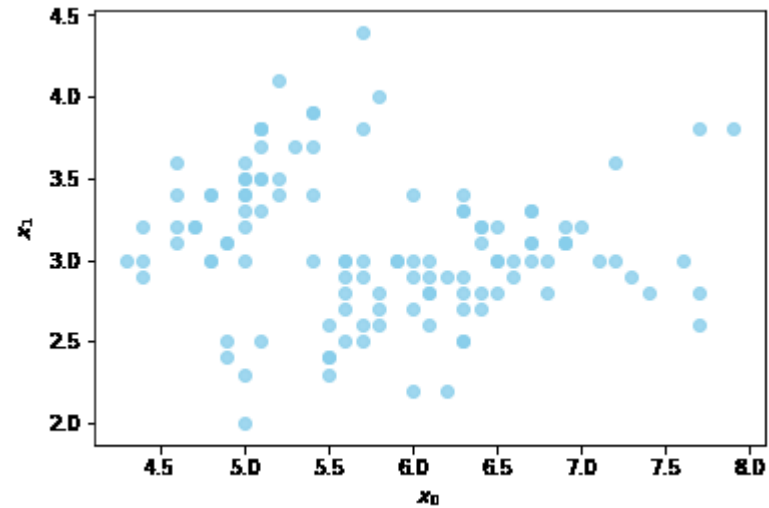


TRAIN TEST SPLIT

<https://machinelearning.com/>

Data set



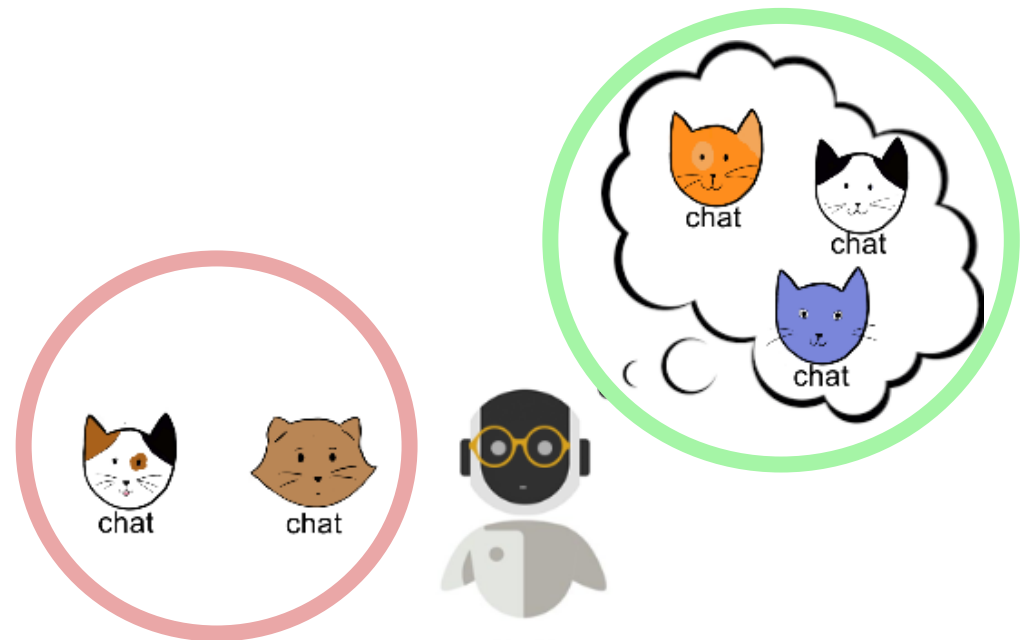
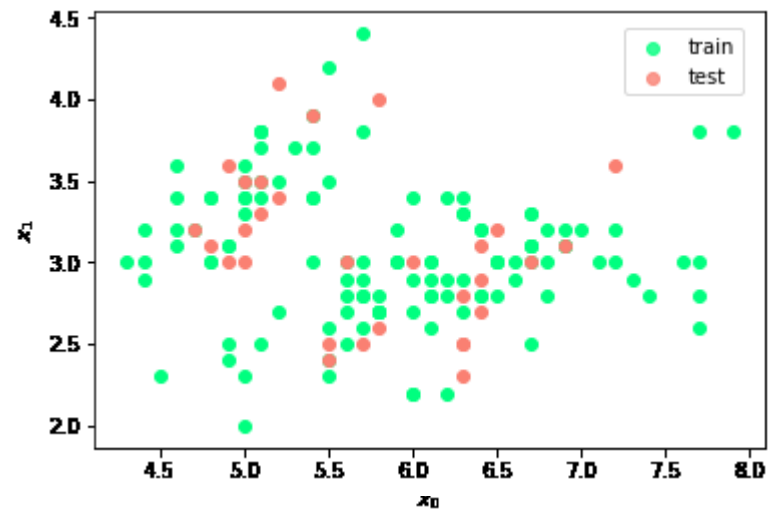
TRAIN TEST SPLIT

<https://machinelearning.com/>

Data set

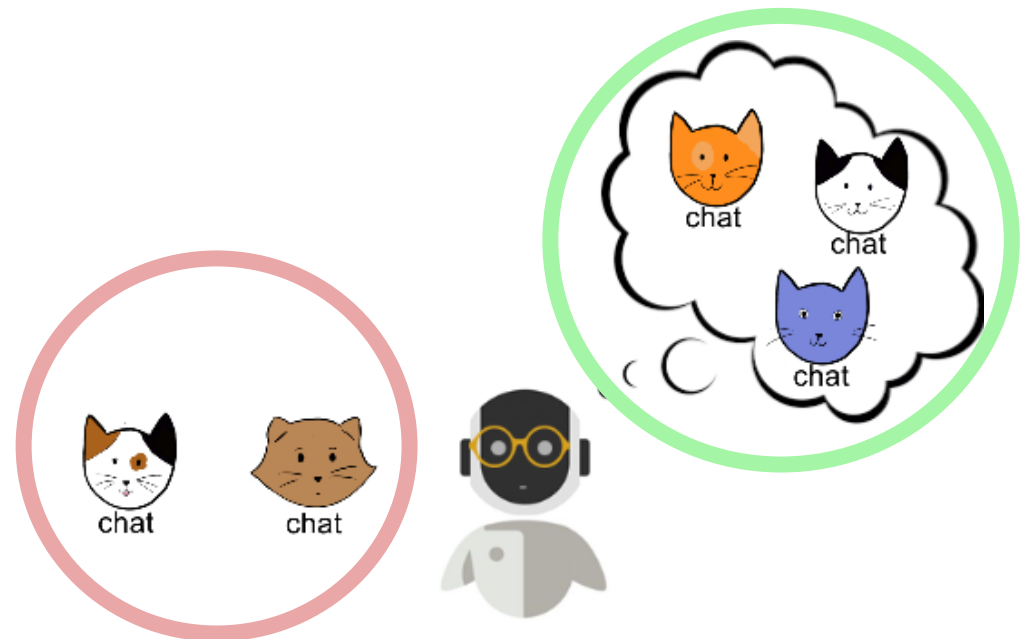
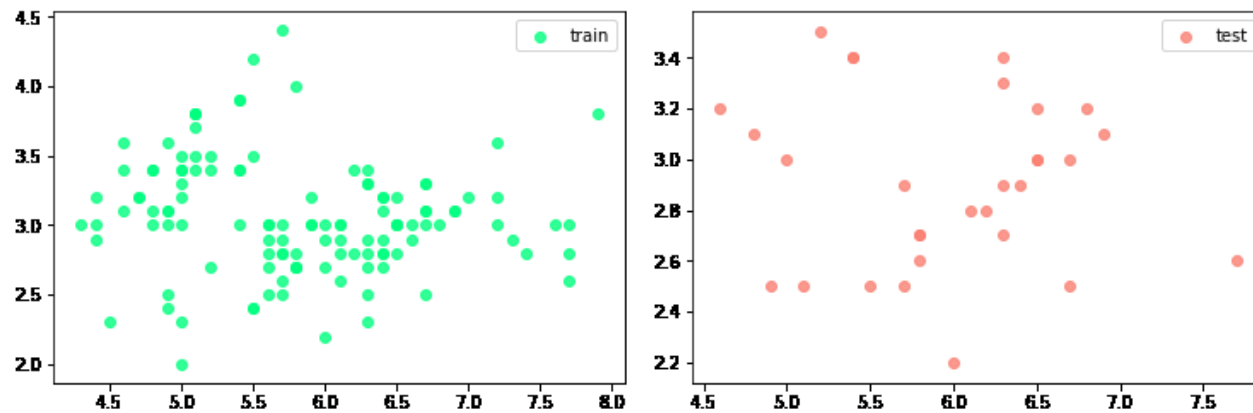
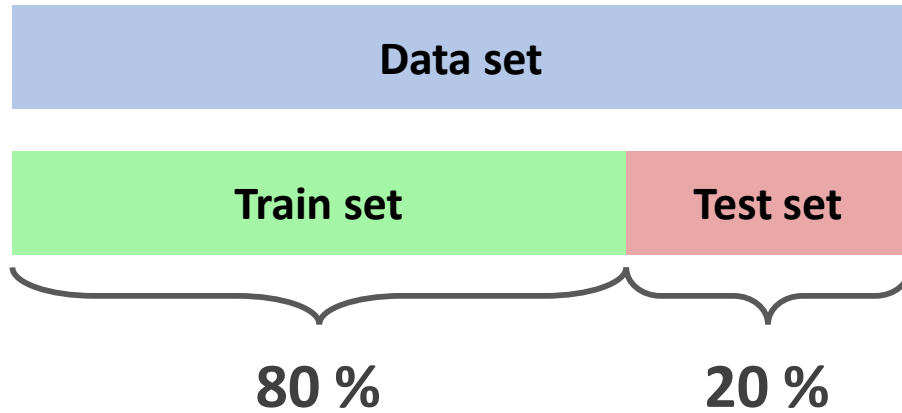
Train set

Test set



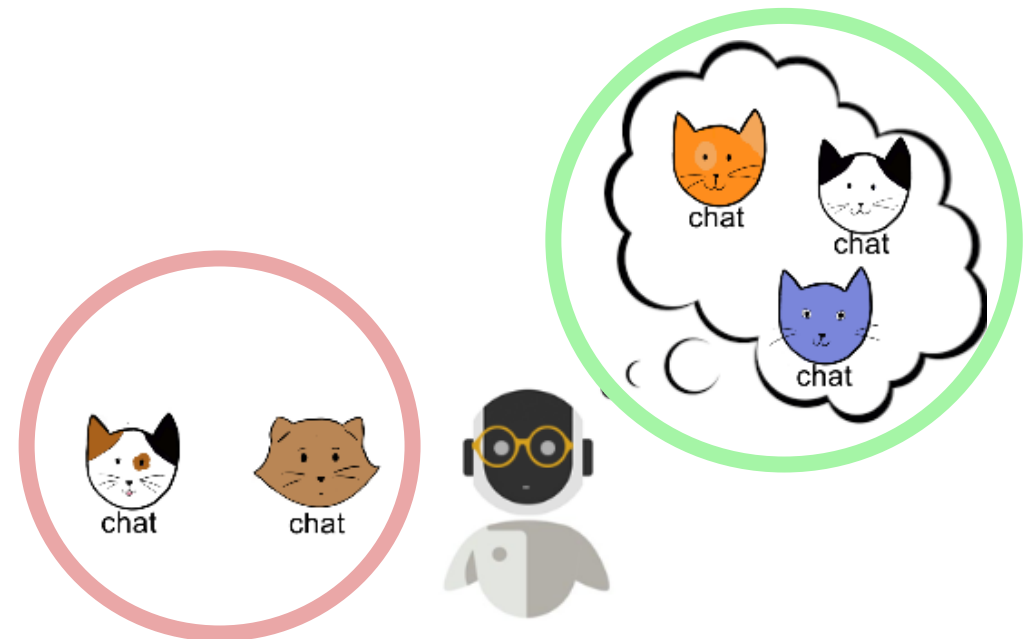
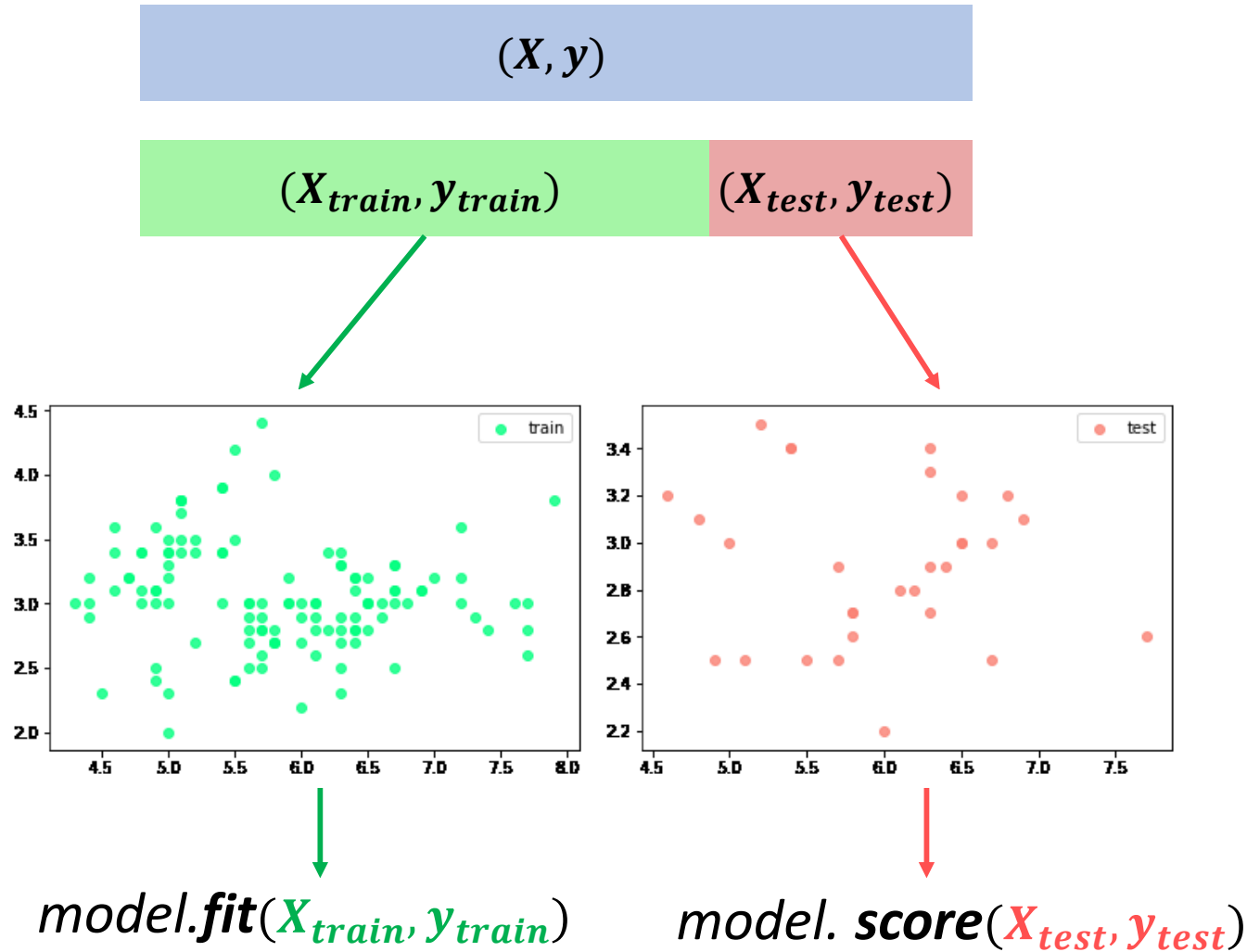
TRAIN TEST SPLIT

<https://machinelearningmastery.com/>



TRAIN TEST SPLIT

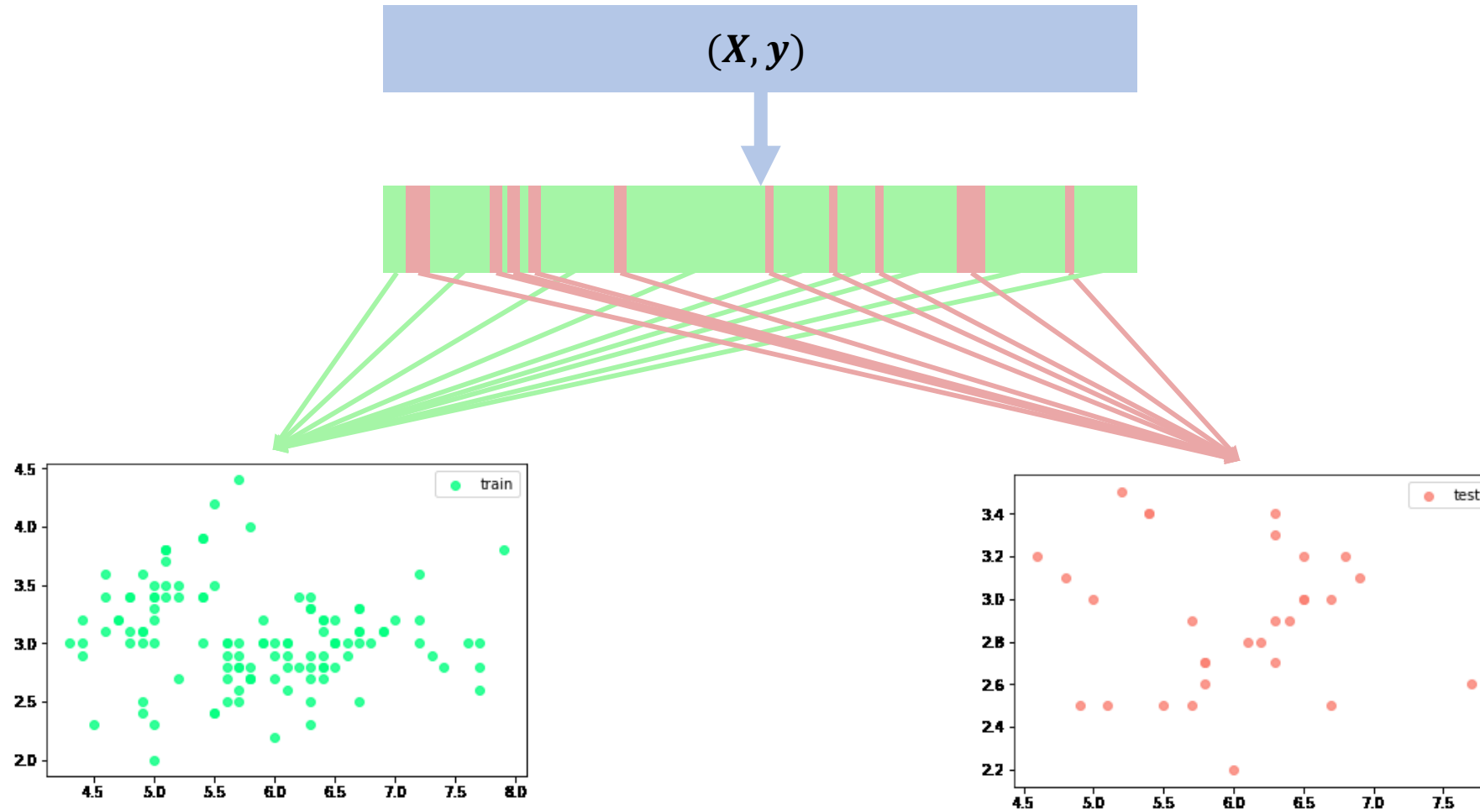
<https://machinelearning.com/>



TRAIN TEST SPLIT

[https://machinelearningmastery.com/](https://machinelearningmastery.com/train-test-split/)

$X_{train}, X_{test}, y_{train}, y_{test} = \text{train_test_split}(X, y, \text{random_state} = 0)$



VALIDATION SET

<https://machinelearning.com/>

Data set

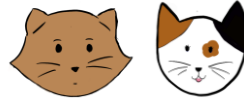
Train set



Val set





Test set




VALIDATION SET

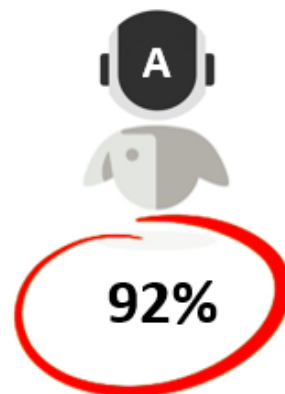
<https://machinelearnia.com/>






Data set			
	Train set	Val set	Test set
	100%	92%	91%
	100%	90%	

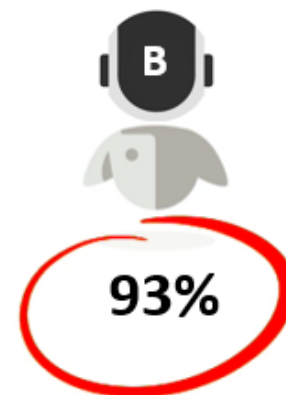
VALIDATION SET

<https://machinelearning.com/>

Train set	Val set
  	 

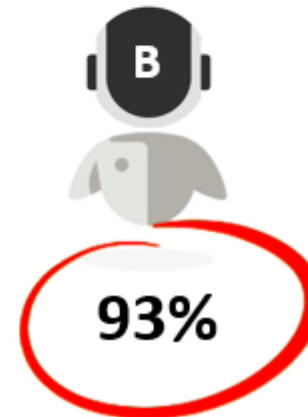
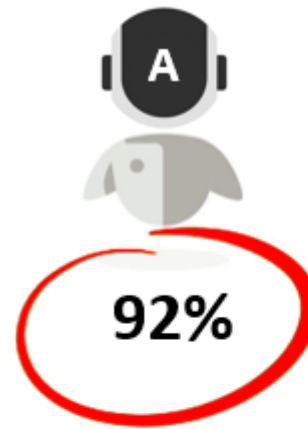


Val set	Train set
 	  



VALIDATION SET

<https://machinelearnia.com/>



CROSS VALIDATION

<https://machinelearningmastery.com/>

	Train set					 A	 B
Split 1	Val	Train	Train	Train	Train	0.92	0.91
Split 2	Train	Val	Train	Train	Train	0.88	0.90
Split 3	Train	Train	Val	Train	Train	0.89	0.91
Split 4	Train	Train	Train	Val	Train	0.93	0.92
Split 5	Train	Train	Train	Train	Val	0.86	0.90
						0.89	0.92

VALIDATION CURVE

[https://machinelearningmastery.com/](https://machinelearningmastery.com/understanding-validation-curves-for-machine-learning/)

```
validation_curve(model, X_train, y_train,  
                 'hyperparamètre', valeurs, cv=5)
```



train_score, val_score

→ Teste toutes les **valeurs** pour un **hyperparamètre** donné.

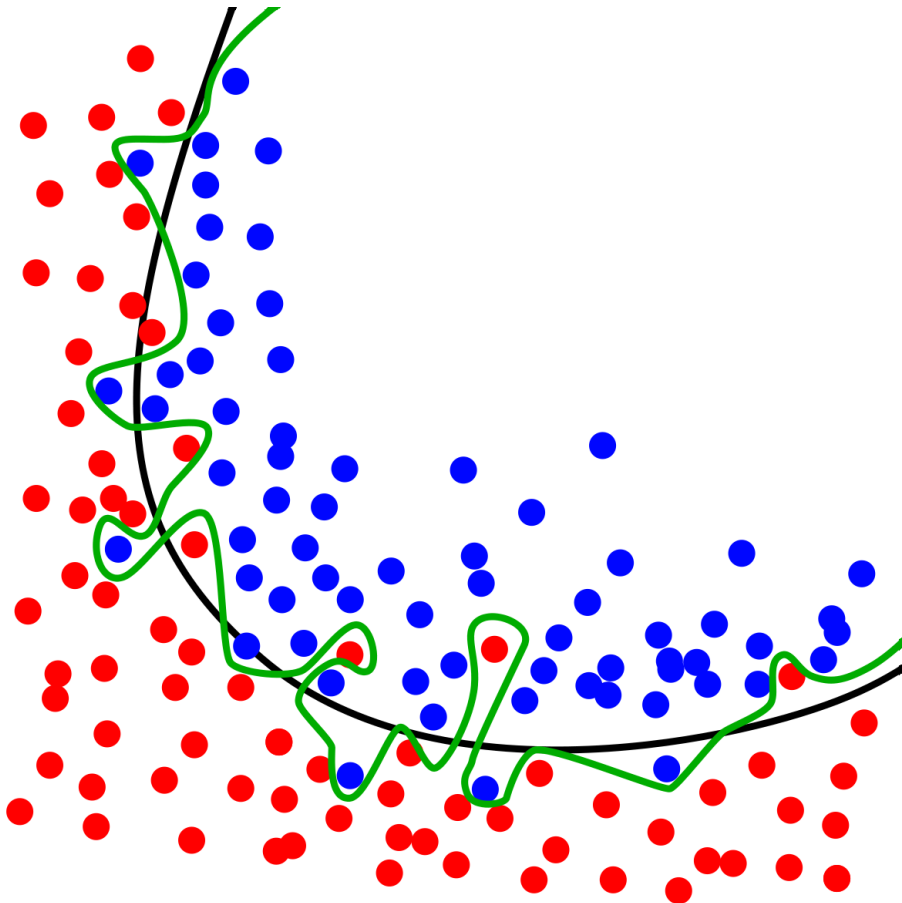
Calcule le **score** sur **Train set** et **Val set** grâce à la Cross Validation

Exemple: **49 valeurs** sont testées avec **cv = 5**

→ alors **train_score, val_score** sont de dimensions **(49, 5)**

OVERFITTING

<https://machinelearningia.com/>



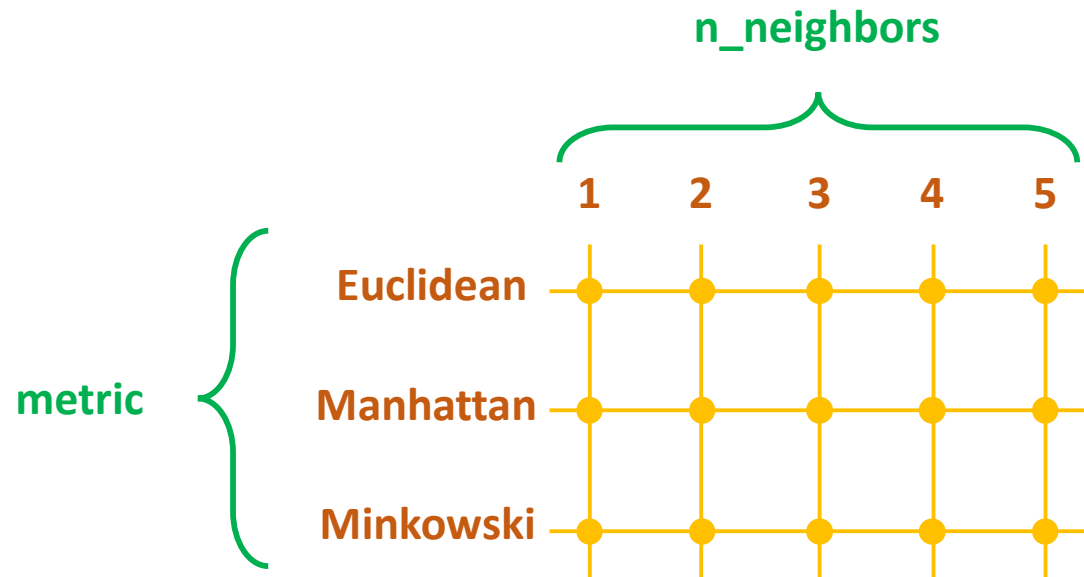
Overfitting: Le modèle s'est trop perfectionné sur le Trainset et a **perdu tout sens de généralisation**.

→ Bon $Train_{score}$

→ Mauvais $Test_{score}$

```
Grid = GridSearchCV(model, param_grid, cv)
```

→ Construit une grille de modèles avec toutes les combinaisons d'hyperparamètres présents dans **param_grid**



La grille est un **estimator** !

→ entraînement: **fit()**

→ **best_score_**

→ **best_params_**

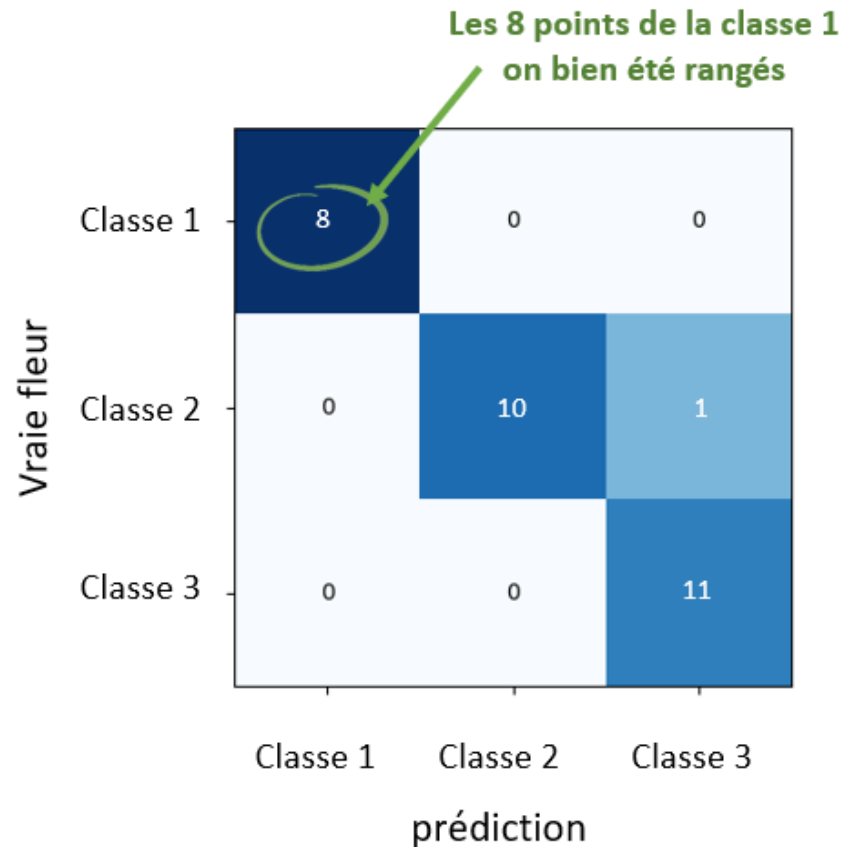
→ **best_estimator_**

CONFUSION MATRIX

<https://machinelearnia.com/>

→ Outil de mesure **très utile** pour évaluer la qualité d'un modèle de **classification**.

→ Montre les **erreurs de classement**:

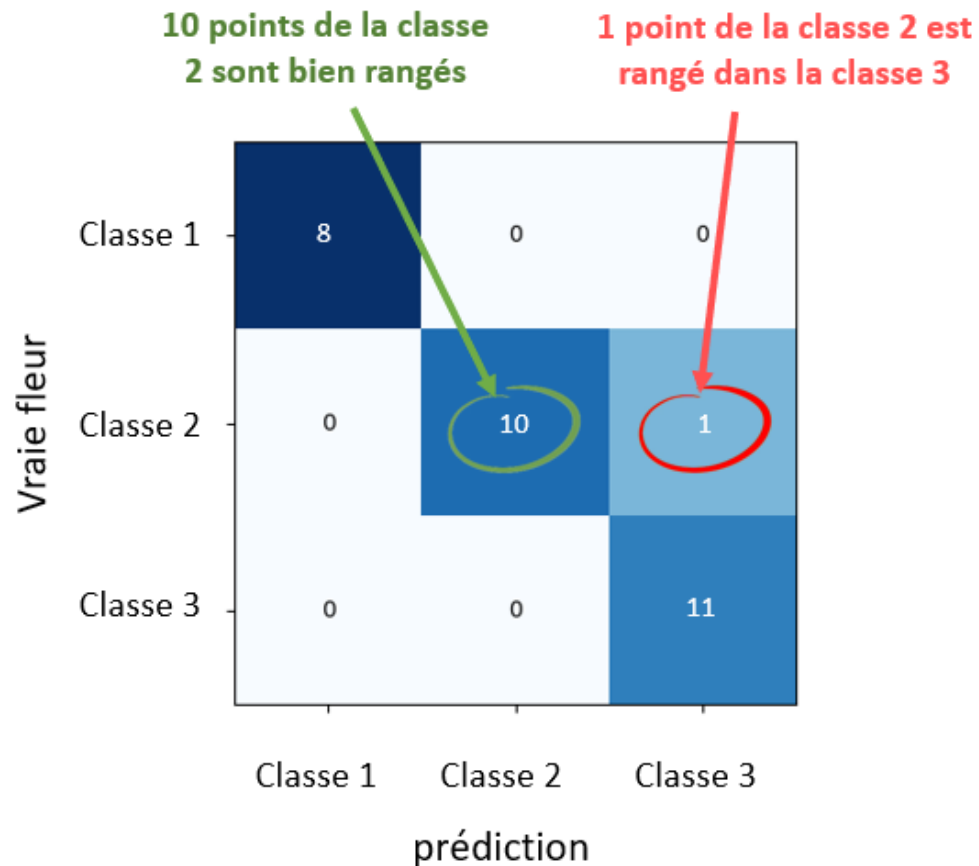


CONFUSION MATRIX

<https://machinelearningia.com/>

→ Outil de mesure **très utile** pour évaluer la qualité d'un modèle de **classification**.

→ Montre les **erreurs de classement**:



LEARNING CURVE

[https://machinelearningmastery.com/](https://machinelearningmastery.com/learning-curve/)

`learning_curve(model, X, y, train_sizes, cv=5)`

└─ N , *train_score*, *val_score*

(X_{train}, y_{train})

Train

Val

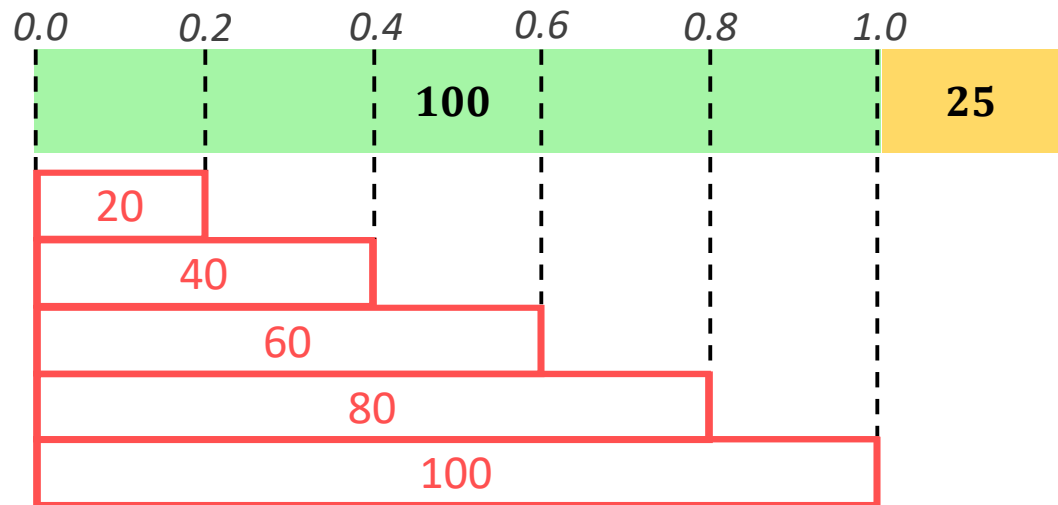
LEARNING CURVE

<https://machinelearnia.com/>

`learning_curve(model, X, y, train_sizes, cv=5)`

↪ *N*, *train_score*, *val_score*

125 points



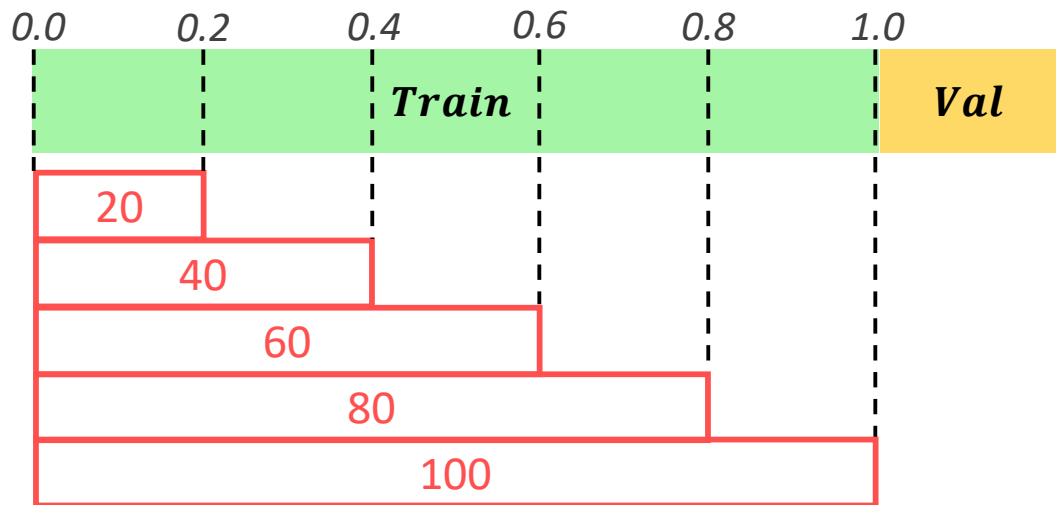
LEARNING CURVE

[https://machinelearningmastery.com/](https://machinelearningmastery.com/learning-curve/)

```
learning_curve(model, X, y, train_sizes, cv=5)
```

↳ *N*, *train_score*, *val_score*

```
train_sizes = np.linspace(0.2, 1.0, 5)
```



LEARNING CURVE

<https://machinelearnia.com/>

`learning_curve(model, X, y, train_sizes, cv=5)`

