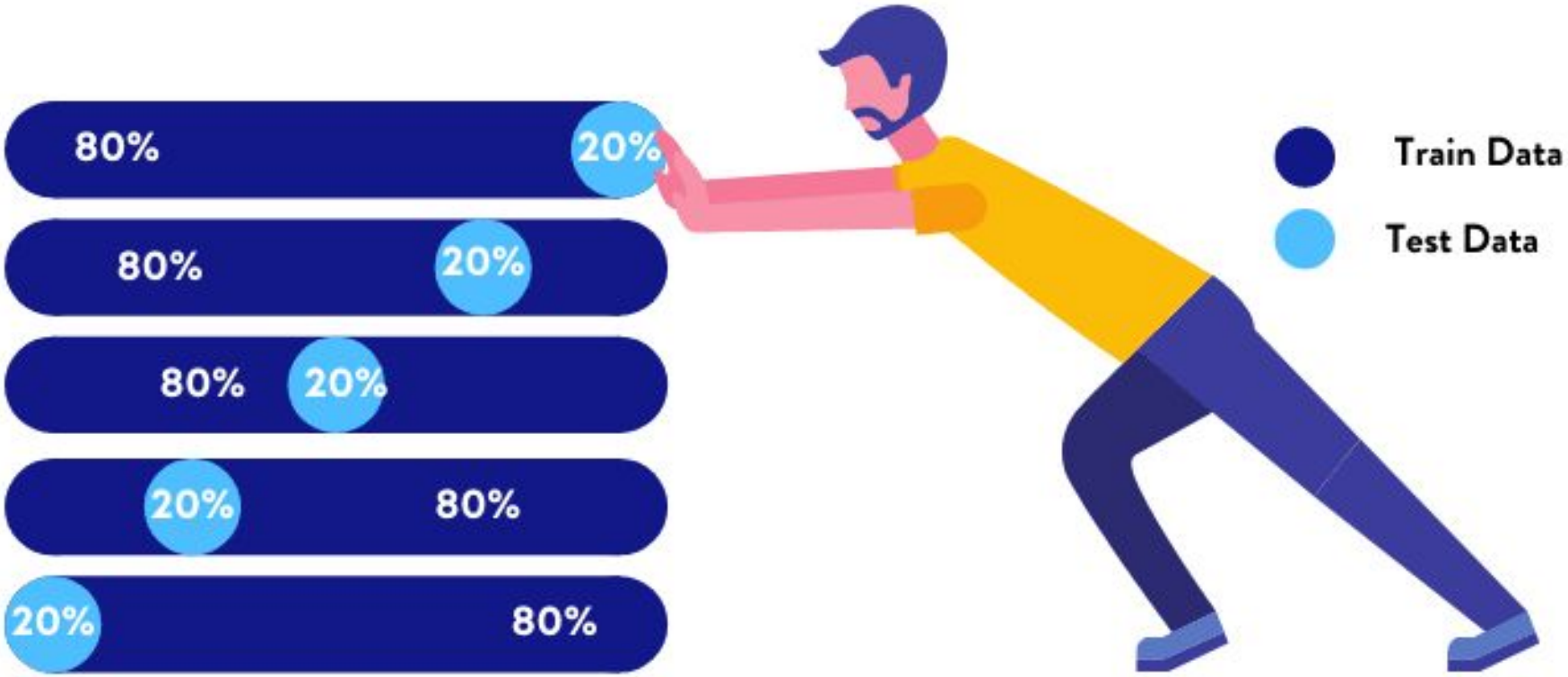
Cross Validation



Suppose we want to use the variables (Chest Pain, Good Blood Circulation,) etc.. to predict if someone has a heart disease

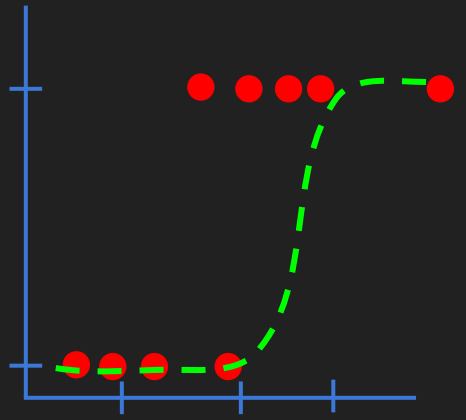
Chest Pain	Good Blood Circulation	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
…	…	…	…	…

We can use this data to predict whether a new patient has heart disease or not

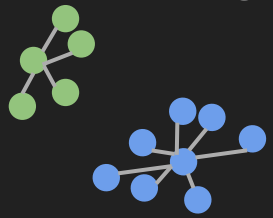
Chest Pain	Good Blood Circulation	Blocked Arteries	Weight	Heart Disease
Yes	No	No	168	**???**

Before we can do this we have to decide which machine learning method would be the best

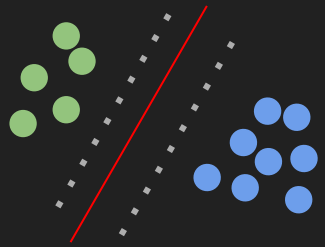
We could use Logistic Regression…



or K-nearest neighbours



or support vector machines(SVM)



and many more machine learning algorithms, how do we decide which one to use?

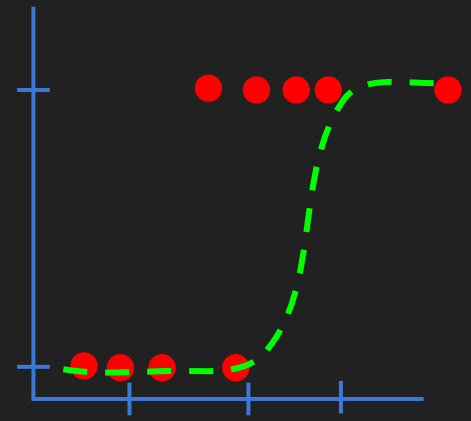
Cross Validation allows us to compare different machine learning methods and get a sense of how well they will work in practice

Let's imagine this blue box represents all the data that we have collected about people with or without heart disease

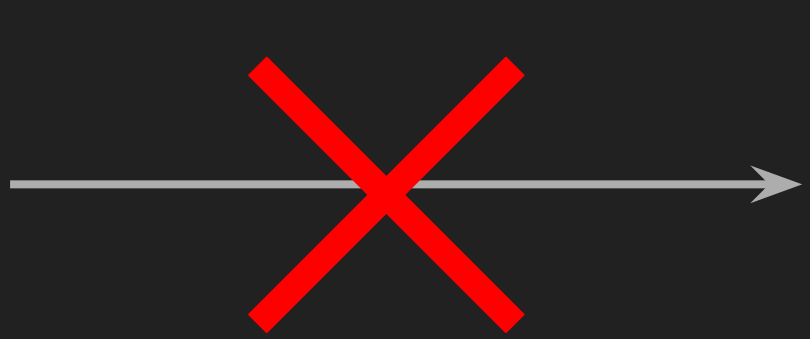
We need to do two things with this data

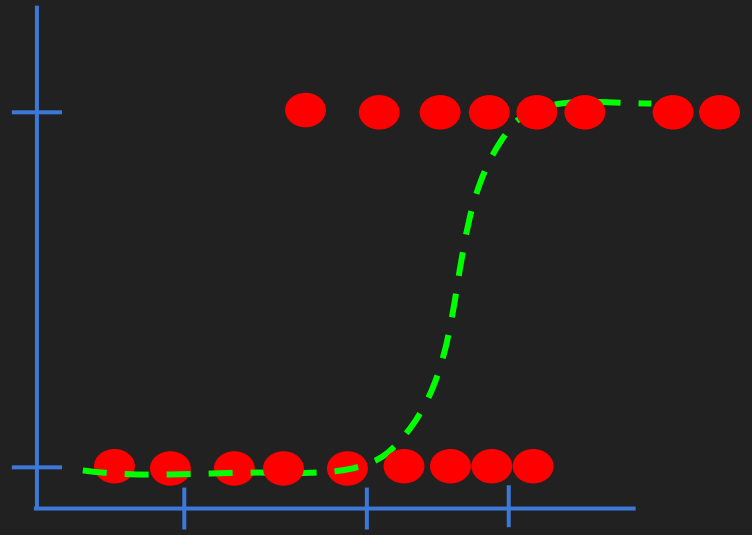
1. Estimate the parameters for the machine learning methods
   - In other words to use Logistic regression we have to use some of the data to estimate the shape of this curve…
   - In machine learning we call this "training the algorithm"
2. Evaluate how well the machine learning methods work
   - In other words we need to find out if this curve will do a good job categorizing new data
   - In ML we call this "testing the algorithm"

To summarize using machine learning we need the data to:
Train the machine learning methods
Test the machine learning methods

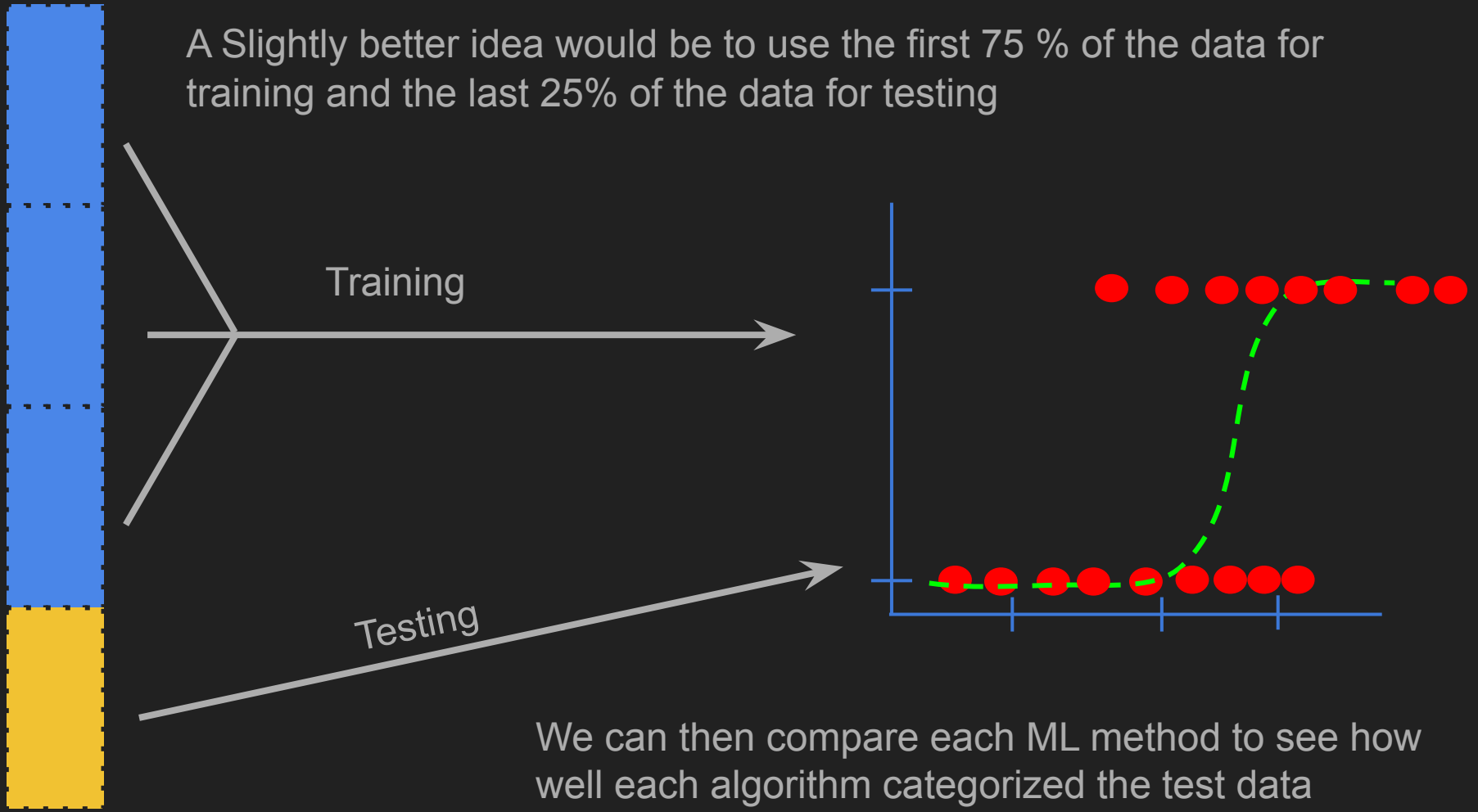


A terrible approach would be to use all of the data to estimate the parameters(i.e train the algorithm) then we would not have any data to test the methods. Reusing the same data to train and test ML model is a bad idea because we need to know how the method will work on the data it was not trained on





A Slightly better idea would be to use the first 75 % of the data for training and the last 25% of the data for testing

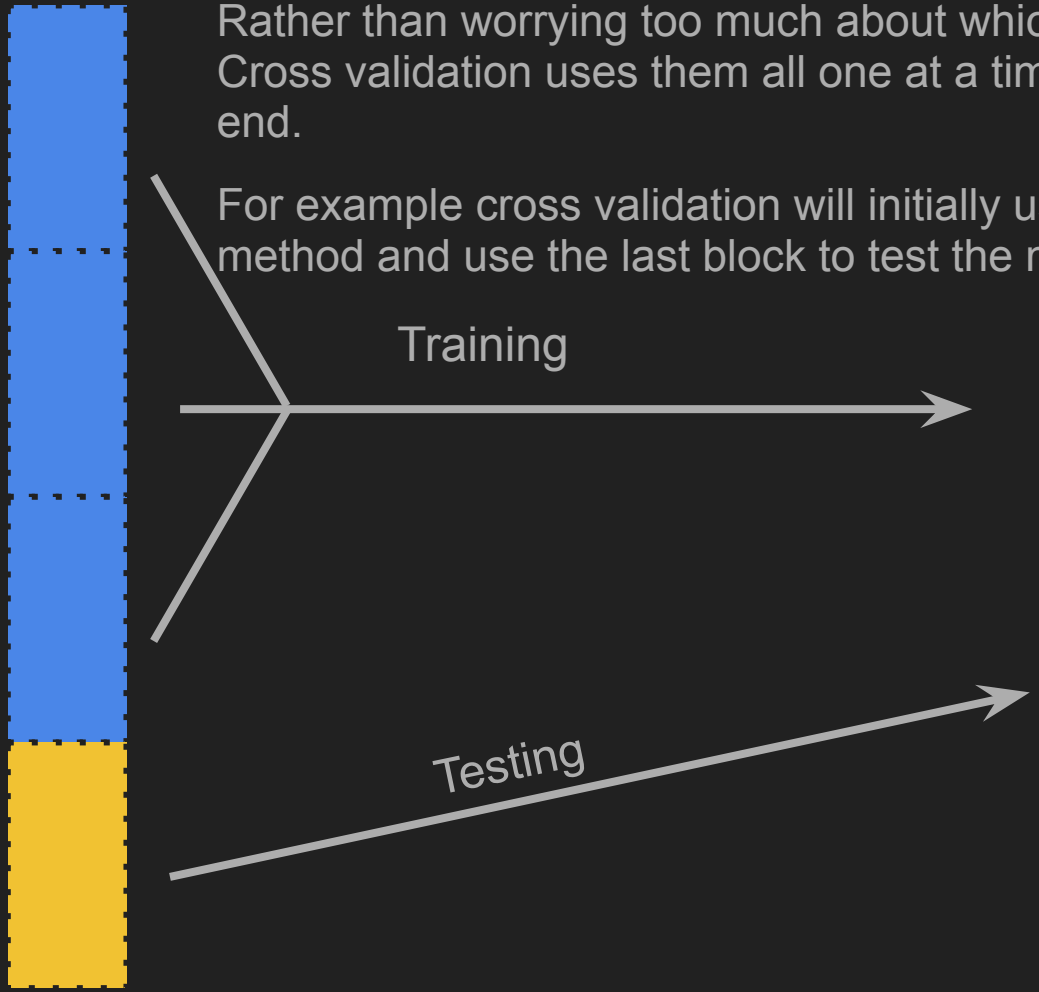


We can then compare each ML method to see how well each algorithm categorized the test data

How do we know that using the first 75% of the data for training and the last 25% of the data for testing is the best way to divide up the data?

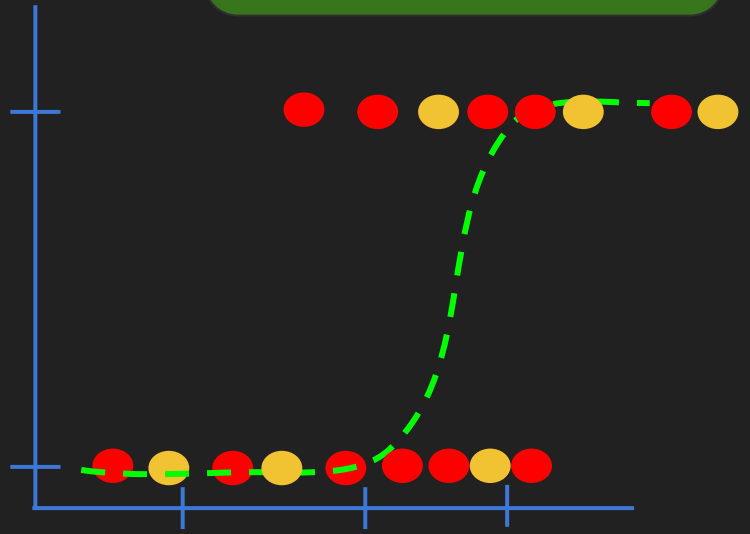
Rather than worrying too much about which block would be best for testing, Cross validation uses them all one at a time and summarizes the results at the end.

For example cross validation will initially use the first three blocks to train the method and use the last block to test the method



Test Data categorization

Correct	Incorrect
5	1



Rather than worrying too much about which block would be best for testing, Cross validation uses them all one at a time and summarizes the results at the end.

For example cross validation will initially use the first three blocks to train the method and use the last block to test the method

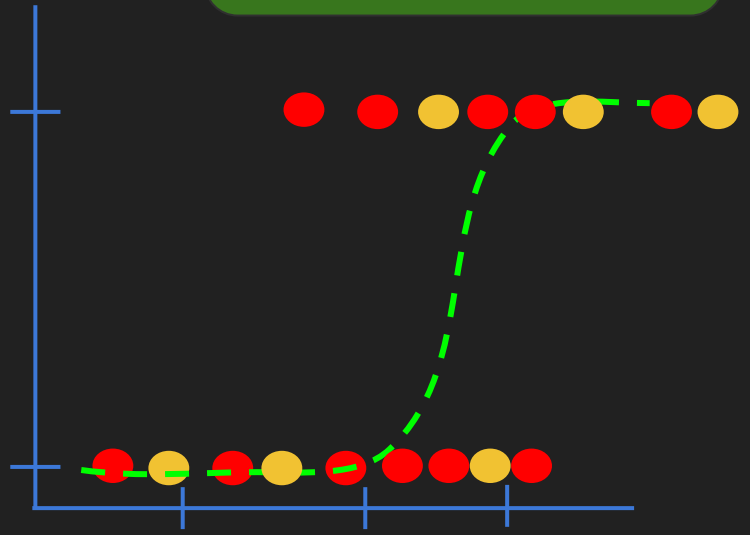
Training

Testing

Then it uses the next combination to train and test the data. And keeps repeating this thill all combination are accounted for.

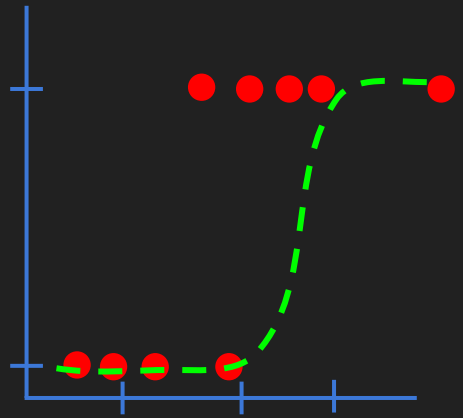
Test Data categorization

Correct	Incorrect
4	2



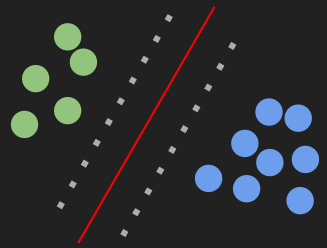
In the end, every block of data is used for testing and we can compare methods by seeing how well they performed.

Logistic Regression



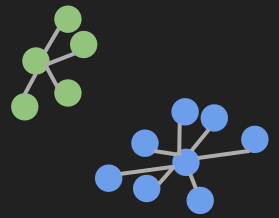
Correct	Incorrect
5	1

Support Vector Machines(SVM)



Correct	Incorrect
18	6

K-nearest neighbours



Correct	Incorrect
10	12

In this example we divided the data into 4 block. This is called "Four Block Cross Validation"

The number of blocks is arbitrary

In an extreme case, we could call each individual data (or sample) a block. This is called "Leave One Out Cross Validation"

Ideally it is pretty normal to divide the data into ten block, this is called "Ten Fold Cross Validation"

Note: For calculating the tuning parameter(λ) in Lasso & Ridge regression we use Ten Fold Cross Validation