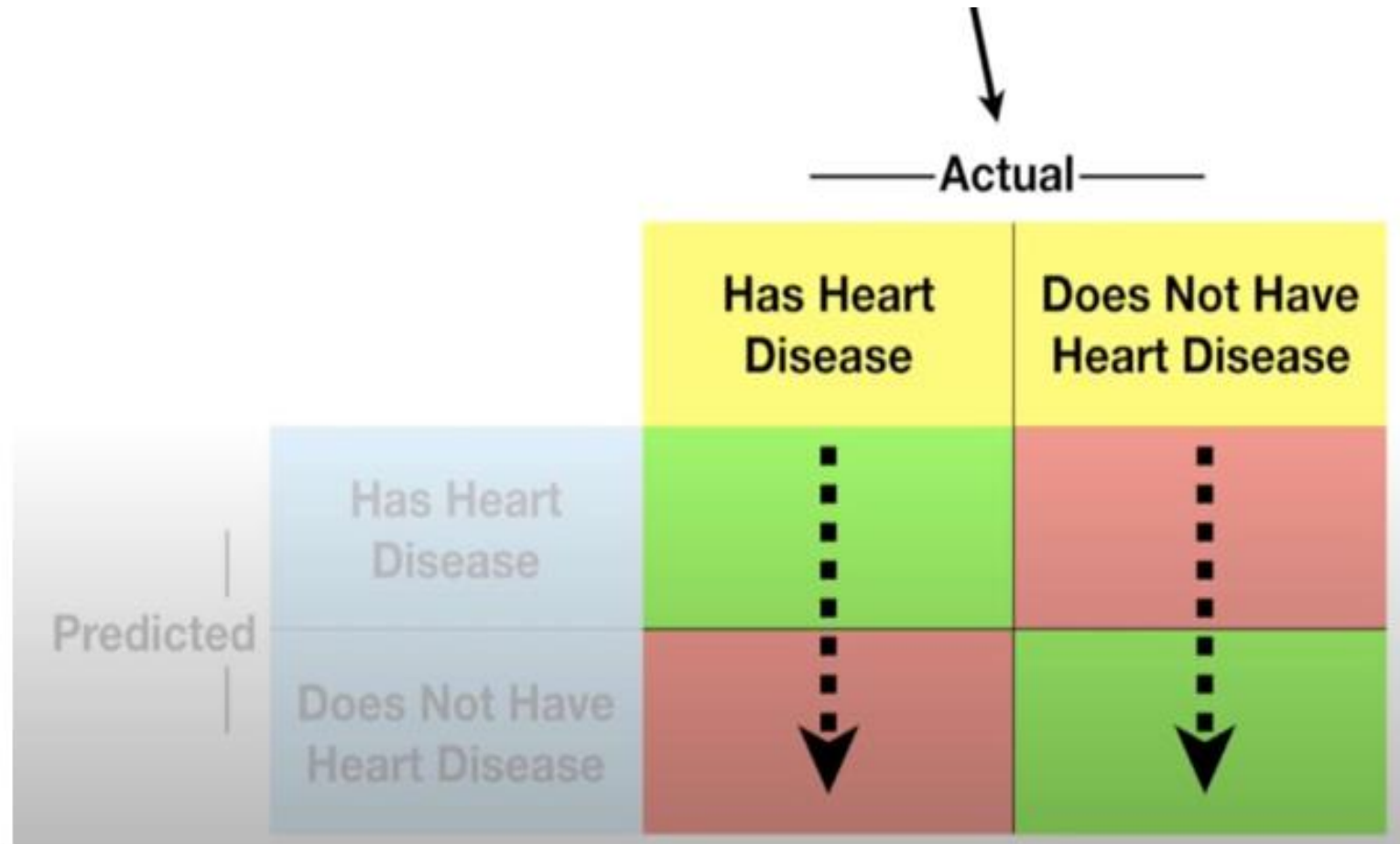
The Confusion Matrix

- The rows in a Confusion Matrix correspond to what the machine learning algorithm predicted...



The Confusion Matrix

- .and the columns correspond to the known truth.



The Confusion Matrix

- Since there are only two categories to choose from
 - : "Has Heart Disease" or
 - "Does Not Have Heart Disease"

		—— Actual ——	
		Has Heart Disease	Does Not Have Heart Disease
Predicted	Has Heart Disease		
	Does Not Have Heart Disease		

The Confusion Matrix

- then the top left corner contains

True Positives.

Actual			
		Has Heart Disease	Does Not Have Heart Disease
Predicted	Has Heart Disease	True Positives	
	Does Not Have Heart Disease		

The Confusion Matrix

- These are the patients that had heart disease that were correctly identified by the algorithm.

		Actual	
		Has Heart Disease	Does Not Have Heart Disease
Predicted	Has Heart Disease	True Positives	
	Does Not Have Heart Disease		

The Confusion Matrix

- The True Negatives are in the bottom right-hand corner.

		Actual	
		Has Heart Disease	Does Not Have Heart Disease
Predicted	Has Heart Disease	True Positives	
	Does Not Have Heart Disease		True Negatives

The Confusion Matrix

- These are the patients that did not have heart disease that were correctly identified by the algorithm.

		Actual	
		Has Heart Disease	Does Not Have Heart Disease
Predicted	Has Heart Disease	True Positives	
	Does Not Have Heart Disease		True Negatives

The Confusion Matrix

- The bottom left-hand corner contains the False Negatives...

		Actual	
		Has Heart Disease	Does Not Have Heart Disease
Predicted	Has Heart Disease	True Positives	
	Does Not Have Heart Disease	False Negatives	True Negatives

The Confusion Matrix

- Lastly, the top right-hand corner contains the False Positives...

		Actual	
		Has Heart Disease	Does Not Have Heart Disease
Predicted	Has Heart Disease	True Positives	False Positives
	Does Not Have Heart Disease	False Negatives	True Negatives

The Confusion Matrix

- False Positives are patients that do not have heart disease, but the algorithm says they do.

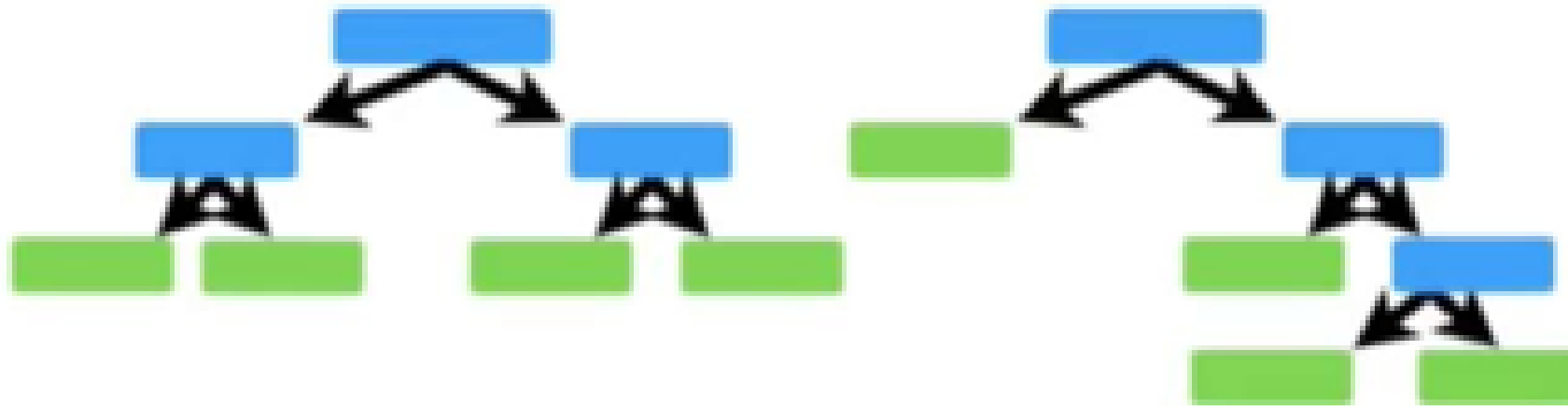
Actual			
Predicted	Has Heart Disease	True Positives	False Positives
	Does Not Have Heart Disease	False Negatives	True Negatives

Terminologies used

- condition positive (P)
the number of real positive cases in the data
- condition negative (N)
the number of real negative cases in the data
- true positive (TP) eqv. with hit
- true negative (TN)
eqv. with correct rejection
- false positive (FP)
eqv. with false alarm, type I error or underestimation
- false negative (FN)
eqv. with miss, type II error or overestimation

The Confusion Matrix

- For example, when we applied the Random Forest to the Testing data..



The Confusion Matrix

There were **142 True Positives**,
patients *with* heart disease that
were correctly classified...



—— Actual ——

		Has Heart Disease	Does Not Have Heart Disease
Predicted	Has Heart Disease	142	
	Does Not Have Heart Disease		

The Confusion Matrix

...and **110 True Negatives**,
patients *without* heart disease
that were correctly classified.

— Actual —

		Actual	
		Has Heart Disease	Does Not Have Heart Disease
Predicted	Has Heart Disease	142	
	Does Not Have Heart Disease		110

The Confusion Matrix

However, the algorithm misclassified **29** patients that *did* have heart disease by saying that they *did not* (**False Negatives**)...

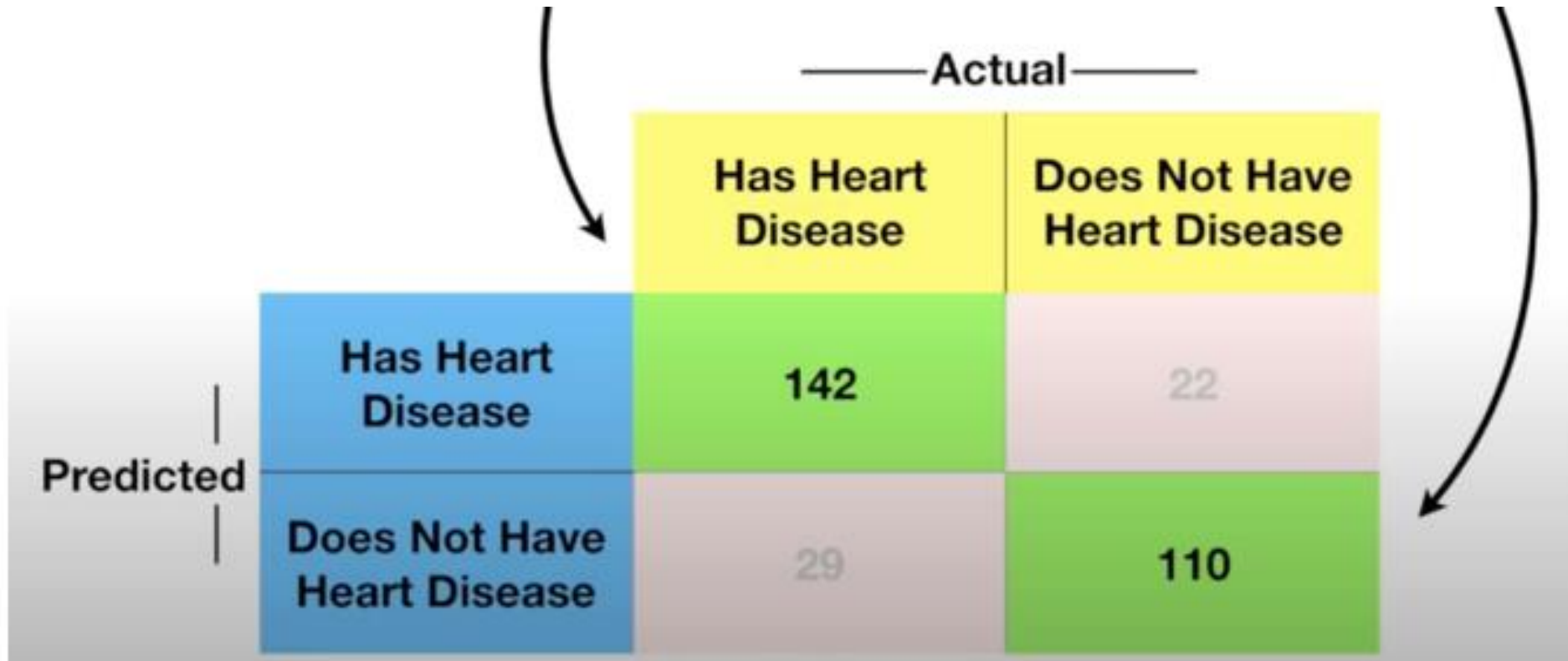


———Actual———

		Has Heart Disease	Does Not Have Heart Disease
Predicted	Has Heart Disease	142	
	Does Not Have Heart Disease	29	110

The Confusion Matrix

- The numbers along the diagonal (the Green Boxes) tell us how many times the samples were correctly classified.



A confusion matrix for heart disease classification. The matrix is a 2x2 grid. The columns are labeled 'Actual' and the rows are labeled 'Predicted'. The diagonal elements, which are green boxes, represent correct classifications: 142 for 'Has Heart Disease' and 110 for 'Does Not Have Heart Disease'. The off-diagonal elements represent misclassifications: 22 for 'Has Heart Disease' predicted as 'Does Not Have Heart Disease' and 29 for 'Does Not Have Heart Disease' predicted as 'Has Heart Disease'.

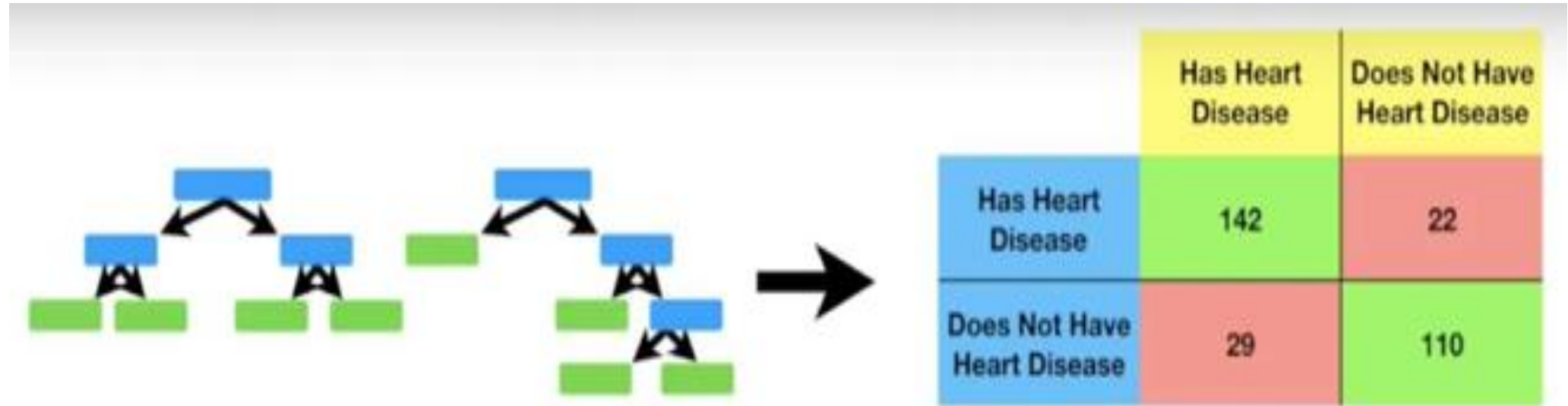
Actual			
Predicted	Has Heart Disease	Does Not Have Heart Disease	
	Has Heart Disease	142	22
	Does Not Have Heart Disease	29	110

The Confusion Matrix

- The numbers not on the diagonal (**the Red Boxes**) are samples the algorithm messed up.

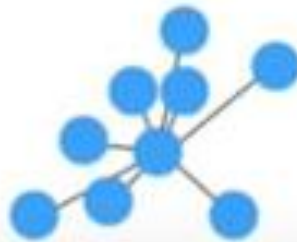
		Actual	
		Has Heart Disease	Does Not Have Heart Disease
Predicted	Has Heart Disease	142	22
	Does Not Have Heart Disease	29	110

The Confusion Matrix



- Now we can compare the Random Forest's Confusion Matrix...

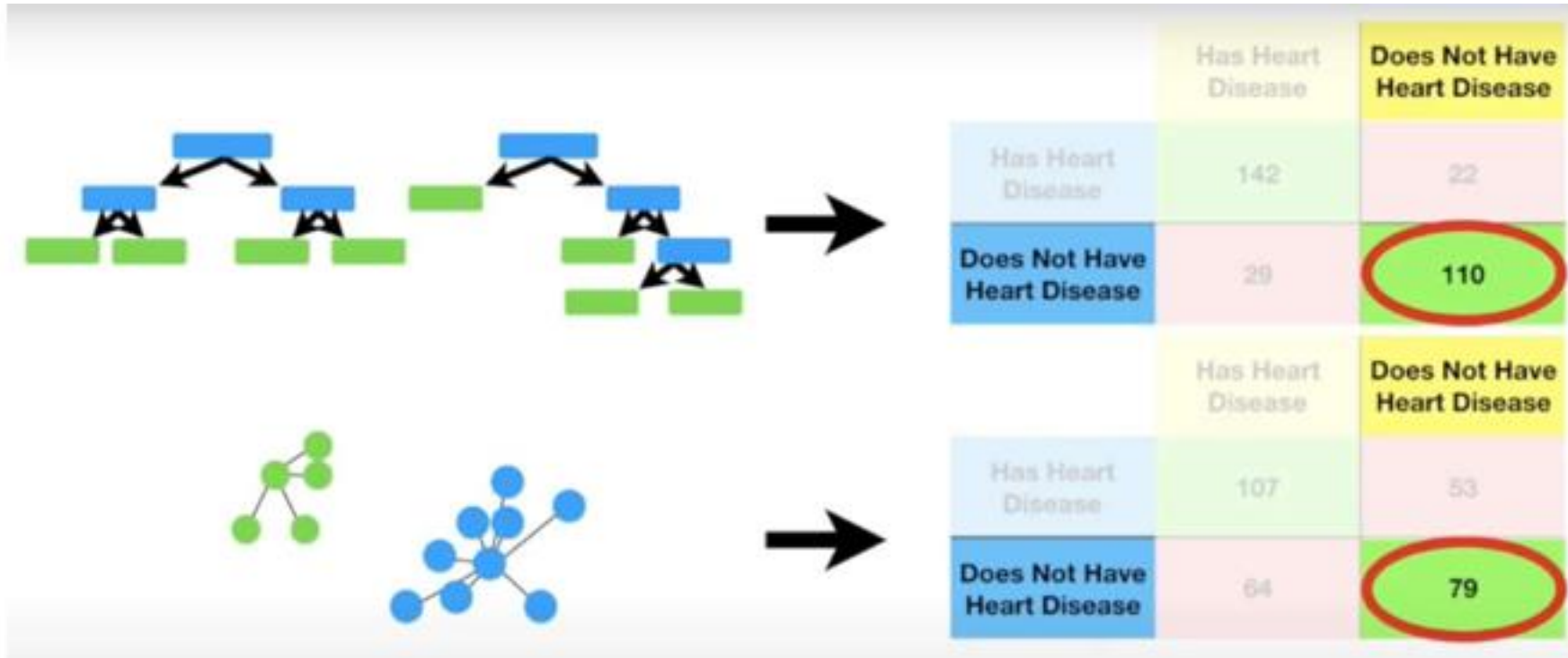
The Confusion Matrix



	Has Heart Disease	Does Not Have Heart Disease
Has Heart Disease	107	53
Does Not Have Heart Disease	64	79

.....to the Confusion Matrix we get when
we use K-Nearest Neighbors.

The Confusion Matrix



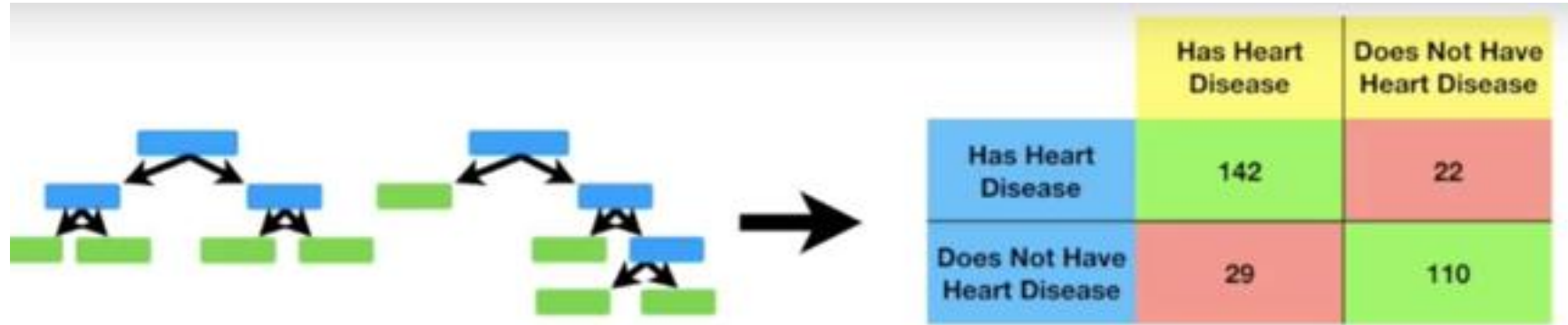
and worse at predicting patients without
Heart Disease (79 vs 110)...

The Confusion Matrix

Lastly, we can apply Logistic Regression to the Testing Dataset and create a Confusion Matrix.



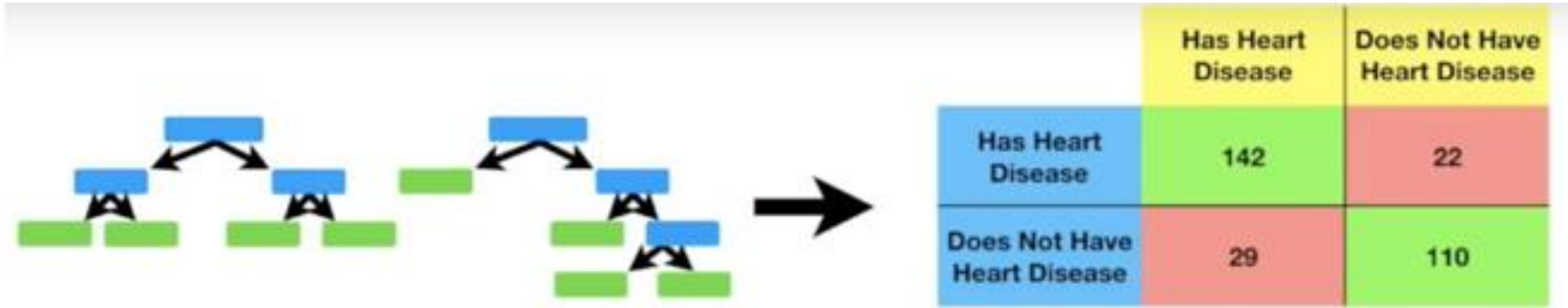
The Confusion Matrix



These two **Confusion Matrices** are very similar and make it hard to choose which machine learning method is a better fit for this data.



The Confusion Matrix



We'll talk about more sophisticated metrics, like **Sensitivity, Specificity, ROC** and **AUC**, that can help us make a decision in the next StatQuests.



Now that we have the basic confusion matrix figured out, let's look at a more complicated one.

The Confusion Matrix

Here's a new data set...



Jurassic Park III	Run for your Wife	Out Kold	Howard the Duck	Favorite Movie
Liked	Didn't Like	Liked	Liked	Troll 2
Didn't Like	Didn't Like	Liked	Didn't Like	Gore Police
Didn't Like	Liked	Liked	Liked	Cool As Ice
...	...	...	...	...

The Confusion Matrix

Now the question is, based on what people think of these movies...



Jurassic Park III	Run for your Wife	Out Kold	Howard the Duck	Favorite Movie
Liked	Didn't Like	Liked	Liked	Troll 2
Didn't Like	Didn't Like	Liked	Didn't Like	Gore Police
Didn't Like	Liked	Liked	Liked	Cool As Ice
...	...	...	...	...

The Confusion Matrix

Jurassic Park III	Run for your Wife	Out Kold	Howard the Duck	Favorite Movie
Liked	Didn't Like	Liked	Liked	Troll 2
Didn't Like	Didn't Like	Liked	Didn't Like	Gore Police
Didn't Like	Liked	Liked	Liked	Cool As Ice
...	...	...	...	...

...can we use a machine learning method to predict their favorite movie.

The Confusion Matrix

Jurassic Park III	Run for your Wife	Out Kold	Howard the Duck	Favorite Movie
Liked	Didn't Like	Liked	Liked	Troll 2
Didn't Like	Didn't Like	Liked	Didn't Like	Gore Police
Didn't Like	Liked	Liked	Liked	Cool As Ice
...	...	...	...	...

If the only options for favorite movie were Troll 2, Gore Police or CoolAs Ice

The Confusion Matrix

...then the confusion matrix would have 3 rows and 3 columns.

		Actual		
		Troll 2	Gore Police	Cool as Ice
Predicted	Troll 2	12	102	93
	Gore Police	112	23	77
	Cool as Ice	83	92	17

The Confusion Matrix

But just like before, the diagonal (the **Green Boxes**) are where the machine learning algorithm did the right thing...

		Actual		
		Troll 2	Gore Police	Cool as Ice
Predicted	Troll 2	12	102	93
	Gore Police	112	23	77
	Cool as Ice	83	92	17

The Confusion Matrix

.and everything else is where
the algorithm messed up

		Actual		
		Troll 2	Gore Police	Cool as Ice
Predicted	Troll 2	12	102	93
	Gore Police	112	23	77
	Cool as Ice	83	92	17

Ultimately, the size of the confusion matrix is determined by the number of things we want to predict.

The Confusion Matrix

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
...	...	...	...	...

In the first example, we were only trying to predict two things: if someone had heart disease or if they didn't...

The Confusion Matrix

. ...and that gave us a confusion matrix with 2 rows and 2 columns.

		Actual	
		Has Heart Disease	Does Not Have Heart Disease
Predicted	Has Heart Disease	142	22
	Does Not Have Heart Disease	29	110

The Confusion Matrix

Jurassic Park III	Run for your Wife	Out Kold	Howard the Duck	Favorite Movie
Liked	Didn't Like	Liked	Liked	Troll 2
Didn't Like	Didn't Like	Liked	Didn't Like	Gore Police
Didn't Like	Liked	Liked	Liked	Cool As Ice
...	...	...	...	...

In the second example, we had three things to choose from.

The Confusion Matrix

.and a confusion matrix with 3 rows and 3 columns.

		Actual		
		Troll 2	Gore Police	Cool as Ice
Predicted	Troll 2	12	102	93
	Gore Police	112	23	77
	Cool as Ice	83	92	17

The Confusion Matrix

If we had 4 things to choose from, we get a confusion matrix with 4 rows and 4 columns...

	Thing 1	Thing 2	Thing 3	Thing 4
Thing 1				
Thing 2				
Thing 3				
Thing 4				

The Confusion Matrix

In summary, a Confusion Matrix tells you what your machine learning algorithm did right...

		Actual	
		Has Heart Disease	Does Not Have Heart Disease
Predicted	Has Heart Disease	True Positives	
	Does Not Have Heart Disease		True Negatives

The Confusion Matrix

In summary, a Confusion Matrix tells you what your machine learning algorithm did right... And what it did wrong.

		Actual	
		Has Heart Disease	Does Not Have Heart Disease
Predicted	Has Heart Disease	True Positives	False Positives
	Does Not Have Heart Disease	False Negatives	True Negatives

Summarized Confusion Matrix

- It is a special kind of contingency table, with two dimensions ("actual" and "predicted"), and identical sets of "classes" in both dimensions (each combination of dimension and class is a variable in the contingency table).

Summarized Confusion Matrix

- Confusion matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning.
- Each row of the matrix represents the instances in an actual class while each column represents the instances in a predicted class, or vice versa - both variants are found in the literature.
- The name stems from the fact that it makes it easy to see whether the system is confusing two classes (i.e. commonly mislabeling one as another).

Summarized Confusion Matrix

- Confusion matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning.
- Each row of the matrix represents the instances in an actual class while each column represents the instances in a predicted class, or vice versa – both variants are found in the literature.
- The name stems from the fact that it makes it easy to see whether the system is confusing two classes (i.e. commonly mislabeling one as another).

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Confusion Matrix Example 2

- Given a sample of 12 pictures, 8 of cats and 4 of dogs, where cats belong to class 1 and dogs belong to class 0,
actual = [1,1,1,1,1,1,1,1,0,0,0,0],
- assume that a classifier that distinguishes between cats and dogs is trained, and we take the 12 pictures and run them through the classifier. The classifier makes 9 accurate predictions and misses 3: 2 cats wrongly predicted as dogs (first 2 predictions) and 1 dog wrongly predicted as a cat (last prediction).
prediction = [0,0,1,1,1,1,1,1,0,0,0,1]
- With these two labeled sets (actual and predictions), we can create a confusion matrix that will summarize the results of testing the classifier:

Confusion Matrix Example 2

In this confusion matrix, of the 8 cat pictures, the system judged that 2 were dogs, and of the 4 dog pictures, it predicted that 1 were cats.

All correct predictions are located in the diagonal of the table (highlighted in bold), so it is easy to visually inspect the table for prediction errors, as values outside the diagonal will represent them.

Predicted class Actual class	Cat	Dog
	Cat	Dog
Cat	6	2
Dog	1	3

Confusion Matrix Example 2

Actual class \ Predicted class	P	N
P	TP	FN
N	FP	TN

Actual class \ Predicted class	Cat	Non-cat
Cat	6 true positives	2 false negatives
Non-cat	1 false positive	3 true negatives

Week

11

Performance metrics

1. Metrics
2. Precision , Accuracy , F1 Score and recall
3. Worked example

1. **Metrics**
2. Precision , Accuracy , F1 Score and recall
3. Worked example

It is extremely important to use **quantitative metrics** for evaluating a machine learning model

- Until now, we relied on the **cost function value** for regression and classification
- Other metrics can be used to **better evaluate** and understand the model
- For classification
 - ✓ Accuracy/Precision/Recall/F1-score, ROC curves,...
- For regression
 - ✓ Normalized RMSE, Normalized Mean Absolute Error (NMAE),...

- Accuracy is also used as a statistical measure of how well a binary classification test correctly identifies or excludes a condition.
- That is, the accuracy is the proportion of correct predictions (both true positives and true negatives) among the total number of cases examined

$$ACC = \frac{TP + TN}{P + N} = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy is a measure of **how close** a given set of guessing from our model are closed to their true value.

$$\text{Accuracy} = \frac{\text{\# Correct classifications}}{\text{\# All classifications}}$$

- If a classifier make 10 predictions and 9 of them are correct, the accuracy is 90%.
- Accuracy is a measure of **how well** a binary classifier correctly identifies or excludes a condition.
- It's the **proportion of correct predictions among the total number of cases examined.**

1. Metrics
2. **Precision , Accuracy , F1 Score and recall**
3. Worked example

- Accuracy is also used as a statistical measure of how well a binary classification test correctly identifies or excludes a condition.
- That is, the accuracy is the proportion of correct predictions (both true positives and true negatives) among the total number of cases examined

$$ACC = \frac{TP + TN}{P + N} = \frac{TP + TN}{TP + TN + FP + FN}$$

The positive predictive value (PPV), also called precision, is defined as,

$$\text{PPV} = \frac{\text{Number of true positives}}{\text{Number of true positives} + \text{Number of false positives}} = \frac{\text{Number of true positives}}{\text{Number of positive calls}}$$

The negative predictive value is defined as:

$$\text{NPV} = \frac{\text{Number of true negatives}}{\text{Number of true negatives} + \text{Number of false negatives}} = \frac{\text{Number of true negatives}}{\text{Number of negative calls}}$$

- In statistical analysis of binary classification, the F-score or F-measure is a measure of a test's accuracy.
- It is calculated from the precision and recall of the test, where the precision is the number of true positive results divided by the number of all positive results, including those not identified correctly, and the recall is the number of true positive results divided by the number of all samples that should have been identified as positive.

$$F_1 = \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{\text{tp}}{\text{tp} + \frac{1}{2}(\text{fp} + \text{fn})}.$$

Suppose that $y = 1$ in presence of a **rare class** that we want to detect

Precision (*How much we are precise in the detection*)

*Of all patients where we classified $y = 1$,
what fraction actually has the disease?*

$$\text{Precision} = \frac{\text{True Positive}}{\text{\# Estimated Positive}} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

Recall (*How much we are good at detecting*)

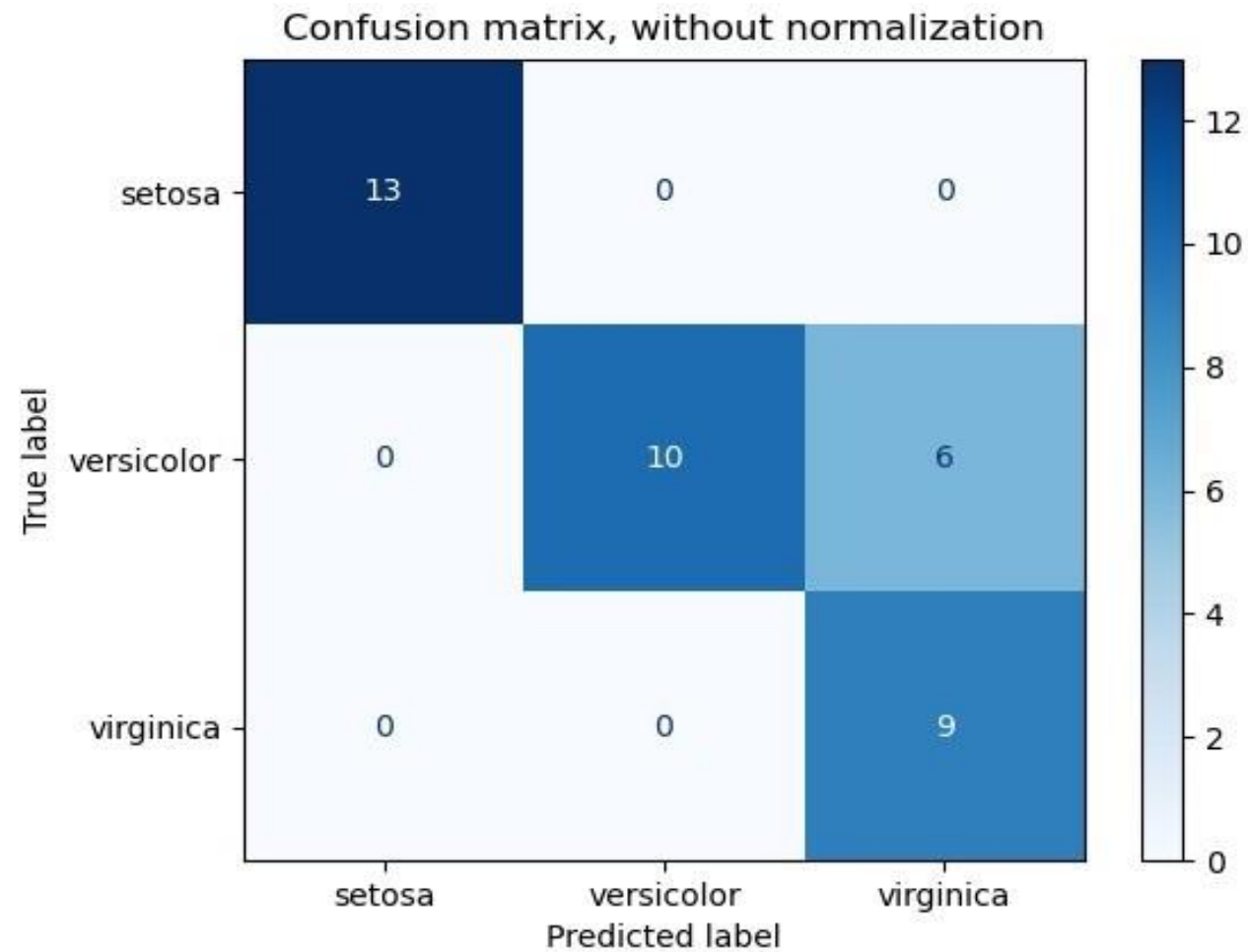
*Of all patients that actually have the disease, what
fraction did we correctly detect as having the disease?*

$$\text{Recall} = \frac{\text{True Positive}}{\text{\# Actual Positive}} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

Confusion matrix

		Actual class	
		1 (p)	0 (n)
Estiamted class	1 (Y)	True positiv e (TP)	False positive (FP)
	0 (N)	False negativ e (FN)	True negativ e (TN)

Confusion Matrix for multiple classes



Different metrics can be computed from the confusion matrix, depending on the class of interest (https://en.wikipedia.org/wiki/Precision_and_recall)

		True condition			
Total population		Condition positive	Condition negative	Prevalence = $\frac{\sum \text{Condition positive}}{\sum \text{Total population}}$	Accuracy (ACC) = $\frac{\sum \text{True positive} + \sum \text{True negative}}{\sum \text{Total population}}$
Predicted condition	Predicted condition positive	True positive , Power	False positive , Type I error	Positive predictive value (PPV), Precision = $\frac{\sum \text{True positive}}{\sum \text{Predicted condition positive}}$	False discovery rate (FDR) = $\frac{\sum \text{False positive}}{\sum \text{Predicted condition positive}}$
	Predicted condition negative	False negative , Type II error	True negative	False omission rate (FOR) = $\frac{\sum \text{False negative}}{\sum \text{Predicted condition negative}}$	Negative predictive value (NPV) = $\frac{\sum \text{True negative}}{\sum \text{Predicted condition negative}}$
		True positive rate (TPR), Recall, Sensitivity, probability of detection = $\frac{\sum \text{True positive}}{\sum \text{Condition positive}}$	False positive rate (FPR), Fall-out, probability of false alarm = $\frac{\sum \text{False positive}}{\sum \text{Condition negative}}$	Positive likelihood ratio (LR+) = $\frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) = $\frac{\text{LR+}}{\text{LR-}}$ $F_1 \text{ score} = \frac{1}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}}$
		False negative rate (FNR), Miss rate = $\frac{\sum \text{False negative}}{\sum \text{Condition positive}}$	Specificity (SPC), Selectivity, True negative rate (TNR) = $\frac{\sum \text{True negative}}{\sum \text{Condition negative}}$	Negative likelihood ratio (LR-) = $\frac{\text{FNR}}{\text{TNR}}$	

1. Metrics
2. Precision , Accuracy , F1 Score and recall
3. **Worked example**

Example 1

suppose the following data set

Feat-1	Feat-2	Feat-3	...	Feat-n	Class
80	Low	M	...	...	Yes
90	Normal	M	...	...	No
88	High	F	...	...	Yes
110	Normal	F	...	...	Yes
95	High	M	...	...	No

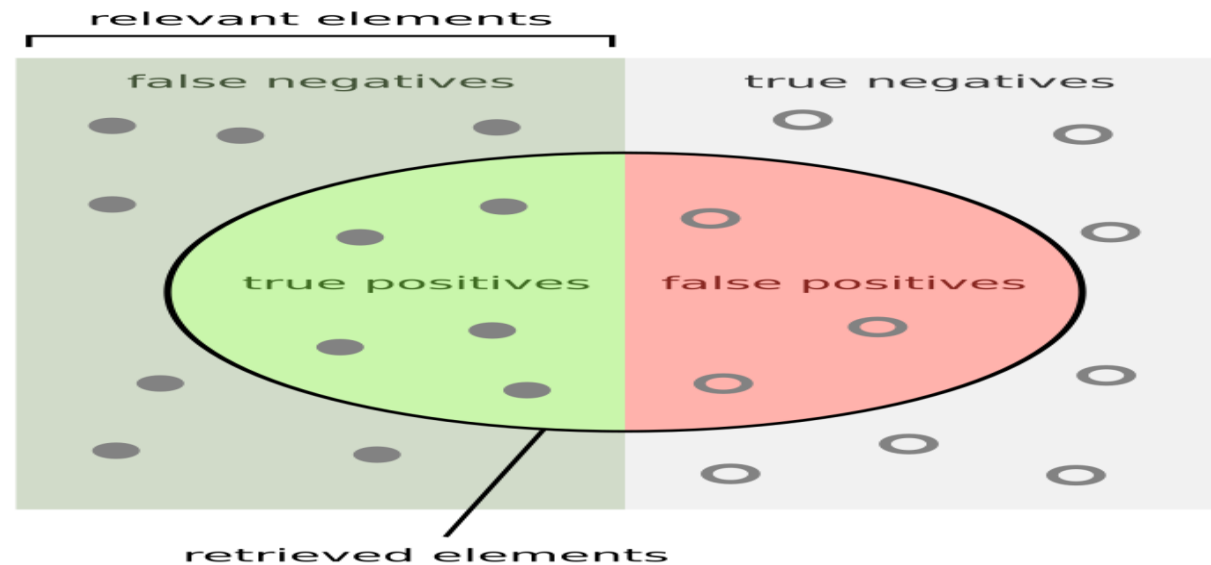
Example 1

suppose the following data set

Feat-1	Feat-2	Feat-3	...	Feat-n	Class
80	Low	M	...	...	Yes
90	Normal	M	...	...	No
88	High	F	...	...	Yes
110	Normal	F	...	...	Yes
95	High	M	...	...	No

Predicted values \ Real Data	She/he is Sick	Not sick
Predicted as sick	TP	FP
Not sick	FN	TN

Construction Confusion Matrix



How many retrieved items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

How many relevant items are retrieved?

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

Suppose the following dataset

```

100 Inst
  40 P Gold
  60 N Gold

Pred1 100P 0N
P= 40/100
R= 40/40
Acc= 40+0/100
////////////////////////////////////

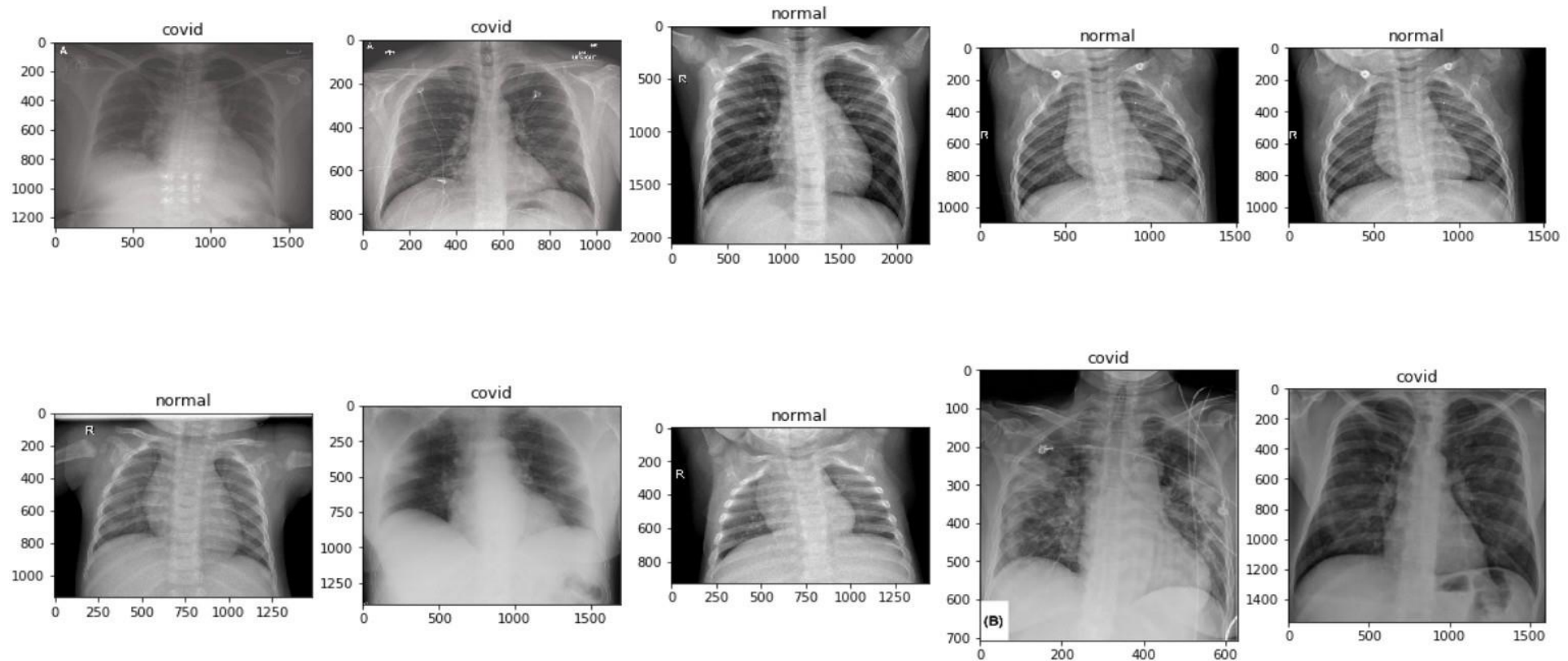
Pred2 0P 100N
P= 0/0
R= 0/40
Acc=0
////////////////////////////////////

Pred3 45P(I37/F8) 55N(I52/F3)
P= 37/45=0.82
R= 37/40=92.5
Acc=(37+52)/100

```

Example 2 : Pneumonia detection

Suppose to have at disposal X-ray images of lungs: **Healthy** people - **Covid-19 disease** patients



This example is **ONLY** for educational purposes

- The COVID-19 X-ray image is curated by Dr. Joseph Cohen, a postdoctoral fellow at the University of Montreal, see <https://josephpcohen.com/w/public-covid19-dataset/>
- The previous data contain only X-ray images of people with a disease. To collect images of healthy people, we can download another X-ray dataset on the platform Kaggle <https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia>
- The analysis is inspired from a tutorial by Adrian Rosebrock: <https://www.pyimagesearch.com/2020/03/16/detecting-covid-19-in-x-ray-images-with-keras-tensorflow-and-deep-learning/>

This example is **ONLY** for educational purposes

We want to use a classifier to perform classification:

- Healthy patients: class 0
- Patients with a disease: class 1

The input data are directly the X-ray images

For these computer vision tasks, the state of the art algorithm are the Convolutional Neural Networks:

- we can use them to classify the images into healthy and disease

This example is **ONLY** for educational purposes

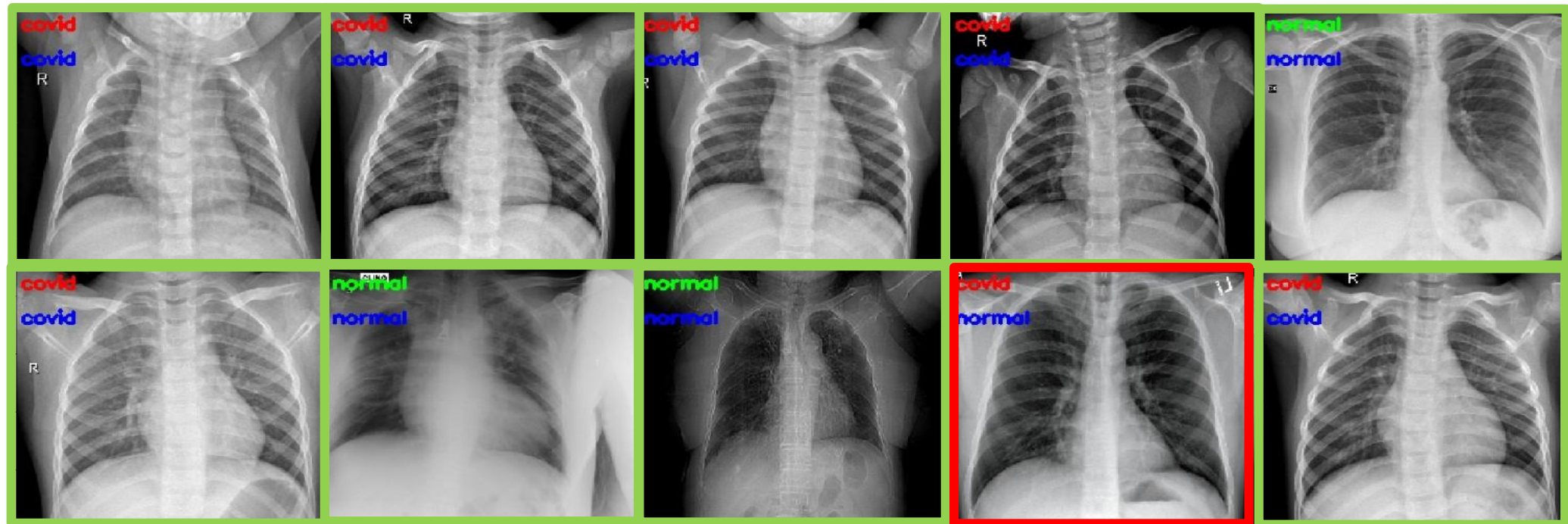
Example 2 : Pneumonia detection

True label

Estimated covid label

Estimated healthy label

Suppose to have at disposal X-ray images of lungs: **Healthy** people - **Covid-19 disease** patients



This example is **ONLY** for educational purposes

Classification results on test set**Sensitivity** (recall, true positive rate)

$$\frac{\text{True Positive}}{\# \text{ Actual Positive}} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} = 0.92$$

Specificity (true negative rate)

$$\frac{\text{True Negative}}{\# \text{ Actual Negative}} = \frac{\text{True Negative}}{\text{False Positive} + \text{True Negative}} = 1$$

Estimated class

		Actual class	
		1 (p)	0 (n)
Estimated class	1 (Y)	True positive 11	False positive 0
	0 (N)	False negative 1	True negative 11

- **Accuracy:** $\approx 96\%$

This example is **ONLY** for educational purposes

Sensitivity (recall, true positive rate)

$$\frac{\text{True Positive}}{\# \text{ Actual Positive}} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} = 0.92$$

Specificity (true negative rate)

$$\frac{\text{True Negative}}{\# \text{ Actual Negative}} = \frac{\text{True Negative}}{\text{False Positive} + \text{True Negative}} = 1$$

- **Sensitivity**: of patients that *do have* COVID-19 (i.e., *true positives*), we could accurately identify them as “COVID-19 positive” 92% of the time using our model
- **Specificity**: of patients that *do not have* COVID-19 (i.e., *true negatives*), we could accurately identify them as “COVID-19 negative” 100% of the time using our model.

This example is **ONLY** for educational purposes

Balancing sensitivity and specificity is incredibly challenging when it comes to medical applications

The results should **always be validated** with another pool of people

Furthermore, we need to be **concerned of what the model is actually**

learning:

- Does the results align with the medical knowledge?
- Was the dataset well representative of the population or there was selection bias?

Issues Affecting Model Selection

- **Accuracy**
 - classifier accuracy: predicting class label
- **Speed**
 - time to construct the model (training time)
 - time to use the model (classification/prediction time)
- **Robustness**: handling noise and missing values
- **Scalability**: efficiency in disk-resident databases
- **Interpretability**
 - understanding and insight provided by the model
- Other measures, e.g., goodness of rules, such as decision tree size or compactness of classification rules



Thank You