tp1-entrainement

November 26, 2024

TD - TP 1 ENTRAINEMENT D'UN MODELE

—-COMPTE RENDU JOSEPH MBODE ——

Exercice 1 : Construction du jeu d'entrainement et de test

[1]:

```python
# 1- données de fleurs d'iris
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris

iris = load_iris()
X = iris.data
y = iris.target
```

[2]:

```python
# 2- Nombre d'exemple dans ce jeu
print("Nombre d'exmples est de :",X.shape)
```

```
Nombre d'exmples est de : (150, 4)
```

1 3- Label de cette base

d'après la taille des données, on obeserve 150 labels

2 4- Classe identifiées de cette base

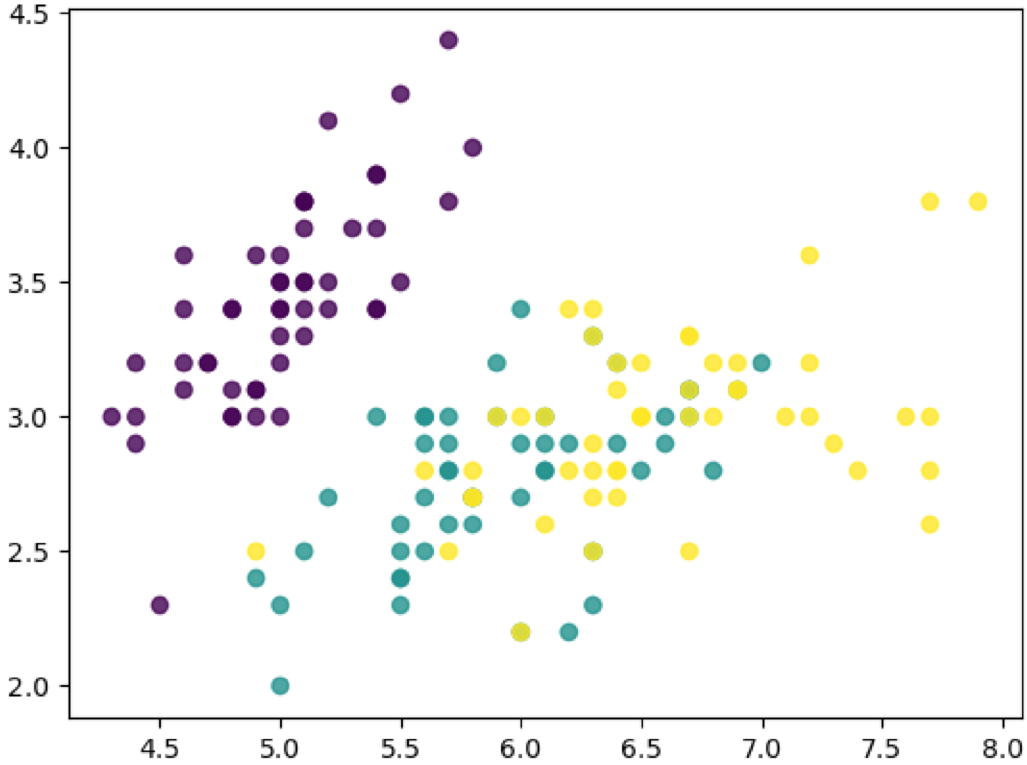
d'après la taille des données, on obeserve 4 classes

[3]:

```python
# 5- Nuage de points
plt.scatter(X[:,0],X[:,1],c=y,alpha=0.8)
```

[3]:

```
<matplotlib.collections.PathCollection at 0x11aca0b2310>
```



Graphiquement on observe 3 classes

```
[4]: # 6 -Divisions du jeu de données
     from sklearn.model_selection import train_test_split
     #X_train, X_test, Y_train, Y_test = train_test_split(X,y,test_size = t)
```

```
[5]: #   a) pour t = 0.5, affichage des dimensions
     X_train, X_test, Y_train, Y_test = train_test_split(X,y,test_size = 0.5)

     print('Train set:', X_train.shape)
     print('Test set:', X_test.shape)
```

```
Train set: (75, 4)
Test set: (75, 4)
```

```
[6]: #   b)pour t = 0.2, affichage des dimensions
     X_train, X_test, Y_train, Y_test = train_test_split(X,y,test_size = 0.2)

     print('Train set:', X_train.shape)
     print('Test set:', X_test.shape)
```

```
Train set: (120, 4)
Test set: (30, 4)
```

3 c)Utilité du paramètre t

il sert à séparer les données en pourcentage ‘1-t’ d’entrainenment et ‘t’ de test

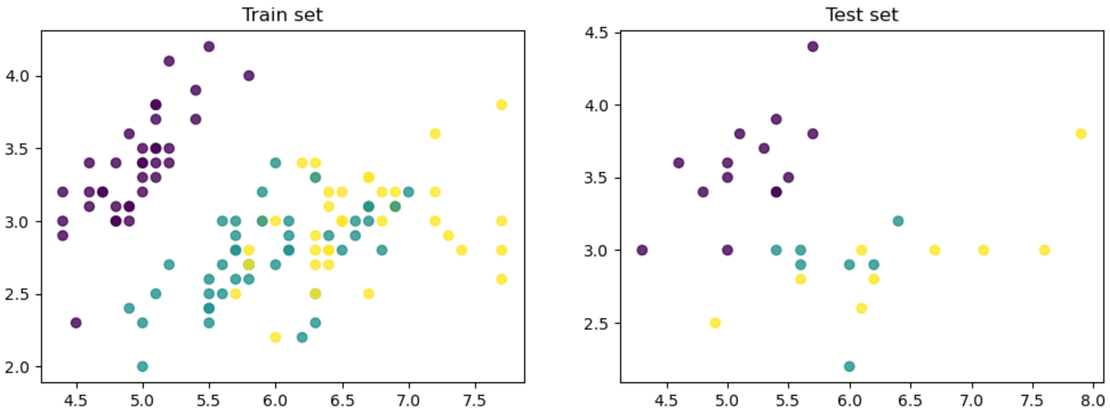
[7]:
```
#7-pour 80% d'entrainement
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size = 0.2)
```

[8]:
```
#   a) nuage de point
plt.figure(figsize=(12,4))

plt.subplot(1,2,1)
plt.scatter(X_train[:,0],X_train[:,1],c=y_train, alpha =0.8)
plt.title('Train set')

plt.subplot(1,2,2)
plt.scatter(X_test[:,0],X_test[:,1], c=y_test, alpha=0.8)
plt.title('Test set')
```

[8]:
```
Text(0.5, 1.0, 'Test set')
```



#b) relancer a) commenter On a une séparation des classes légèrement différente et une densité de points différentes, mais avec une dispersion similaire et unique malgré les relances

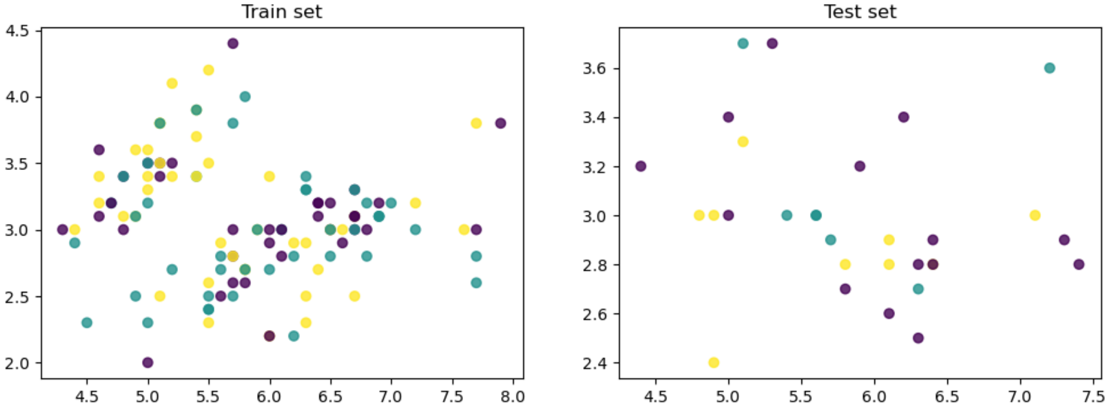
[9]:
```
# 8- reconstruction du jeu d'entrainement et de test avec le paramètre random
X_train, X_test, Y_train, Y_test = train_test_split(X,y,test_size = 0.
 ↪2,random_state=5)
```

[10]:
```
#   a) visualisation des nuages de points
#X_train, X_test, Y_train, Y_test = train_test_split(X,y,test_size = 0.
 ↪2,random_state=5)


plt.figure(figsize=(12,4))
plt.subplot(1,2,1)
plt.scatter(X_train[:,0],X_train[:,1], c=y_train, alpha = 0.8)
plt.title('Train set')

plt.subplot(1,2,2)
plt.scatter(X_test[:,0],X_test[:,1], c=y_test, alpha = 0.8)
plt.title('Test set')
```

[10]:
```
Text(0.5, 1.0, 'Test set')
```



[11]:
```
#   b) rôle du paramètre random_state

#on assure la même répartition à chaque fois qu'on executera le code
```

Exercice 2 : Entrainement et évaluation

[12]:
```
from sklearn.neighbors import KNeighborsClassifier
```

[13]:
```
#1- modèle en fixant le nombre de voisin à 1
model = KNeighborsClassifier(1)
```

[14]:
```
#2- Entrainement  des données avec la méthode fit
model.fit(X_train,y_train)
```

[14]:
```
KNeighborsClassifier(n_neighbors=1)
```

[15]:
```
#3- Evaluation du model (model.score)
#   a) sur les données train
m_s=model.score(X_train,y_train)
print(f'Données entrainée: le modèle a {m_s*100}% de probabilité de véracité'.
 ↪format(m_s))
```

```python
#   b) sur les données non vues par le modèle
m_n_s = model.score(X_test,y_test)
print(f'Données non entrainées: le modèle a {m_n_s*100}% de probabilité de␣
 ↪véracité'.format(m_n_s))
```

```
Données entrainée: le modèle a 100.0% de probabilité de véracité
Données non entrainées: le modèle a 30.0% de probabilité de véracité
```

Exercice 3 :Amélioration de l'entrainement et évaluation du modèle

1 - Comparaison entre 3 & 4 voisins

[16]:
```python
model3 = KNeighborsClassifier(3)
model4 = KNeighborsClassifier(4)
```

[17]:
```python
X_train, X_test, Y_train, Y_test = train_test_split(X,y,test_size = 0.
 ↪2,random_state=15)
```

[18]:
```python
print('entrainement sur les données de l Iris')
model3.fit(X_train,Y_train)
m3_s = model3.score(X_train,Y_train)
print('score du model3 {}'.format(m3_s))

model4.fit(X_train,Y_train)
m4_s = model4.score(X_train,Y_train)
print('score du model4 {}'.format(m4_s))
```

```
entrainement sur les données de l Iris
score du model3 0.95
score du model4 0.95
```

2- evaleuation finale : l'évaluation finale sera faite sur le score 4,

2-

[ ]:
```python
```

[ ]:
```python
```

3- modification du jeu et nouvelle performance :

[19]:
```python
X_train, X_test, Y_train, Y_test = train_test_split(X,y,test_size = 0.
 ↪3,random_state=40)
```

[20]:
```python
model3.fit(X_train,Y_train)
m3_s = model3.score(X_train,Y_train)
print('score du model3 {}'.format(m3_s))
```

```
model4.fit(X_train,Y_train)
m4_s = model4.score(X_train,Y_train)
print('score du model4 {}'.format(m4_s))
```

```
score du model3 0.9523809523809523
score du model4 0.9428571428571428
```

Exercice 4 : Optimisation du modèle de recherche

[21]:
```
from sklearn.model_selection import GridSearchCV
```

1-définitions hyperparamètres

[22]:
```
param_grid = {'n_neighbors':np.arange(1,50), 'metric':['euclidian','manhattan']}
```

2-Grille d'estimateurs

[23]:
```
Grid = GridSearchCV(KNeighborsClassifier(), param_grid, cv=5)
```

3- Construction du modèle

[24]:
```
Grid.fit(X_train,Y_train)
```

```
C:\ProgramData\anaconda3\Lib\site-
packages\sklearn\model_selection\_validation.py:378: FitFailedWarning:
245 fits failed out of a total of 490.
The score on these train-test partitions for these parameters will be set to
nan.
If these failures are not expected, you can try to debug them by setting
error_score='raise'.

Below are more details about the failures:
--------------------------------------------------------------------------------
245 fits failed with the following error:
Traceback (most recent call last):
  File "C:\ProgramData\anaconda3\Lib\site-
packages\sklearn\model_selection\_validation.py", line 686, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\ProgramData\anaconda3\Lib\site-
packages\sklearn\neighbors\_classification.py", line 213, in fit
    self._validate_params()
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 600,
in _validate_params
    validate_parameter_constraints(
  File "C:\ProgramData\anaconda3\Lib\site-
packages\sklearn\utils\_param_validation.py", line 97, in
validate_parameter_constraints
```

```
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'metric' parameter of
KNeighborsClassifier must be a str among {'jaccard', 'matching', 'l1',
'chebyshev', 'haversine', 'hamming', 'cityblock', 'correlation', 'canberra',
'sokalsneath', 'dice', 'yule', 'pyfunc', 'wminkowski', 'manhattan',
'braycurtis', 'l2', 'sokalmichener', 'precomputed', 'sqeuclidean',
'nan_euclidean', 'minkowski', 'p', 'mahalanobis', 'kulsinski', 'cosine',
'russellrao', 'euclidean', 'seuclidean', 'rogerstanimoto', 'infinity'} or a
callable. Got 'euclidian' instead.

  warnings.warn(some_fits_failed_message, FitFailedWarning)
C:\ProgramData\anaconda3\Lib\site-
packages\sklearn\model_selection\_search.py:952: UserWarning: One or more of the
test scores are non-finite: [       nan        nan        nan        nan
nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
        nan 0.91428571 0.9047619  0.92380952 0.93333333 0.93333333
 0.93333333 0.94285714 0.93333333 0.93333333 0.93333333 0.95238095
 0.94285714 0.93333333 0.94285714 0.94285714 0.93333333 0.92380952
 0.93333333 0.94285714 0.94285714 0.94285714 0.93333333 0.91428571
 0.9047619  0.91428571 0.9047619  0.94285714 0.91428571 0.91428571
 0.9047619  0.9047619  0.8952381  0.91428571 0.8952381  0.91428571
 0.91428571 0.91428571 0.88571429 0.8952381  0.8952381  0.88571429
 0.87619048 0.87619048 0.88571429 0.88571429 0.87619048 0.87619048
 0.85714286 0.86666667]
  warnings.warn(
```

[24]:
```
GridSearchCV(cv=5, estimator=KNeighborsClassifier(),
             param_grid={'metric': ['euclidian', 'manhattan'],
                         'n_neighbors': array([ 1,  2,  3,  4,  5,  6,  7,  8,
 9, 10, 11, 12, 13, 14, 15, 16, 17,
       18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
       35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49])})
```

4- Meilleurs paramètres et scores

[25]:
```
Grid.best_params_
```

[25]:
```
{'metric': 'manhattan', 'n_neighbors': 11}
```

[26]:
```
Grid.best_score_
```

[26]:
```
0.9523809523809523
```

5- sauvegarde du modèle

[27]:
```
model = Grid.best_estimator_
```

6- evaluation des performances sur le jeu test:

[28]:
```
model.score(X_test,Y_test)
```

[28]:
```
0.9777777777777777
```

7- Performance grace à la matrice de confusionµ

[29]:
```
from sklearn.metrics import confusion_matrix

confusion_matrix(Y_test, model.predict(X_test))
```

[29]:
```
array([[16,  0,  0],
       [ 0, 14,  0],
       [ 0,  1, 14]], dtype=int64)
```

Interpretation : le modèle semble performant avec 44 prédictions correctes sur les 45 testées, sauf 1 seule instance de classe de 2 identifiées comme classe 1

Exercice 5 : Interpretation des courbe par apprentissage

[30]:
```
from sklearn.model_selection import learning_curve
```

1- representation de N: N représente la proportion , ou le nombre de données d'entrainement utilisées pour la courbe d'apprentisage

[31]:
```
pourcentage_debut = 0.1
pourcentage_fin = 1.0
nombre_de_lots = 10
```

[32]:
```
N, train_score, val_score = learning_curve(model, X_train, Y_train, train_sizes␣
 ↪= np.linspace(pourcentage_debut, pourcentage_fin, nombre_de_lots), cv = 5)
```

```
C:\ProgramData\anaconda3\Lib\site-
packages\sklearn\model_selection\_validation.py:778: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:
Traceback (most recent call last):
  File "C:\ProgramData\anaconda3\Lib\site-
packages\sklearn\model_selection\_validation.py", line 767, in _score
    scores = scorer(estimator, X_test, y_test)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\metrics\_scorer.py",
line 444, in _passthrough_scorer
    return estimator.score(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 668,
in score
    return accuracy_score(y, self.predict(X), sample_weight=sample_weight)
                             ^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-
packages\sklearn\neighbors\_classification.py", line 234, in predict
    neigh_ind = self.kneighbors(X, return_distance=False)
                ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\neighbors\_base.py",
line 810, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples,  but n_samples = 8, n_neighbors =
11

  warnings.warn(
C:\ProgramData\anaconda3\Lib\site-
packages\sklearn\model_selection\_validation.py:778: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:
Traceback (most recent call last):
  File "C:\ProgramData\anaconda3\Lib\site-
packages\sklearn\model_selection\_validation.py", line 767, in _score
    scores = scorer(estimator, X_test, y_test)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\metrics\_scorer.py",
line 444, in _passthrough_scorer
    return estimator.score(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 668,
in score
    return accuracy_score(y, self.predict(X), sample_weight=sample_weight)
                             ^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-
packages\sklearn\neighbors\_classification.py", line 234, in predict
    neigh_ind = self.kneighbors(X, return_distance=False)
                ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\neighbors\_base.py",
line 810, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples,  but n_samples = 8, n_neighbors =
11

  warnings.warn(
C:\ProgramData\anaconda3\Lib\site-
```

```
packages\sklearn\model_selection\_validation.py:778: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:
Traceback (most recent call last):
  File "C:\ProgramData\anaconda3\Lib\site-
packages\sklearn\model_selection\_validation.py", line 767, in _score
    scores = scorer(estimator, X_test, y_test)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\metrics\_scorer.py",
line 444, in _passthrough_scorer
    return estimator.score(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 668,
in score
    return accuracy_score(y, self.predict(X), sample_weight=sample_weight)
                             ^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-
packages\sklearn\neighbors\_classification.py", line 234, in predict
    neigh_ind = self.kneighbors(X, return_distance=False)
                ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\neighbors\_base.py",
line 810, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples,  but n_samples = 8, n_neighbors =
11

  warnings.warn(
C:\ProgramData\anaconda3\Lib\site-
packages\sklearn\model_selection\_validation.py:778: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:
Traceback (most recent call last):
  File "C:\ProgramData\anaconda3\Lib\site-
packages\sklearn\model_selection\_validation.py", line 767, in _score
    scores = scorer(estimator, X_test, y_test)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\metrics\_scorer.py",
line 444, in _passthrough_scorer
    return estimator.score(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 668,
in score
    return accuracy_score(y, self.predict(X), sample_weight=sample_weight)
                             ^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-
packages\sklearn\neighbors\_classification.py", line 234, in predict
    neigh_ind = self.kneighbors(X, return_distance=False)
                ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\neighbors\_base.py",
line 810, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples,  but n_samples = 8, n_neighbors =
11

  warnings.warn(
C:\ProgramData\anaconda3\Lib\site-
packages\sklearn\model_selection\_validation.py:778: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:
Traceback (most recent call last):
  File "C:\ProgramData\anaconda3\Lib\site-
packages\sklearn\model_selection\_validation.py", line 767, in _score
    scores = scorer(estimator, X_test, y_test)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\metrics\_scorer.py",
line 444, in _passthrough_scorer
    return estimator.score(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 668,
in score
    return accuracy_score(y, self.predict(X), sample_weight=sample_weight)
                             ^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-
packages\sklearn\neighbors\_classification.py", line 234, in predict
    neigh_ind = self.kneighbors(X, return_distance=False)
                ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\neighbors\_base.py",
line 810, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples,  but n_samples = 8, n_neighbors =
11

  warnings.warn(
C:\ProgramData\anaconda3\Lib\site-
packages\sklearn\model_selection\_validation.py:778: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:
Traceback (most recent call last):
  File "C:\ProgramData\anaconda3\Lib\site-
packages\sklearn\model_selection\_validation.py", line 767, in _score
    scores = scorer(estimator, X_test, y_test)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\metrics\_scorer.py",
line 444, in _passthrough_scorer
    return estimator.score(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 668,
in score
    return accuracy_score(y, self.predict(X), sample_weight=sample_weight)
                             ^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-
packages\sklearn\neighbors\_classification.py", line 234, in predict
    neigh_ind = self.kneighbors(X, return_distance=False)
                ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\neighbors\_base.py",
line 810, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples,  but n_samples = 8, n_neighbors =
11

  warnings.warn(
C:\ProgramData\anaconda3\Lib\site-
packages\sklearn\model_selection\_validation.py:778: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:
Traceback (most recent call last):
  File "C:\ProgramData\anaconda3\Lib\site-
packages\sklearn\model_selection\_validation.py", line 767, in _score
    scores = scorer(estimator, X_test, y_test)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\metrics\_scorer.py",
line 444, in _passthrough_scorer
    return estimator.score(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 668,
in score
    return accuracy_score(y, self.predict(X), sample_weight=sample_weight)
                             ^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-
packages\sklearn\neighbors\_classification.py", line 234, in predict
    neigh_ind = self.kneighbors(X, return_distance=False)
                ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\neighbors\_base.py",
line 810, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples,  but n_samples = 8, n_neighbors =
11

  warnings.warn(
C:\ProgramData\anaconda3\Lib\site-
packages\sklearn\model_selection\_validation.py:778: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:
Traceback (most recent call last):
  File "C:\ProgramData\anaconda3\Lib\site-
packages\sklearn\model_selection\_validation.py", line 767, in _score
    scores = scorer(estimator, X_test, y_test)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\metrics\_scorer.py",
line 444, in _passthrough_scorer
    return estimator.score(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 668,
in score
    return accuracy_score(y, self.predict(X), sample_weight=sample_weight)
                             ^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-
packages\sklearn\neighbors\_classification.py", line 234, in predict
    neigh_ind = self.kneighbors(X, return_distance=False)
                ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\neighbors\_base.py",
line 810, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples,  but n_samples = 8, n_neighbors =
11

  warnings.warn(
C:\ProgramData\anaconda3\Lib\site-
packages\sklearn\model_selection\_validation.py:778: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:
Traceback (most recent call last):
  File "C:\ProgramData\anaconda3\Lib\site-
packages\sklearn\model_selection\_validation.py", line 767, in _score
    scores = scorer(estimator, X_test, y_test)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\metrics\_scorer.py",
line 444, in _passthrough_scorer
    return estimator.score(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 668,
in score
    return accuracy_score(y, self.predict(X), sample_weight=sample_weight)
                             ^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-
packages\sklearn\neighbors\_classification.py", line 234, in predict
    neigh_ind = self.kneighbors(X, return_distance=False)
                ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\neighbors\_base.py",
line 810, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples,  but n_samples = 8, n_neighbors =
11

  warnings.warn(
C:\ProgramData\anaconda3\Lib\site-
packages\sklearn\model_selection\_validation.py:778: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:
Traceback (most recent call last):
  File "C:\ProgramData\anaconda3\Lib\site-
packages\sklearn\model_selection\_validation.py", line 767, in _score
    scores = scorer(estimator, X_test, y_test)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\metrics\_scorer.py",
line 444, in _passthrough_scorer
    return estimator.score(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py", line 668,
in score
    return accuracy_score(y, self.predict(X), sample_weight=sample_weight)
                             ^^^^^^^^^^^^^^^

  File "C:\ProgramData\anaconda3\Lib\site-
packages\sklearn\neighbors\_classification.py", line 234, in predict
    neigh_ind = self.kneighbors(X, return_distance=False)
                ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

  File "C:\ProgramData\anaconda3\Lib\site-packages\sklearn\neighbors\_base.py",
line 810, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples,  but n_samples = 8, n_neighbors =
11

  warnings.warn(
```

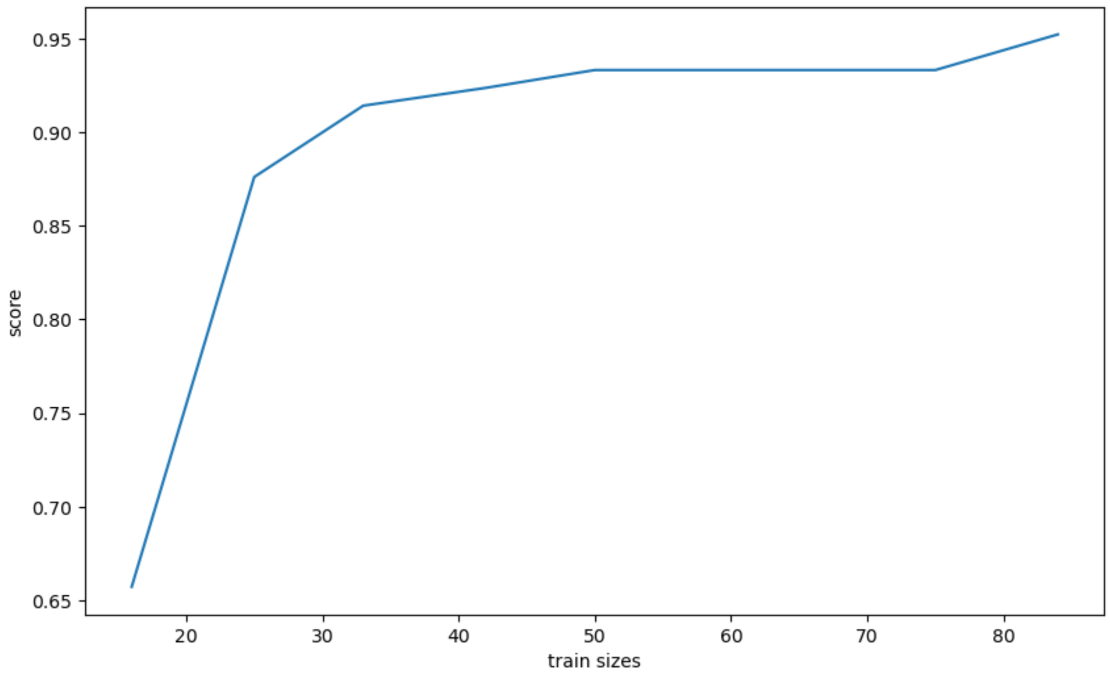
2- Scores moyens des données de validation

[33]:

```python
plt.figure(figsize=(10,6))
plt.plot(N,val_score.mean(axis=1),label='validation')
plt.ylabel('score')
plt.xlabel('train sizes')
```

[33]:

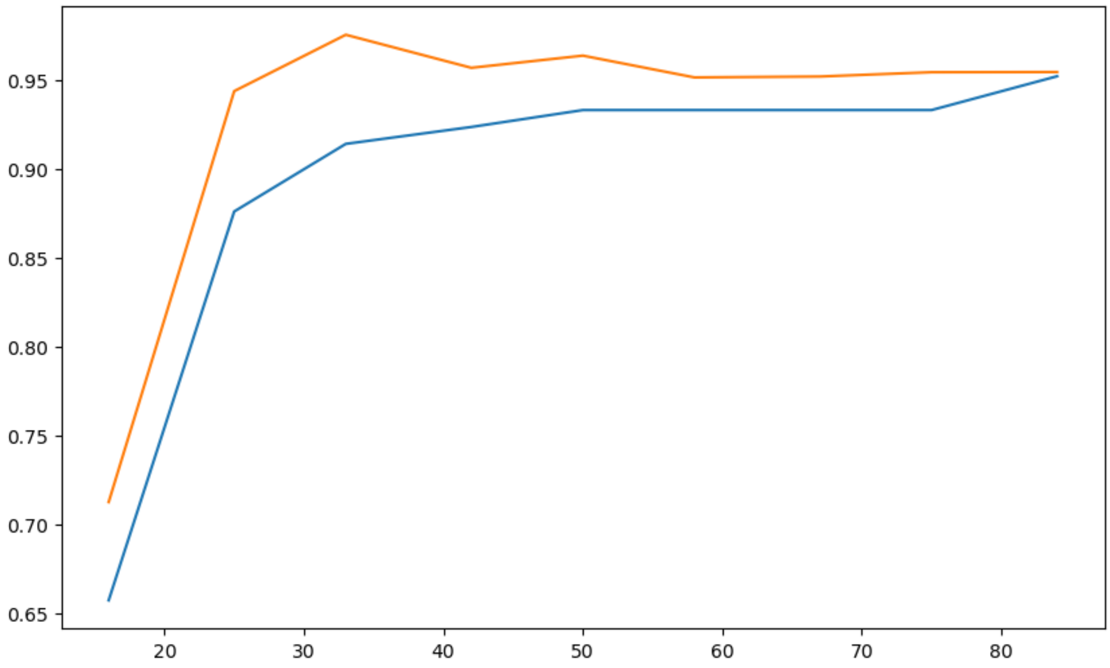
```
Text(0.5, 0, 'train sizes')
```



3- Affichage des scores de validation et d’entrainement

[34]:
```
plt.figure(figsize=(10,6))
plt.plot(N,val_score.mean(axis=1),label='validation')
plt.plot(N,train_score.mean(axis=1),label='entrainement')
```

[34]:
```
[<matplotlib.lines.Line2D at 0x11acc360990>]
```



4- Interpretation :

Vu que la validation converge à peu près en 95 avec les données d'entrainement vues par le modèle, on peut dire que celui-ci n'a pas besoin de rajouter plus de données