

La lecture d'un rapport statistique de classification

Table des matières

I. Métriques de la classification	3
A. Comprendre et analyser la classification	3
II. Exercice : Quiz	7
III. Exemple de classification	8
A. Machine learning : classification.....	8
IV. Exercice : Quiz	12
V. Essentiel	13
VI. Auto-évaluation	13
A. Exercice	13
B. Test.....	14
Solutions des exercices	16

I. Métriques de la classification

Durée : 1 h

Environnement de travail : visual studio

Prérequis : Python & bases de ML

Contexte

L'évaluation d'un modèle de *machine learning* pour un problème de classification passe par l'interprétation du rapport de performance de ce modèle. Nous allons d'abord aborder les concepts théoriques : précision, Recall, spécificité, sensibilité. Nous appliquerons ces connaissances à un exemple.

A. Comprendre et analyser la classification

Pourquoi la métrique accuracy n'est pas suffisante ?

Avant de savoir si la précision est une bonne ou une mauvaise mesure à utiliser pour évaluer un modèle, il est indispensable de définir ce qu'est la précision dans un problème de classification ML. La condition ou la qualité d'être vrai, correct ou exact ; l'absence d'erreur ou de défaut est l'exactitude en anglais.

Exemple

Parlons de la précision en ML à l'aide d'un exemple : les données d'accidents de drones.

Il existe deux modalités :

- Atterrissage en toute sécurité (étiquette = 1)
- Crashed (étiquette = 0)

Un modèle capable de prédire est censé donner une précision de 100 % dans un problème de classification binaire :

- Tous les « Atterrissage en toute sécurité (étiquette = 1) » comme Atterri en toute sécurité (prédiction = 1)
- Tous les accidents « Crashed (étiquette = 0) » comme Crashed (étiquette = 0)

En réalité, obtenir 100 % de précision ou 0 % d'erreur est impossible.

Le taux de bonnes prédictions peut être utilisé comme métrique d'évaluation sous certaines réserves.

Lorsque les données sont équilibrées, la précision est une bonne mesure pour évaluer notre modèle. Par contre, si les données sont déséquilibrées, l'exactitude n'est pas une mesure correcte de l'évaluation.

Qu'entend-on par données équilibrées et déséquilibrées ?

En prenant notre exemple d'accident de vol de drones, les données contiennent en termes de pourcentage d'étiquettes les valeurs ci-dessous :

- Atterri en toute sécurité (étiquette = 1) à hauteur d'environ 50 %
- Crashed (étiquette = 0) environ 50 % dans la variable cible

Elles sont alors considérées comme des données équilibrées.

Par contre, si Atterri en toute sécurité (étiquette = 1) est en pourcentage supérieur à Crashed (0), les données sont considérées comme des données déséquilibrées.

Exemple

Comprenons pourquoi la précision est médiocre pour des données déséquilibrées :

Prenons l'exemple où :

- Les atterrissages en toute sécurité (étiquette = 1) représentent 97 % ,
- Les cas où les avions se sont écrasés sont de 3 % dans la variable cible.

Même lorsque le modèle ne parvient pas à prédire les accidents, sa précision est toujours de 97 % puisque les données contiennent 97 % de *Landed Safely* ou atterrissages saufs.

Ainsi, le taux de bonnes prédictions n'est pas une mesure valide pour les données déséquilibrées. Comme cette métrique n'est pas pondérée, on comprend pourquoi on ne peut pas utiliser cette métrique telle quelle pour qualifier les modèles.

Précision - Recall

Pour donner une information sur les performances du modèle, on préférera utiliser :

La sensibilité ou recall

Elle mesure la capacité du modèle à prédire un résultat positif lorsqu'il est effectivement positif.

La précision

Elle répond à la question : parmi ces prédictions positives, lesquelles étaient réellement des observations positives ?

Précision, recall ou sensibilité et précision peuvent être résumés par le tableau ci-dessous.

		RÉEL			
		0	1		
PRÉDICTION	0	TP	FN	$TP / (TP + FN)$	Sensibilité ou Recall
	1	FP	TN	$TN / (FP + FN)$	Spécificité
		$TP / (TP + FP)$	$TN / (FN + FN)$		
		Précision	Valeur négative prédictive		

Fondamental

La précision n'est pas toujours une meilleure ou une mauvaise mesure pour l'évaluation. En cas de déséquilibre des données, nous disposons de nombreuses mesures telles que le recall, la précision, le f1-score et bien plus encore. La mesure de l'évaluation dépend également du scénario et des métriques qui font sens pour l'entreprise concernée.

Matrice de confusio

En pratique, on utilise un outil qui permet d'évaluer la qualité du modèle, c'est la matrice de confusion.

Grâce à cette matrice de confusion, nous obtenons davantage de métriques pour évaluer le modèle.

- **Vrais positifs** : variables cibles étiquetées comme positives qui sont en réalité positives.
- **Faux positifs** : variables cibles étiquetées comme positives qui sont en réalité négatives.
- **Vrais négatifs** : variables cibles étiquetées comme négatives qui sont en réalité négatives.
- **Faux négatifs** : variables cibles étiquetées comme négatives qui sont en réalité positives.

Sensibilité

On définit la Sensibilité comme :

$$Se = TP / (TP + FN)$$

Elle est également connue sous le nom de taux de vrais positifs. Elle est utilisée dans le cas de tests médicaux, car le but est d'éviter que certains cas ne soient pas détectés.

Spécificité

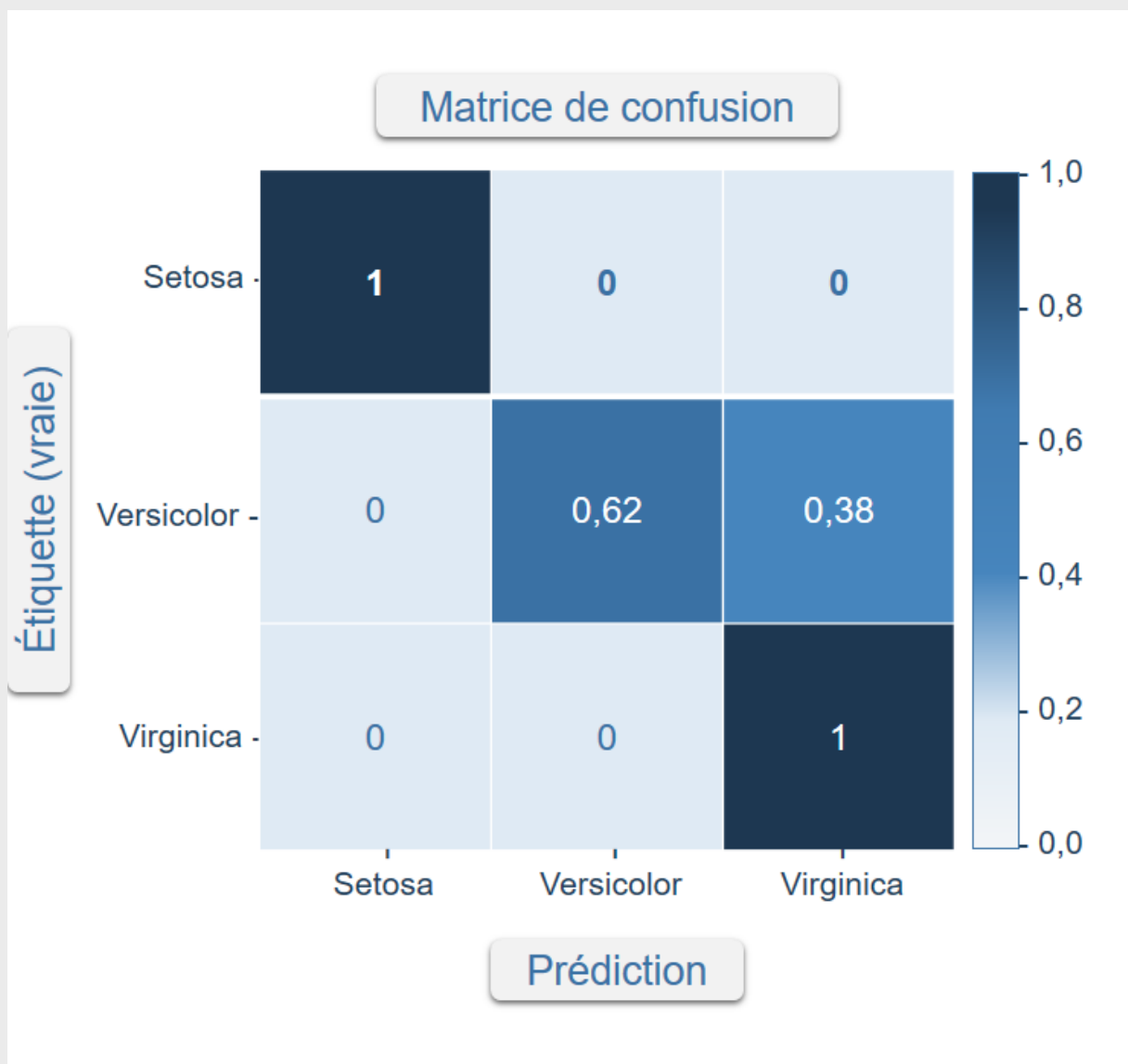
On définit la Spécificité comme :

$$Sp = TN / (TN + FP)$$

Elle est également connue sous le nom de taux de vrais négatifs. Cette mesure est utilisée dans les traitements très agressifs, car on souhaite éviter de détruire les cellules ou tissus sains.

On trouve également des matrices de confusion pour des problèmes à plusieurs classes ($n > 2$).

Exemple Matrice de confusion pour 3 classes



Les éléments sur la diagonale totalisent le nombre d'occurrences pour lesquelles les prédictions sont égales à la valeur de l'étiquette. Le modèle semble performer correctement sauf pour la catégorie versicolor.

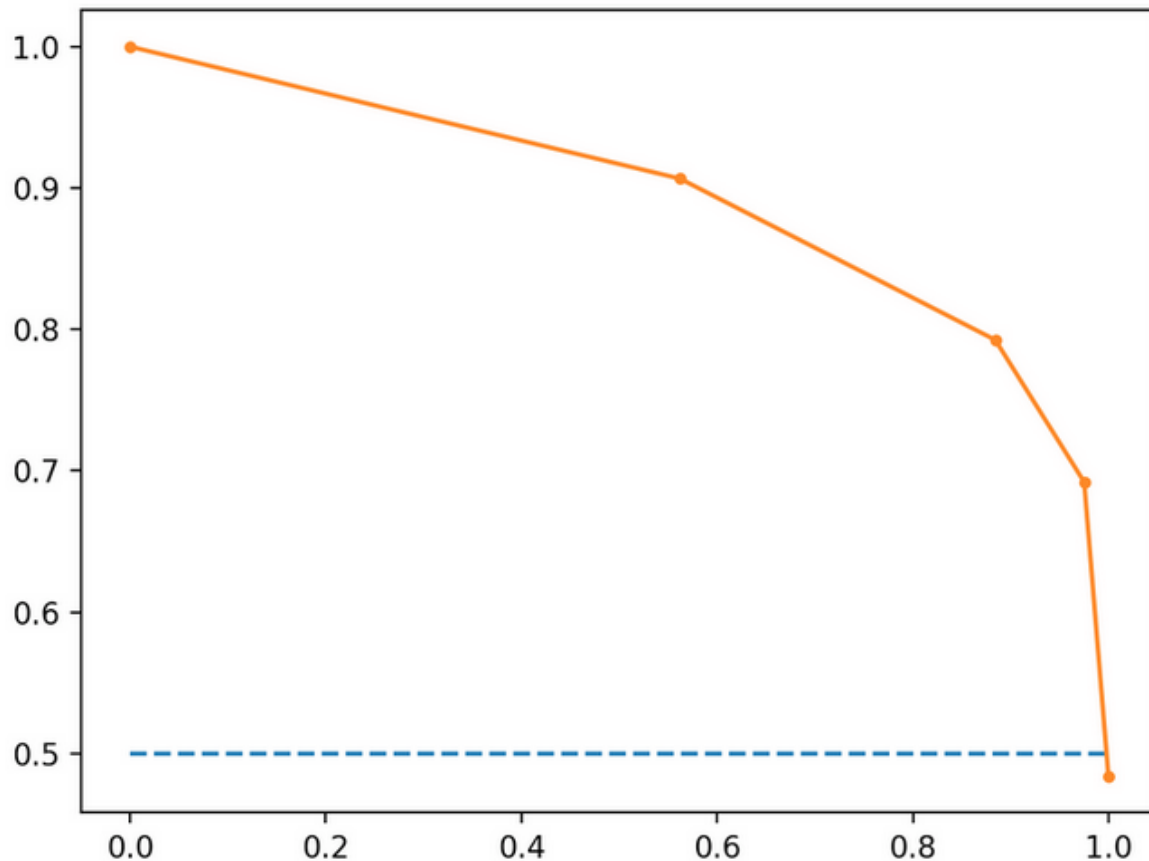
Courbe ROC

Pour un modèle prédictif donné, les courbes ROC (généralement appropriées pour les classifications binaires) illustrent la relation entre le taux de vrais positifs et le taux de faux positifs. La courbe ROC est obtenue en traçant le taux de vrais positifs :

Sensitivity en fonction du taux de faux positifs (1-Spécificité)

Les courbes Précision-Rappel illustrent, en utilisant différents seuils de probabilité, le comportement entre le taux de vrais positifs et la valeur prédictive positive pour un modèle prédictif.

Les courbes ROC sont appropriées lorsque les observations sont équilibrées entre chaque classe, tandis que les courbes Précision-Rappel sont appropriées pour les ensembles de données déséquilibrées.



Exercice : Quiz

[solution n°1 p.17]

Question 1

La courbe ROC est préférablement utilisée pour les classifications de type multi classes, et ne nécessite pas de préparation particulière des données.

- ☐ Vrai
- ☐ Faux

Question 2

La sensibilité est le paramètre que vous cherchez à optimiser dans l'utilisation de test lors de la détection de cas malade pour une épidémie très contagieuse (type Covid).

- ☐ Vrai
- ☐ Faux

Question 3

Lorsque vous voulez que votre modèle réalise des prédictions exactes, en particulier dans le cas de diagnostic de maladies aux traitements avec de nombreux effets indésirables (vrais positifs), vous cherchez à maximiser la spécificité de votre modèle.

- ☐ Vrai
- ☐ Faux

Question 4

Lorsque vous voulez que votre modèle ne manque aucune détection de cas positifs, vous voulez réduire le nombre de faux négatifs.

- ☐ Vrai
- ☐ Faux

Question 5

Dans le cas d'un problème de classification, peu importe le nombre de classes, vous pouvez utiliser l'*accuracy* pour mesurer la performance de vos modèles. C'est la métrique.

- ☐ Vrai
- ☐ Faux

III. Exemple de classification

A. Machine learning : classification

Méthode	Entraînement du modèle
Le modèle de classification multi classes est entraîné sur le jeu de données iris. Ce jeu de données possède 3 classes.	
<pre> 1 # imports 2 import sklearn # import de sklearn 3 from sklearn.datasets import load_iris # chargement du jeu de données iris 4 from sklearn.model_selection import train_test_split # separation train_test 5 import pandas as pd # tableur 6 from sklearn.preprocessing import StandardScaler # standardisation 7 from sklearn.linear_model import Perceptron # modèle 8 from sklearn.metrics import classification_report # rapport de classification 9 from sklearn.metrics import confusion_matrix # matrice de confusion 10 from sklearn.metrics import accuracy_score # evaluation de la précision (exactitude des précisions) 11 import seaborn as sns # outil de visualisation 12 13 # chargement des données 14 data = load_iris() 15 16 # description du jeu de données 17 print(data['DESCR']) 18 19 # données cibles (étiquette) 20 print(data["target"]) 21 22 # on prépare les données en créant un dataframe approprié 23 iris_ds = pd.DataFrame(data["data"], columns = data['feature_names']) 24 iris_ds["target"] = data["target"] 25 iris_ds.head() 26 27 # creation du jeu d'entraînement de validation et de test 28 data_train, data_test = train_test_split(iris_ds) 29 30 # outil pour standardiser le jeu de données 31 sc = StandardScaler() 32 sc.fit(data_train) </pre>	


```

33
34 # transformer les données (caractéristiques)
35 data_train_std = sc.transform(data_train)
36 data_test_std = sc.transform(data_test)
37 y_train = data_train["target"]
38 y_test = data_test["target"]
39
40 # Le perceptron est un modèle de prédiction dans scikit learn
41 ppn = Perceptron(n_iter_no_change=40, eta0=0.1, random_state=0)
42 ppn.fit(data_train_std, y_train)

```

Description du jeu de données

Vous pouvez accéder aux détails du jeu de données organisés comme ci-après.

Vous y retrouvez les informations sur les variables (pétales, sépales et la classe).

```

.. _iris_dataset:

Iris plants dataset
-----

**Data Set Characteristics:**

:Number of Instances: 150 (50 in each of three classes)
:Number of Attributes: 4 numeric, predictive attributes and the class
:Attribute Information:
  - sepal length in cm
  - sepal width in cm
  - petal length in cm
  - petal width in cm
  - class:
    - Iris-Setosa
    - Iris-Versicolour
    - Iris-Virginica

:Summary Statistics:

=====  =====  =====  =====  =====
              Min    Max    Mean     SD    Class Correlation
=====  =====  =====  =====  =====
sepal length:  4.3    7.9    5.84    0.83     0.7826
sepal width:   2.0    4.4    3.05    0.43    -0.4194
petal length:  1.0    6.9    3.76    1.76     0.9490 (high!)
petal width:   0.1    2.5    1.20    0.76     0.9565 (high!)
=====  =====  =====  =====  =====

:Missing Attribute Values: None
:Class Distribution: 33.3% for each of 3 classes.
:Creator: R.A. Fisher
:Donor: Michael Marshall (MARSHALL@PLU@io.arc.nasa.gov)
:Date: July, 1988

The famous Iris database, first used by Sir R.A. Fisher. The dataset is taken
from Fisher's paper. Note that it's the same as in R, but not as in the UCI
Machine Learning Repository, which has two wrong data points.

```

Le jeu de données iris se présente comme ci-dessous :

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	0
1	4.9	3	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5	3.6	1.4	0.2	0

Méthode Évaluation du modèle

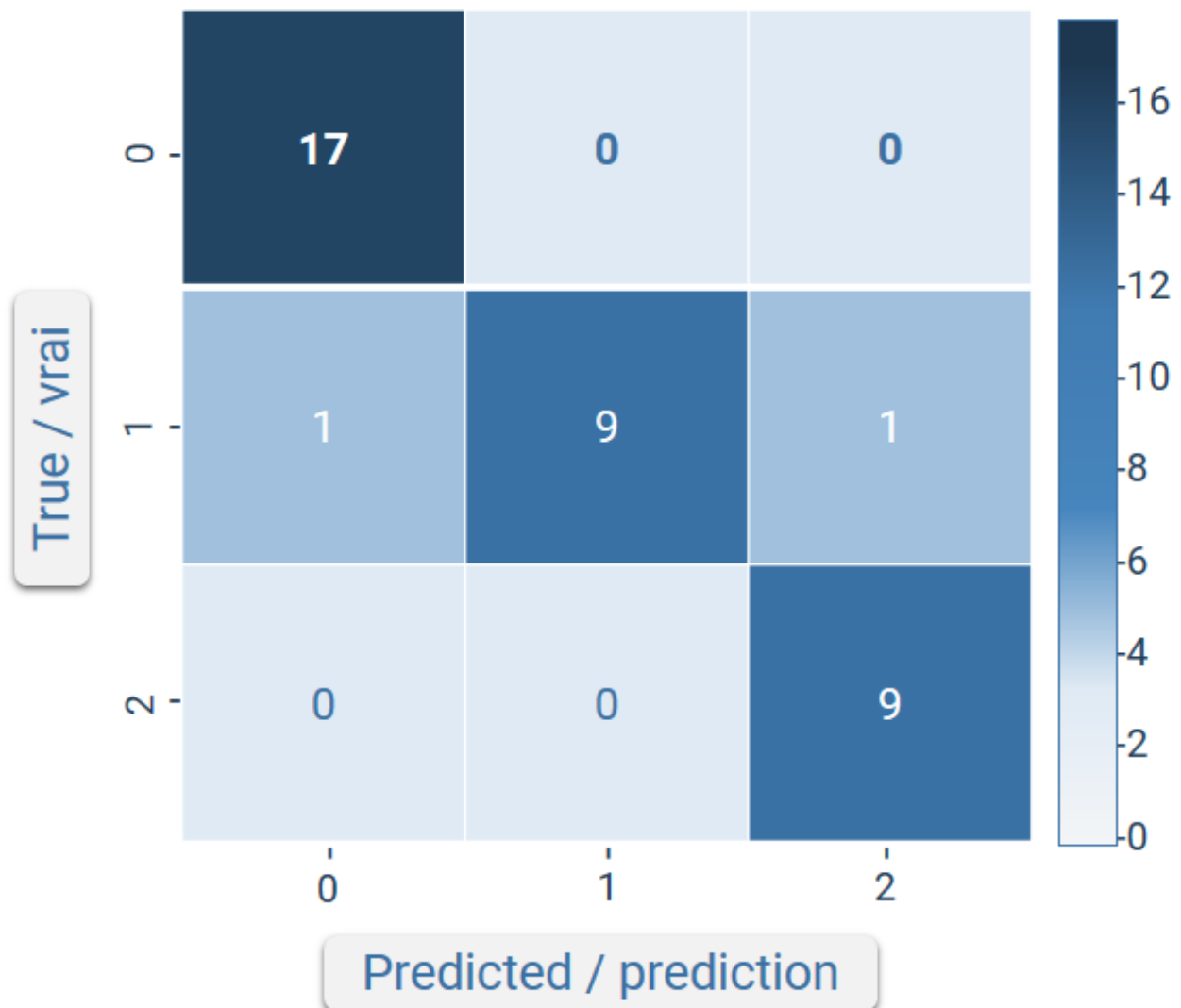
Vous réalisez l'évaluation du modèle sur le jeu de test que vous avez défini auparavant :

```
1 # prédictions (mauvaises classifications)
2 y_pred = ppn.predict(data_test_std)
3 print('Misclassified samples/Mauvaises classifications: %d' % (y_test != y_pred).sum())
4
5 # réalisation de la matrice de confusion
6 print('Accuracy/Exactitude des précisions: %.2f' % accuracy_score(y_test, y_pred))
```

Méthode Analyse du rapport de classification

```
1 # affichage du rapport de classification
2 print(classification_report(y_test, y_pred, target_names=data['target_names']))
3
4 class_map = dict(zip(list(range(3)), data['target_names']))
5 _y_pred = [class_map[el] for el in y_pred]
6 _y_test = [class_map[el] for el in y_test]
7
8 cf_matrix = confusion_matrix(_y_test, _y_pred, labels=data['target_names'])
9
10
11
12 # améliorations cosmétiques de la matrice de confusion
13 sns.heatmap(cf_matrix, annot=True,
14             fmt='', cmap='Blues')
15 plt.xlabel('predicted/prediction', fontsize=10)
16 plt.ylabel('true/vrai', fontsize=10)
```

Le code ci-dessus vous permet de générer la matrice de confusion :



Vous observez un grand nombre de prédictions correctes ou les valeurs prédites sont identiques aux valeurs de référence.

Pour un modèle de classification, vous voulez maximiser les nombres sur la diagonale. Tout nombre à droite ou à gauche est en réalité une mauvaise classification.

Vous générez également le rapport de classification :

	precision	recall	f1-score	support
setosa	0.94	1.00	0.97	17
versicolor	0.90	0.82	0.86	11
virginica	0.90	0.90	0.90	10
accuracy			0.92	38
macro avg	0.91	0.91	0.91	38
weighted avg	0.92	0.92	0.92	38

Vous retrouvez dans ce rapport de classification les métriques (précision, recall, f1-score et support).

La précision mesure le taux de vrais positifs parmi les détections :

Précision = Vrais Positifs / (vrais positifs + faux positifs)

Le recall définit par taux de vrais positifs retrouvés :

Recall = Vrais Positifs / (vrais positifs + faux négatifs)

Le support est défini par le nombre d'occurrences ou d'observations d'une classe donnée.

Ici, le jeu de données est légèrement déséquilibré en faveur de setosa (17 observations) contre 10 observations pour virginica.

Le f1-score est calculé selon la formule suivante :

$f1\text{-score} = 2 * (\text{Précision} * \text{Recall}) / (\text{Précision} + \text{Recall})$

Ce score tient compte de la spécificité et de la sensibilité et, par extension, du nombre de faux négatifs et de faux positifs.

Enfin, on distingue la **macro avg** (moyenne arithmétique) de la **weighted avg** (moyenne pondérée). La moyenne pondérée tient compte des déséquilibres de classes. Pour donner une valeur estimative de la performance de votre modèle de classification, la moyenne pondérée est vivement conseillée afin de ne pas, le cas échéant, surestimer le pouvoir prédictif de votre modèle.

Exercice : Quiz

[solution n°2 p.18]

Question 1

La précision est identique au recall.

- ☐ Vrai
- ☐ Faux

Question 2

Le f-1 score tient compte des faux positifs et des faux négatifs.

- ☐ Vrai
- ☐ Faux

Question 3

Il n'existe pas de métrique qui prend en compte sensibilité et spécificité.

- ☐ Vrai
- ☐ Faux

Question 4

Plus le recall est élevé, plus le taux de faux négatifs est élevé.

- ☐ Vrai
- ☐ Faux

Question 5

Plus la précision est élevée, plus le recall est bas.

- ☐ Vrai
- ☐ Faux

V. Essentiel

Un rapport de classification est utilisé pour mesurer la qualité des prédictions à partir d'un algorithme de classification. Combien de prédictions sont vraies et combien sont fausses ? Plus précisément, les vrais positifs, les faux positifs, les vrais négatifs et les faux négatifs sont utilisés pour prédire les mesures d'un rapport de classification.

- TN / True Negative ou VN / Vrais Négatifs : lorsqu'un cas était négatif et prédit négatif,
- TP / True Positive ou VP / Vrais Positifs : lorsqu'un cas était positif et prédit positif,
- FN / False Negative ou FN / Faux Négatifs : lorsqu'un cas était positif, mais prédit négatif,
- FP / False Positive ou FP / Faux Positifs : lorsqu'un cas était négatif, mais prédit positif,

On définit alors la capacité à détecter des cas positifs :

Sensibilité = $TP / (TP + FN)$

On définit aussi le caractère spécifique dans la classification des cas positifs :

Spécificité = $TP / (TP + FP)$

Dans le cas d'une classification à classes, on utilise une matrice de confusion qui permet de voir les erreurs de classification. Dans le cas d'une classification binaire, on utilise la courbe ROC et la matrice de confusion.

VI. Auto-évaluation

A. Exercice

Vous allez travailler avec le jeu de données suivant : Jeu de données OpenML¹

Il représente l'ensemble de données du centre de service de transfusion sanguine tirées du centre de services de transfusion sanguine de la ville de Hsin-Chu à Taïwan. Il s'agit d'un problème de classification.

Pour démontrer la pertinence du modèle prédictif, cette étude a adopté la base de données des donneurs du centre de service de transfusion sanguine de la ville de Hsin-Chu à Taïwan. Pour construire un modèle prédictif, les scientifiques ont sélectionné 748 donneurs au hasard dans la base de données des donneurs.

Informations sur les attributs

¹ <https://www.openml.org/d/1464>

- V1 : Récence - mois depuis le dernier don
- V2 : Fréquence - nombre total de dons
- V3 : Quantité - sang total donné en c. c.
- V4 : Temps - mois depuis le premier don est une variable binaire représentant s'il/elle a donné du sang en mars 2007 (1 signifie donner du sang ; 0 signifie ne pas donner de sang).

L'attribut cible est une variable binaire représentant si le donneur a donné du sang en mars 2007 (2 signifie donner du sang ; 1 signifie ne pas donner de sang).

Question 1

[solution n°3 p.19]

Comment récupérez-vous le jeu de données dont l'ID est 1464 sur le site OpenML : OpenML¹ ? Pouvez-vous écrire une fonction qui permet de récupérer les caractéristiques et la variable cible ?

Question 2

[solution n°4 p.19]

Comment entraînez-vous un modèle de classification de type Régression Logistique sur ce jeu de données ?

Question 3

[solution n°5 p.19]

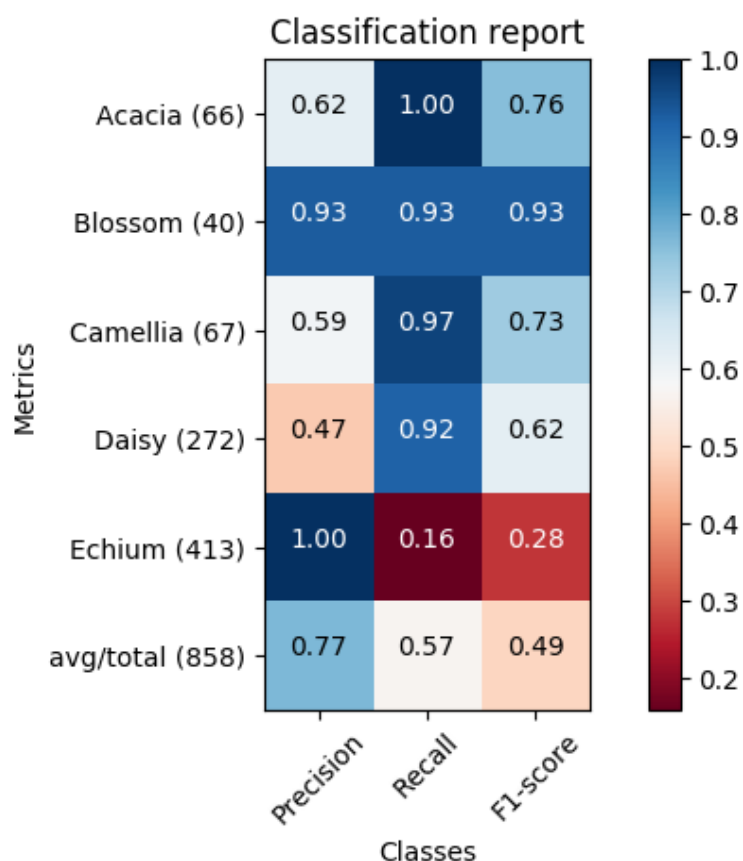
Comment évaluez-vous la performance du modèle ? Quels outils utilisez-vous ?

B. Test

Exercice 1 : Test

[solution n°6 p.22]

Voici le résultat d'une classification :



Question 1

¹ <https://www.openml.org/d/1464>

Pour une classe donnée, plus le modèle possède une Précision élevée (au sens Précision vs Recall), plus le taux de bonnes classifications pour cette classe est élevé.

- ☐ Vrai
- ☐ Faux

Question 2

Ce modèle est parfait pour détecter des Echium, car il a peu de faux négatifs.

- ☐ Vrai
- ☐ Faux

Question 3

Dans le diagramme, les métriques précision, recall et f1-score sont des métriques équivalentes.

- ☐ Vrai
- ☐ Faux

Question 4

Pour la classe Blossom, il y a une erreur, car il n'est pas possible d'avoir un recall et une précision identique.

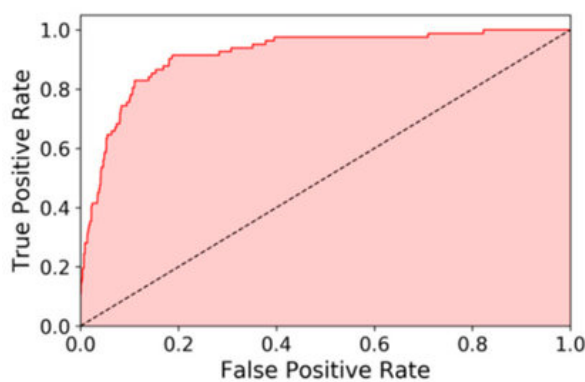
- ☐ Vrai
- ☐ Faux

Question 5

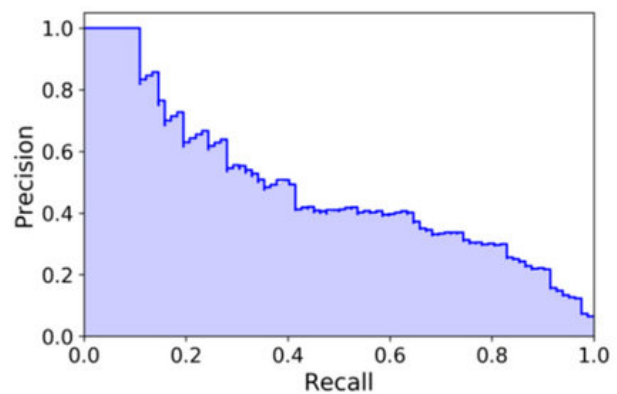
Le f1-score n'est pas sensible aux faux négatifs.

- ☐ Vrai
- ☐ Faux

Question 6



(a)



(b)

- La courbe de gauche est une courbe ROC
(taux de vrais positifs en fonction de 1- (Taux vrais négatifs))
- La courbe de droite est une courbe Precision-Recall

- ☐ Vrai
- ☐ Faux


Solutions des exercices

Exercice p. 7 Solution n°1**Question 1**

La courbe ROC est préférablement utilisée pour les classifications de type multi classes, et ne nécessite pas de préparation particulière des données.

☐ Vrai

☒ Faux


 Dans le cas de l'utilisation d'une courbe ROC, vous comparez 2 classes. Dans le cas d'un problème multi classes, vous devrez donc convertir le problème en autant de courbes ROC qu'il y a de classes (1 classe vs le reste). En pratique, on utilise la courbe ROC pour des problèmes de classification binaire.

Question 2

La sensibilité est le paramètre que vous cherchez à optimiser dans l'utilisation de test lors de la détection de cas malade pour une épidémie très contagieuse (type Covid).

☒ Vrai

☐ Faux


 Dans des problèmes de diagnostic, vous voulez éviter à tout prix les faux négatifs. Vous voulez donc augmenter la Sensibilité : $TP / (TP + FN)$.

Question 3

Lorsque vous voulez que votre modèle réalise des prédictions exactes, en particulier dans le cas de diagnostic de maladies aux traitements avec de nombreux effets indésirables (vrais positifs), vous cherchez à maximiser la spécificité de votre modèle.

☒ Vrai

☐ Faux


 La spécificité ou le vrai taux de négatifs est donné par la formule : $Sp = TN / (TN + FP)$. Plus les FP sont faibles et plus la Spécificité augmente.

Question 4

Lorsque vous voulez que votre modèle ne manque aucune détection de cas positifs, vous voulez réduire le nombre de faux négatifs.

☒ Vrai

☐ Faux

 Vous voulez réduire la possibilité de faux négatifs, quitte à augmenter le nombre de faux positifs. C'est le cas, par exemple, pour certains tests de dépistage.

Question 5

Dans le cas d'un problème de classification, peu importe le nombre de classes, vous pouvez utiliser l'*accuracy* pour mesurer la performance de vos modèles. C'est la métrique.

☐ Vrai

☒ Faux

- Q L'exactitude ou précision en français est une métrique que vous pouvez utiliser pour évaluer votre classification, lorsque les classes sont équilibrées. En pratique, on recommande d'utiliser une matrice de confusion.

Exercice p. 12 Solution n°2

Question 1

La précision est identique au recall.

☐ Vrai

☒ Faux

- Q Précision = $TP / (TP + FP)$, parmi les détections positives lesquelles étaient vraiment positives ?

Recall = $TP / (TP + FN)$, parmi toutes les observations des données positives, combien d'observations ont été retrouvées par le modèle ?

Question 2

Le f-1 score tient compte des faux positifs et des faux négatifs.

☒ Vrai

☐ Faux

- Q C'est une mesure qui permet de prendre en compte les faux positifs et les faux négatifs. Elle est cependant parfois mal comprise de la part du client de la solution d'IA.

Question 3

Il n'existe pas de métrique qui prend en compte sensibilité et spécificité.

☐ Vrai

☒ Faux

- Q Le f-1 score, le f-2 score et bien d'autres mesures prennent en compte les faux positifs et faux négatifs.

Question 4

Plus le recall est élevé, plus le taux de faux négatifs est élevé.

☐ Vrai

☒ Faux

- Q Plus le recall est élevé, moins le nombre de faux négatifs est élevé. Ces deux grandeurs évoluent de manière inversée.

Question 5

Plus la précision est élevée, plus le recall est bas.

☒ Vrai

☐ Faux

- Q Idéalement, un modèle parfait maintient un niveau de précision à 1, quel que soit le niveau de recall. En pratique cependant, la précision chute avec le recall.

p. 14 Solution n°3

Vous utilisez le code ci-dessous afin de récupérer les données (`fetch_openml`)

```

1 from sklearn.datasets import fetch_openml
2 from sklearn.preprocessing import StandardScaler
3 from sklearn.pipeline import make_pipeline
4 from sklearn.linear_model import LogisticRegression
5 from sklearn.model_selection import train_test_split
6
7 def exercice_classification(id=1464):
8
9
10
11     # Vous récupérez le jeu de données dont l'identifiant est 1464 sur openml
12     X, y = fetch_openml(data_id=id, return_X_y=True)
13     return(X,y)
14
15 X_data, y_data = exercice_classification(id=1464)

```

p. 14 Solution n°4

Vous utilisez `StandardScaler`, `make_pipeline`, `logisticregression` pour créer votre modèle. `StandardScaler` est utilisé pour normaliser vos données. `Logistic Regression` est un modèle de classification. On stratifie les données selon la valeur de `y`.

```

1 from sklearn.datasets import fetch_openml
2 from sklearn.preprocessing import StandardScaler
3 from sklearn.pipeline import make_pipeline
4 from sklearn.linear_model import LogisticRegression
5 from sklearn.model_selection import train_test_split
6
7 X, y = fetch_openml(data_id=1464, return_X_y=True)
8 X_train, X_test, y_train, y_test = train_test_split(X, y, stratify=y)
9
10 clf = make_pipeline(StandardScaler(), LogisticRegression(random_state=0))
11 clf.fit(X_train, y_train)

```

p. 14 Solution n°5

On utilise la matrice de confusion et le rapport de classification que l'on peut créer avec les commandes ci-dessous.

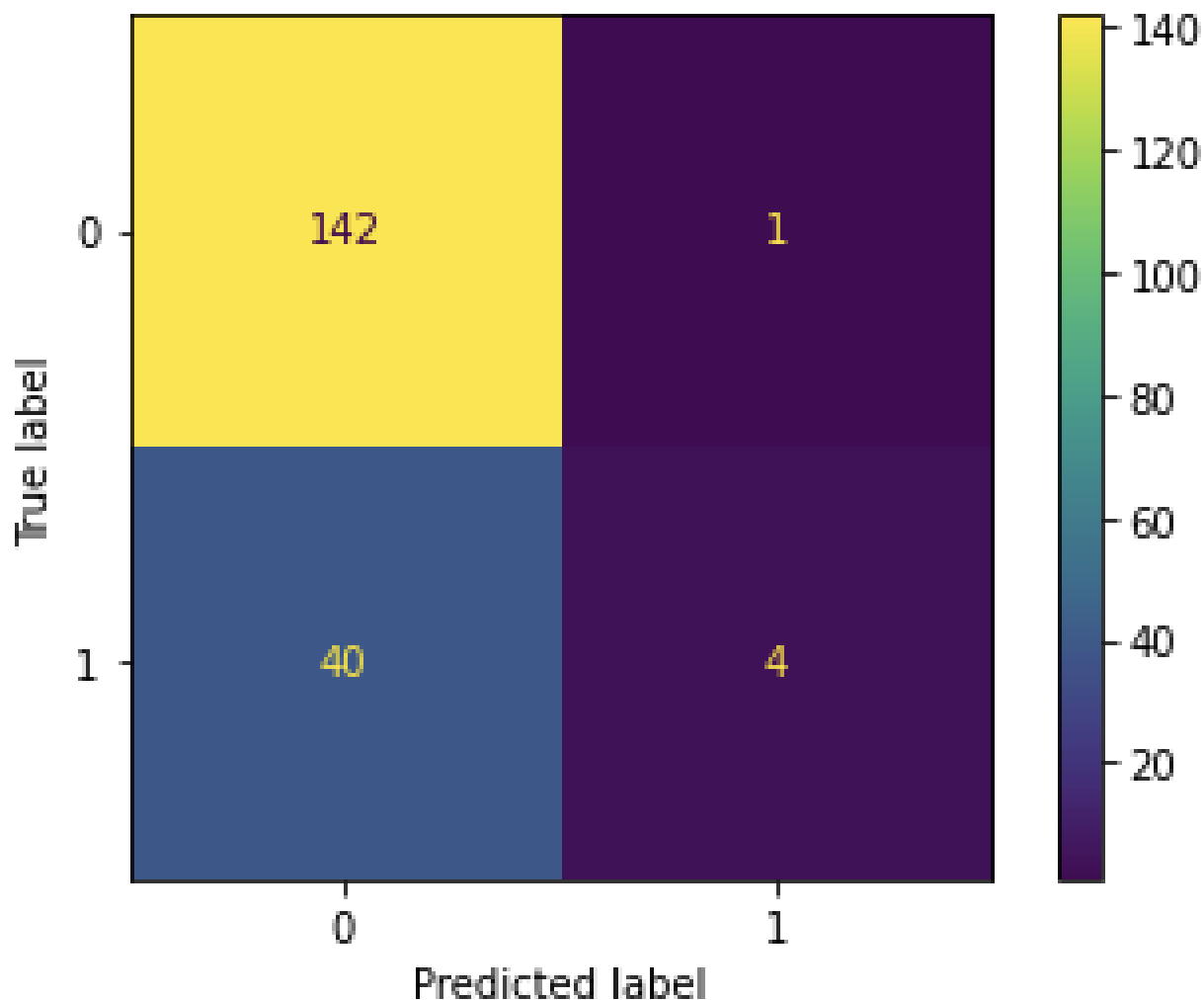
Pour évaluer la performance du modèle vous utiliserez la matrice de confusion et la courbe Précision-recall.

```

1 # Matrice de confusion
2 from sklearn.metrics import confusion_matrix
3 from sklearn.metrics import ConfusionMatrixDisplay
4
5
6 y_pred = clf.predict(X_test)
7 cm = confusion_matrix(y_test, y_pred)
8
9 cm_display = ConfusionMatrixDisplay(cm).plot()

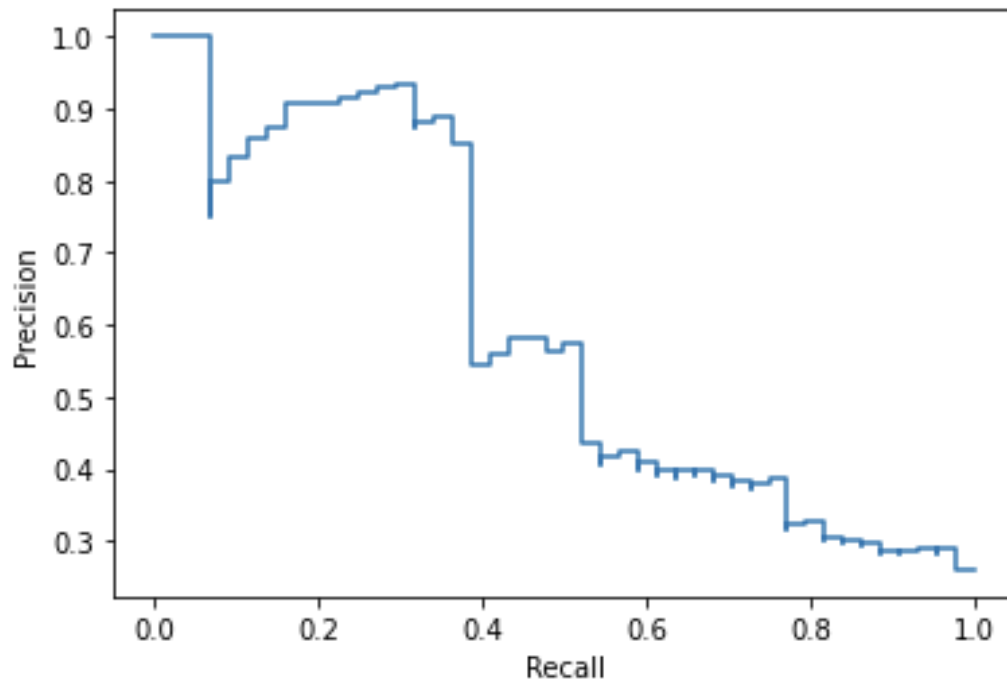
```

Matrice de confusion :



```
1 # Courbe PR-RC
2 from sklearn.metrics import precision_recall_curve
3 from sklearn.metrics import PrecisionRecallDisplay
4
5 prec, recall, _ = precision_recall_curve(y_test, y_score, pos_label=clf.classes_[1])
6 pr_display = PrecisionRecallDisplay(precision=prec, recall=recall).plot()
```

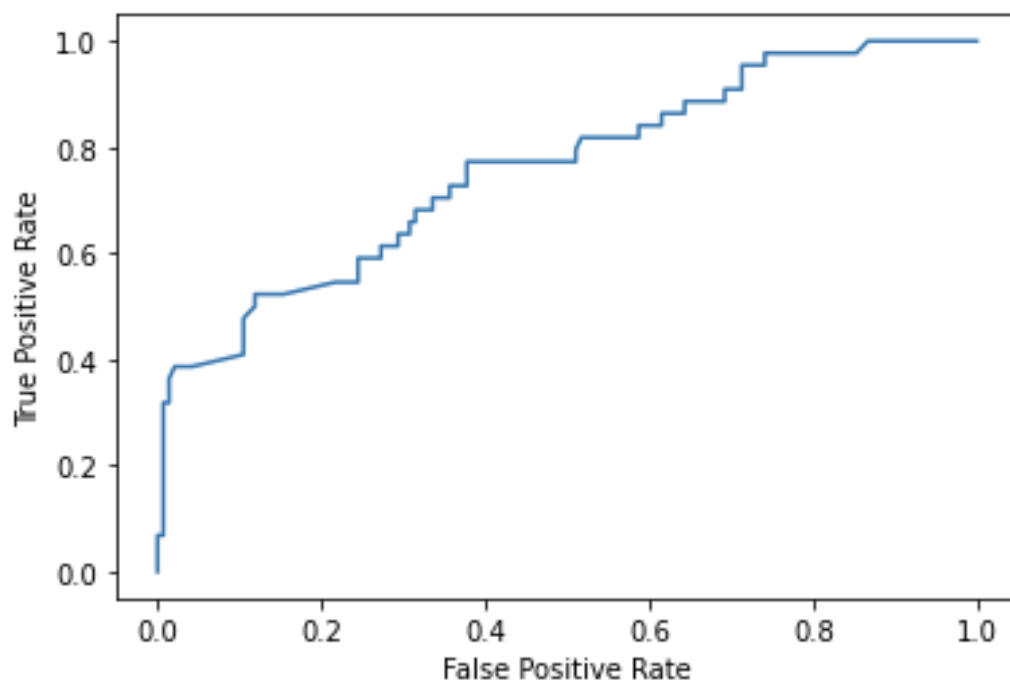
Courbe Précision-Recall :



```

1 # Courbe ROC
2 from sklearn.metrics import roc_curve
3 from sklearn.metrics import RocCurveDisplay
4 y_score = clf.decision_function(X_test)
5 # On extrait les taux de faux positifs et de vrais positifs
6 fpr, tpr, _ = roc_curve(y_test, y_score, pos_label=clf.classes_[1])
7 roc_display = RocCurveDisplay(fpr=fpr, tpr=tpr).plot()

```




Exercice p. 14 Solution n°6

Question 1

Pour une classe donnée, plus le modèle possède une Précision élevée (au sens Précision vs Recall), plus le taux de bonnes classifications pour cette classe est élevé.

☒ Vrai

☐ Faux


 Une grande précision implique un faible taux de faux positifs. La formule de la Précision est inversement proportionnelle au nombre de faux positifs.

Question 2

Ce modèle est parfait pour détecter des Echium, car il a peu de faux négatifs.

☐ Vrai

☒ Faux


 Pour détecter, on ne doit pas pouvoir avoir des faux négatifs. Le recall étant bas, mais la précision étant maximale, on peut utiliser ce modèle : s'il prédit que la fleur est du type Echium, elle l'est très certainement.

Question 3

Dans le diagramme, les métriques précision, recall et f1-score sont des métriques équivalentes.

☐ Vrai

☒ Faux


 Ces trois mesures sont toutes les 3 différentes, la précision étant sensible aux faux positifs, le recall aux faux négatifs et le f1-score qui est sensible aux deux.

Question 4

Pour la classe Blossom, il y a une erreur, car il n'est pas possible d'avoir un recall et une précision identique.

☐ Vrai

☒ Faux


 Rien ne stipule que précision et recall doivent être systématiquement différents.

Question 5

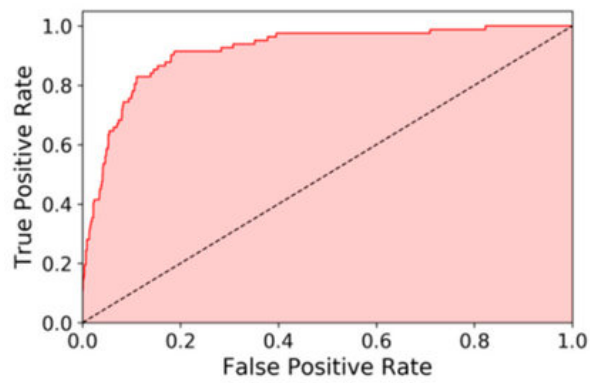
Le f1-score n'est pas sensible aux faux négatifs.

☐ Vrai

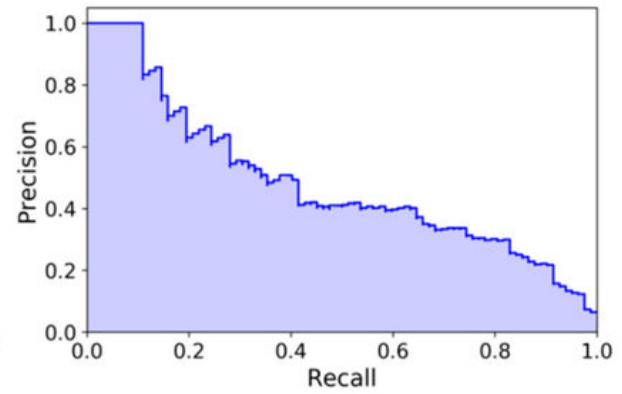
☒ Faux

 Le f1-score est sensible aux faux positifs et aux faux négatifs.

Question 6



(a)



(b)

- La courbe de gauche est une courbe ROC
(taux de vrais positifs en fonction de 1- (Taux vrais négatifs))
- La courbe de droite est une courbe Precision-Recall

☒ Vrai

☐ Faux



La courbe ROC est donnée par le TPR en fonction de FPR. La courbe de PR est celle à droite.