



ANTICIPARE LA CRESCITA CON LE NUOVE COMPETENZE SUI BIG DATA – EDIZIONE 2

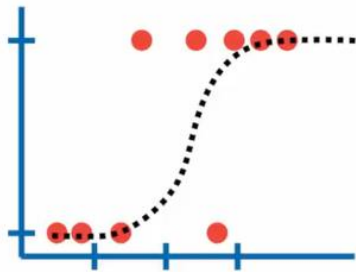
Operazione Rif. PA 2019-11596/RER “Anticipare la crescita con le nuove competenze sui Big Data - Edizione 2”, approvata dalla Regione Emilia-Romagna con DGR n° 789 del 20 maggio 2019 e co-finanziata dal Fondo Sociale Europeo PO 2014-2020

Prog. 4 Ed. 7 Titolo “Tecnologie & Software di Data Science

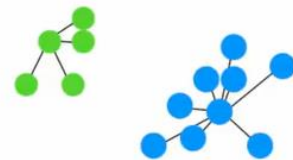


CONFUSION MATRIX

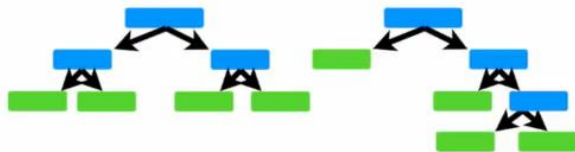
To do this, we could use
Logistic Regression...



...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?

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

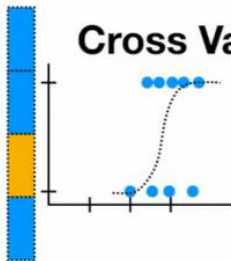
Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
...				

Training Data

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

Testing Data

NOTE: This would be an
excellent opportunity to use
Cross Validation.



Cross Validation....

...it's no
big deal!!!

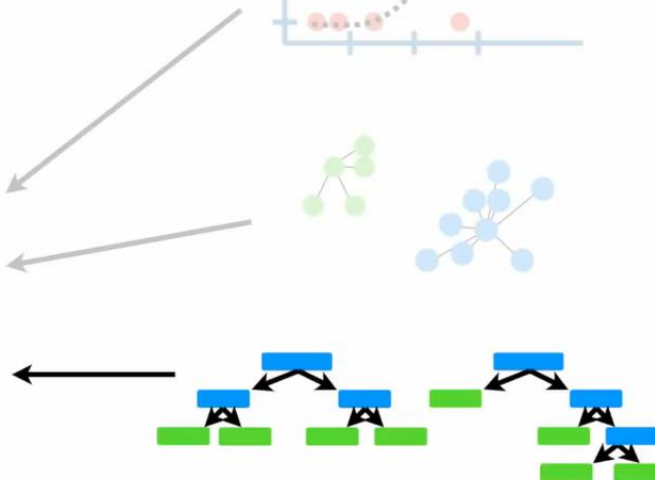
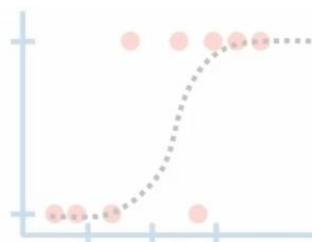
Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
...				

Training Data

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

Testing Data

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



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

Diagram illustrating the structure of a Confusion Matrix. The rows represent the predicted values, and the columns represent the actual values. The matrix is divided into four quadrants based on the predicted and actual outcomes.

		Actual	
		Has Heart Disease	Does Not Have Heart Disease
Predicted	Has Heart Disease	True Positive (Green)	False Positive (Red)
	Does Not Have Heart Disease	False Negative (Red)	True Negative (Green)

Horizontal dashed arrows indicate the flow from the predicted row to the actual column.

...and the columns correspond to the known truth.

Diagram illustrating the structure of a Confusion Matrix. The rows represent the predicted values, and the columns represent the actual values. The matrix is divided into four quadrants based on the predicted and actual outcomes.

		Actual	
		Has Heart Disease	Does Not Have Heart Disease
Predicted	Has Heart Disease	True Positive (Green)	False Positive (Red)
	Does Not Have Heart Disease	False Negative (Red)	True Negative (Green)

Vertical dashed arrows indicate the flow from the predicted row to the actual column.

...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 **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

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

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

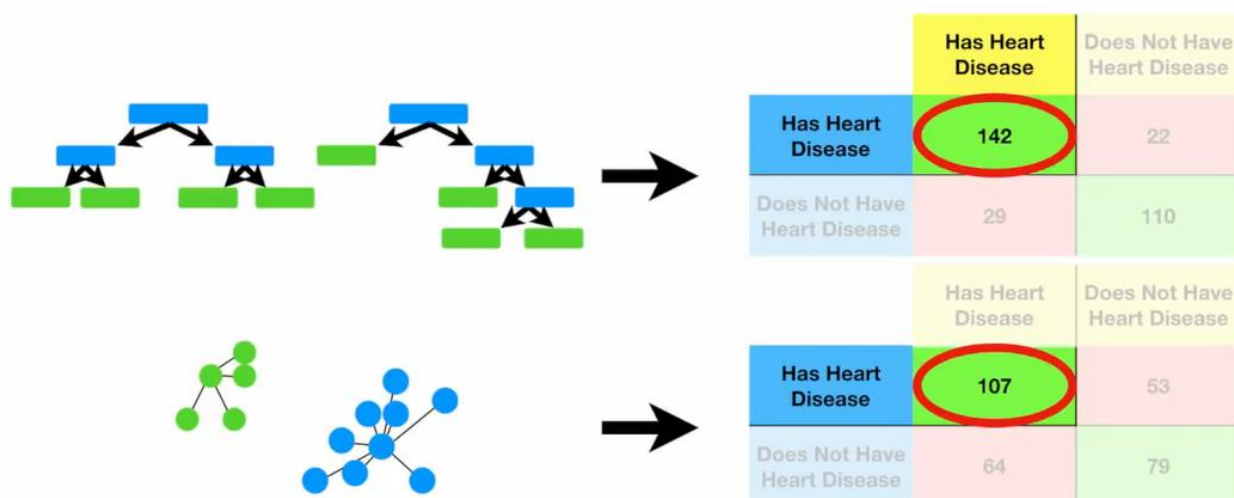
Lastly, the top right-hand corner contains the
False Positives...

		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

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

...and the algorithm misclassified 22 patients that *did not* have heart disease by saying that they *did* (**False Positives**).

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



The figure illustrates two different classification approaches and their corresponding confusion matrices.

Top Classification (Decision Tree):

- The decision tree structure shows internal nodes (blue) and leaf nodes (green).
- The confusion matrix shows the results of classification:

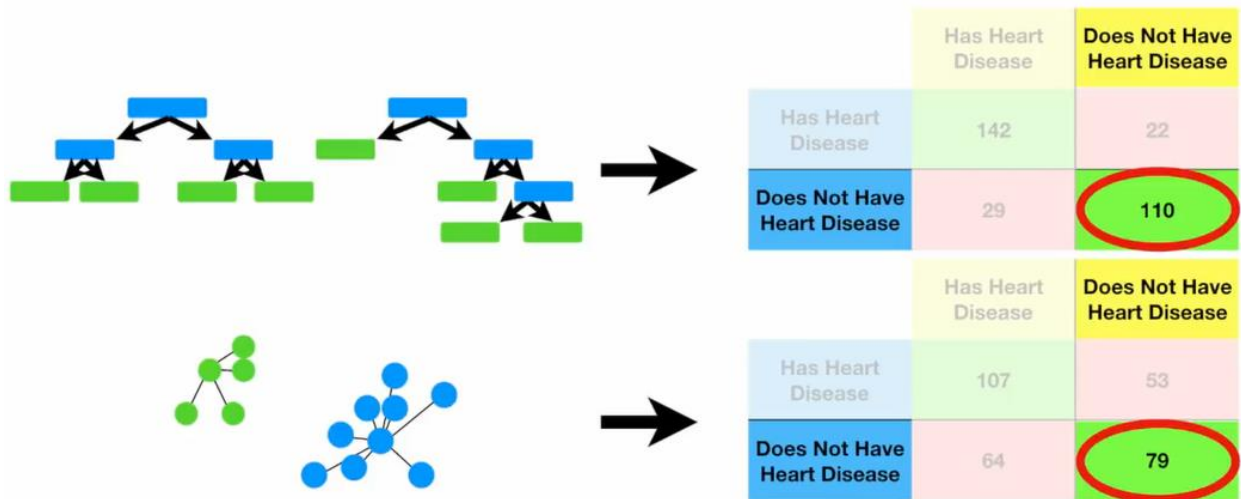
	Has Heart Disease	Does Not Have Heart Disease
Has Heart Disease	142	22
Does Not Have Heart Disease	29	110

Bottom Classification (Graph Clustering):

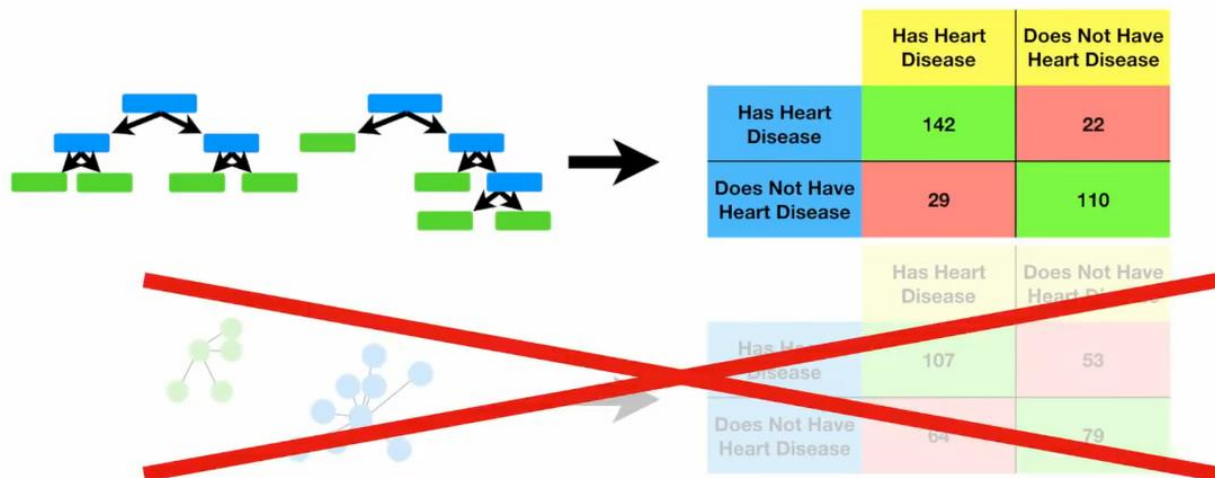
- The graph clustering structure shows two clusters of nodes, one green and one blue.
- The confusion matrix shows the results of classification:

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

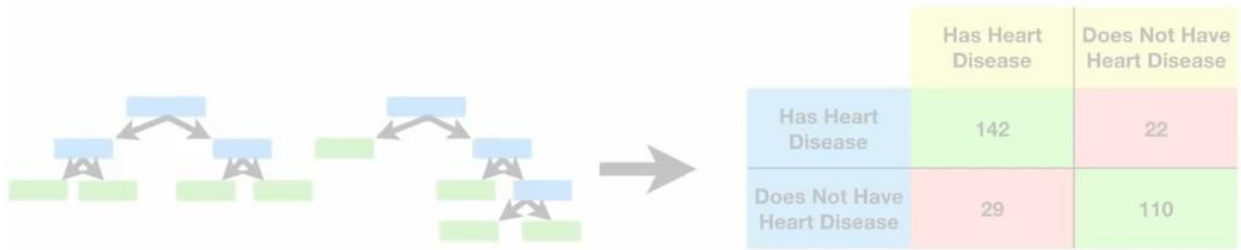
K-Nearest Neighbors was worse than the **Random Forest** at predicting patients *with* Heart Disease (**107** vs **142**)...



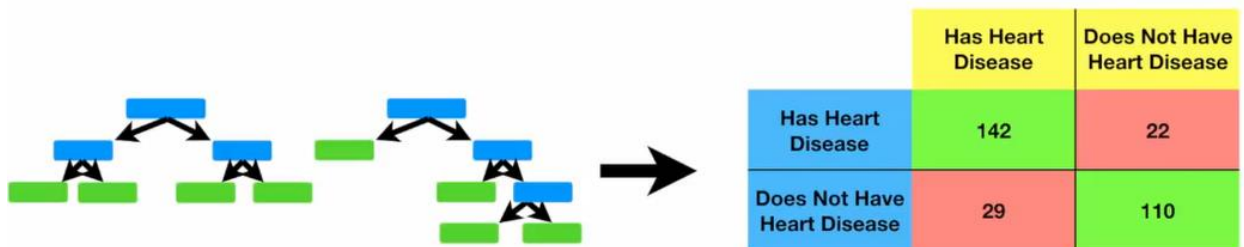
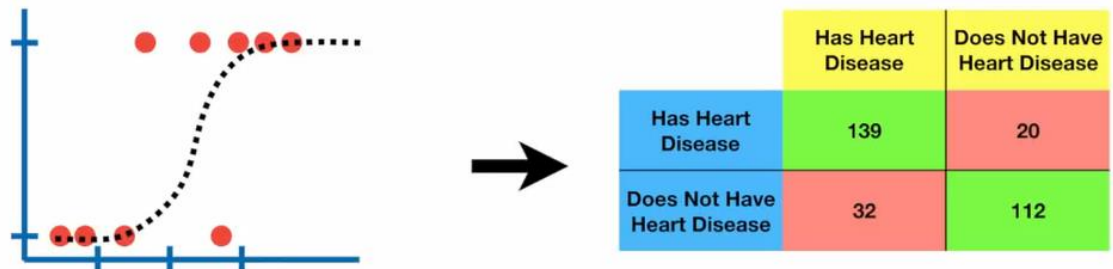
...and worse at predicting patients *without* Heart Disease (79 vs 110)...



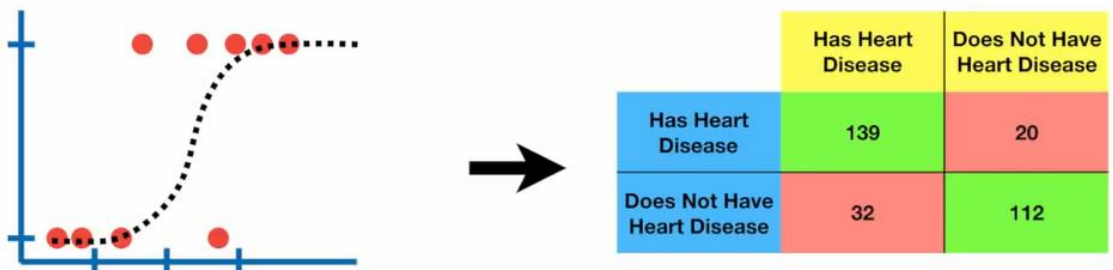
...so if we had to choose between using the **Random Forest** and **K-Nearest Neighbors**, we would choose the **Random Forest**.



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



Sensitivity, Specificity, ROC and AUC

		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

...and if we had 40 things to choose from, we get a confusion matrix with 40 rows and 40 columns.



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

Once we've filled out the **Confusion Matrix**, we can calculate two useful metrics:
Sensitivity and **Specificity**.

		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

In this case, **Sensitivity** tells us what percentage of patients *with* heart disease were correctly identified.

$$\text{Sensitivity} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

		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

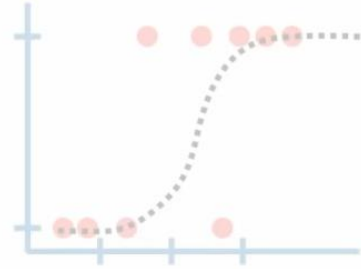
Specificity tells us what percentage of patients *without* heart disease were correctly identified.

$$\text{Specificity} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}}$$

		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

$$\text{Sensitivity} = \frac{139}{139 + 32} = 0.81$$

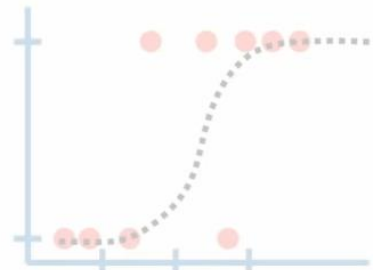
Sensitivity tells us that **81%** of the people *with* Heart Disease were correctly identified by the **Logistic Regression** model.



	Has Heart Disease	Does Not Have Heart Disease
Has Heart Disease	139	20
Does Not Have Heart Disease	32	112

$$\text{Specificity} = \frac{112}{112 + 20} = 0.85$$

Specificity tells us that **85%** of the people *without* Heart Disease were correctly identified by the **Logistic Regression** model.



	Has Heart Disease	Does Not Have Heart Disease
Has Heart Disease	139	20
Does Not Have Heart Disease	32	112

$$\text{Sensitivity} = \frac{142}{142 + 29} = 0.83$$

$$\text{Specificity} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}}$$



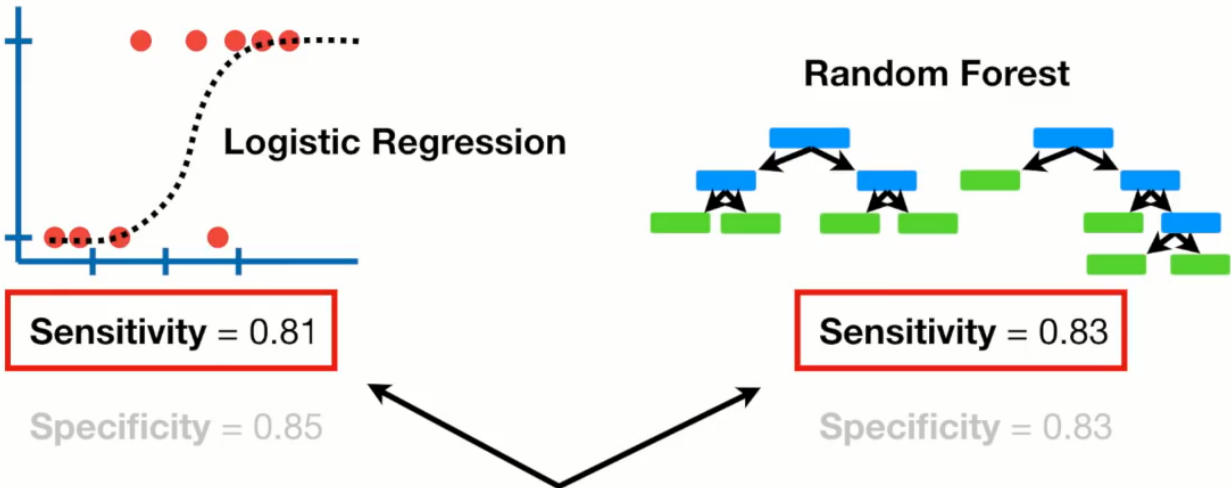
		Actual	
		Has Heart Disease	Does Not Have Heart Disease
Predicted	Has Heart Disease	142	22
	Does Not Have Heart Disease	29	110

$$\text{Sensitivity} = \frac{142}{142 + 29} = 0.83$$

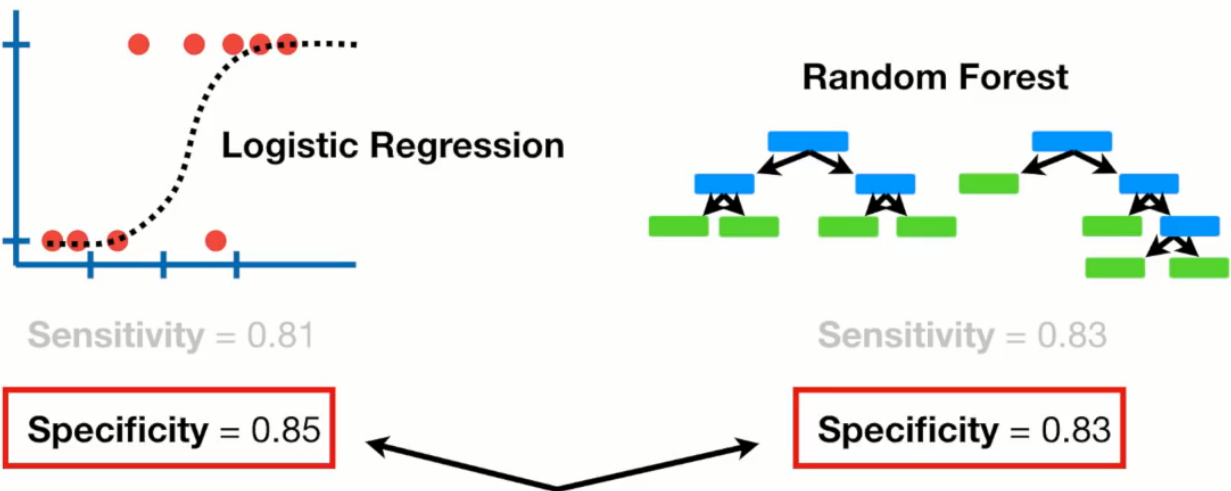
$$\text{Specificity} = \frac{110}{110 + 22} = 0.83$$



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



Sensitivity tells us that the **Random Forest** is slightly better at correctly identifying *positives*, which, in this case, are patients *with* heart disease.



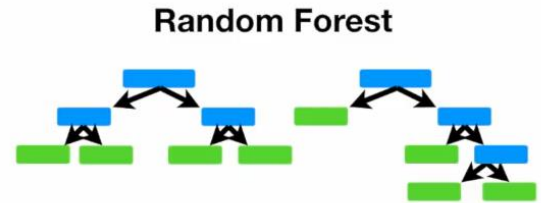
Specificity tells us that **Logistic Regression** is slightly better at correctly identifying *negatives*, which, in this case, are patients *without* heart disease.



Sensitivity = 0.81

Specificity = 0.85

We would choose the **Logistic Regression** model if correctly identifying patients **without** heart disease was more important than correctly identifying patients **with** heart disease.



Sensitivity = 0.83

Specificity = 0.83

Alternatively, we would choose the **Random Forest** model if correctly identifying patients **with** heart disease was more important than correctly identifying patients **without** heart disease.

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

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

$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

$$accuracy = \frac{TP + TN}{TP + FN + TN + FP}$$

$$specificity = \frac{TN}{TN + FP}$$