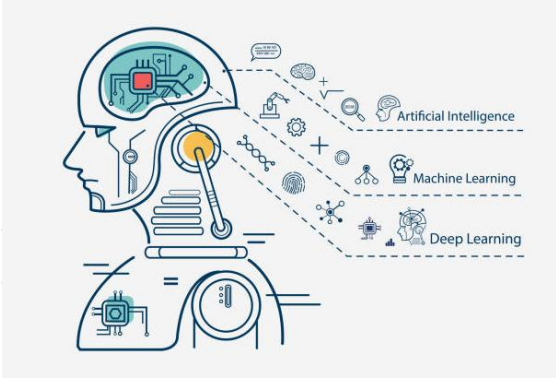
Introduction au Machine Learning

Mise en forme des données

Nom :	**Note : / 20**
	Classe :

Résumé :

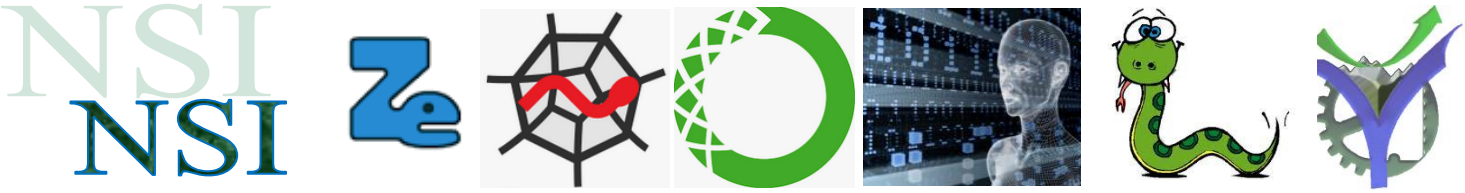
Nous examinons dans ce document quelques étapes préalables à l'utilisation de données dans un cadre de Machine Learning. Puis quelques outils permettant la mesure des performances de ces algorithmes de classification.



(1)

Sommaire[1]

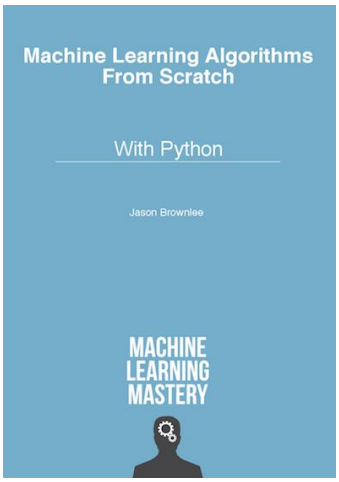
[1] Illustration empruntée au site : https://github.com/devAmoghS/Machine-Learning-with-Python



Remerciements

Un remerciement spécial à M. Jason Brownlee qui nous a donné l'autorisation d'utiliser ces travaux pour ces cours. En particulier son ouvrage disponible à l'adresse suivante :

https://machinelearningmastery.com/machine-learning-algorithms-from-scratch/



Dictionnaire anglais - français

standard deviation = écart-type
contrive = inventer
accuracy = précision
Mean Absolute Error MEA = erreur absolue moyenne
Root Mean Squared Error RMSE = Erreur moyenne quadratique
stream = flux

1 Introduction au machine learning

Pour débuter notre parcours visualisez la présentation visible sur le lien ci-dessous

https://www.youtube.com/watch?v=BLZo9QLt0UY

Puis répondre aux questions suivantes :

Q1. Que signifie le mot data ?

Q2. Indiquez les trois types de données utilisées dans le machine learning.

Q3. C'est quoi le clickstream ?

Q4. Citez les trois parties principales d'un projet de Machine Learning.

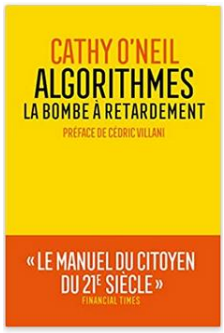
Q5. Donner la définition d'un modèle.

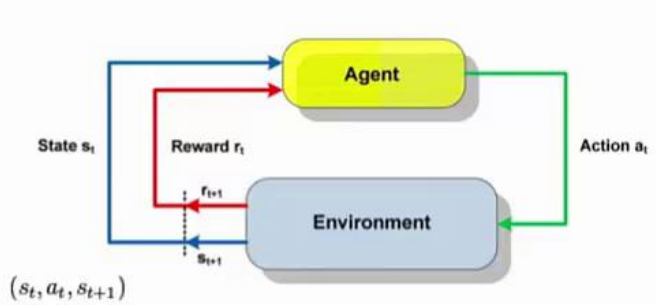
Q6. Définir l'apprentissage supervisé.

Q7. Donner quelques exemples de Machine Learning supervisé :

Q8. Quelle est la caractéristique principale de l'apprentissage non supervisé ?

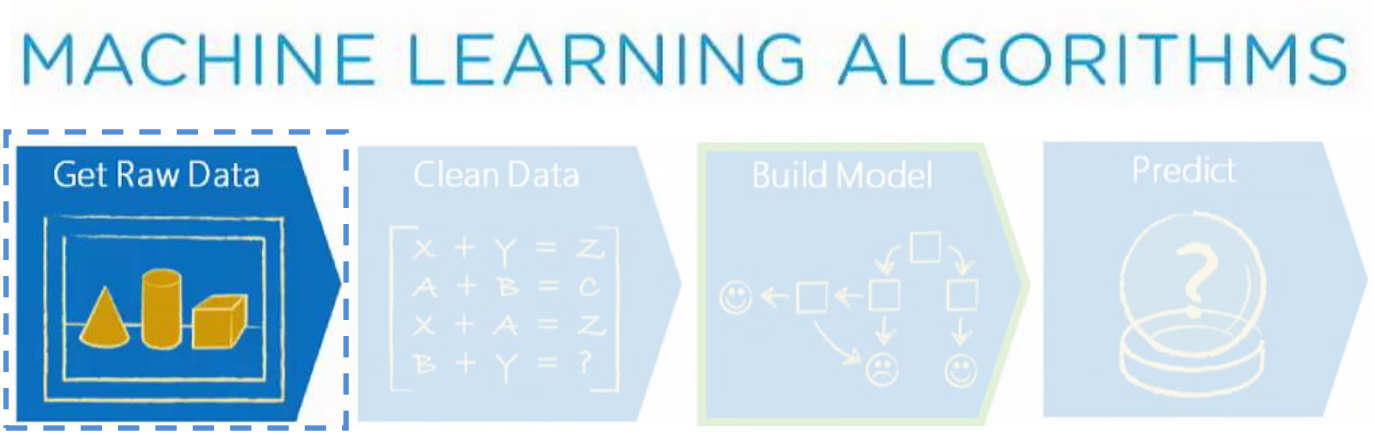
Q9. L'apprentissage par réenforcement (21'39'') : Après avoir vu la vidéo en complément[2] sensibilisant sur la conception des algorithmes avec réenforcement indiquez quelques dangers potentiels dans la conception et l'utilisation de ces algorithmes.





[2] https://www.youtube.com/watch?v=5BzJSvX6nXA

2 Chargement des données



2.1 Première étape : la lecture des données brutes.

Les données sont ici présentes dans des fichiers de type CSV. Étudions un premier script très simple de lecture de ces fichiers : load_csv_essai_0.py

Q10. Utilisez le script avec le fichier Fichier_A.csv puis avec un fichier Toto.csv inexistant. Comparez les résultats obtenus dans les deux cas.

Reprendre la même étude avec le script load_csv_essai_1.py

Q11. Que constatez-vous maintenant ?

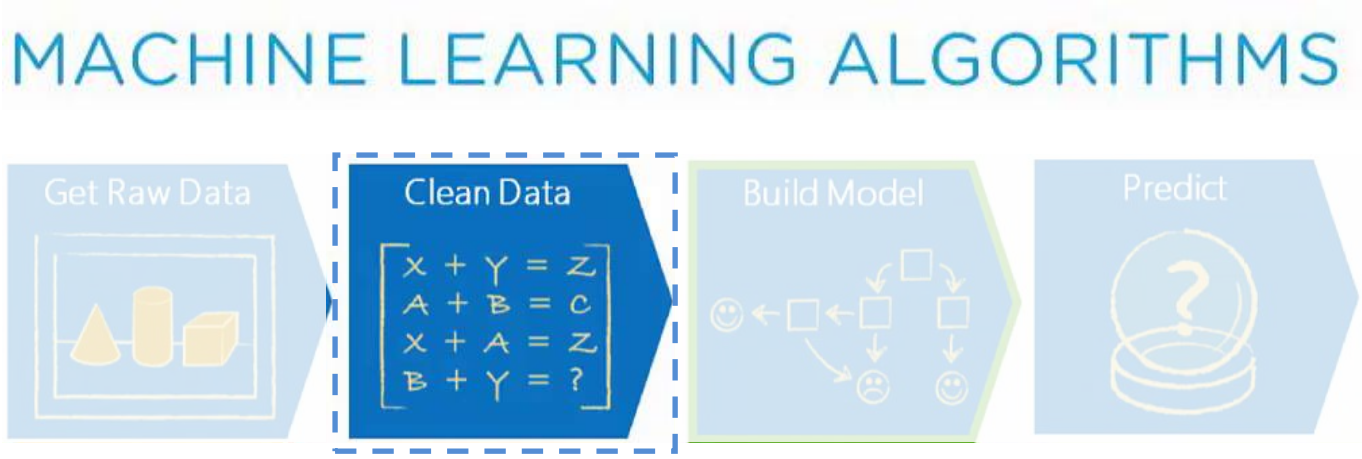
Vous venez d'utiliser la gestion des exceptions permise par Python. Elle permet 'd'attraper' des erreurs provoquées par l'exécution d'un script et de les gérer 'proprement'.

Ressources :

Explication du principe de fonctionnement : w3schools.com
https://www.w3schools.com/python/python_try_except.asp

Compléments avec la liste des exceptions built-in :
http://sdz.tdct.org/sdz/les-exceptions-9.html

2.2 Nettoyage des données

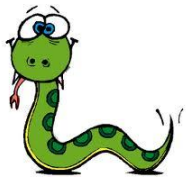


Q12. Vous projetez de réaliser une analyse sur des données envoyées par votre chef de service. C'est urgent il n'a pas eu le temps de les nettoyer. Curieux vous examinez ces données Fichier_B.csv avec votre éditeur de textes préféré.

Combien d'anomalies constatez-vous ?

```
1  Feature A;Feature B;Feature C;Feature D
2  1;1,0;un;A
3  2;2,0;deux;B
4  3;3,0;;C
5  4;4,0;quatre;C
6  5;5,0;cinq;A
7  6;6,0;six;
8  7;7,0;sept;B
9  8;8,0;huit;B
10 9;9,0;neuf;C
11
12 11;11,0;onze;C
13 12;12,0;douze;A
14 13;13,0;treize;B
15 14;14,0;quatorze;A
16 15;15,0;quinze;B
17 16;16,0;seize;B
18 17;17,0;dixsept;A
19 18;10000,0;dixhuit;A
20 19;19,0;dixneuf;C
21 20;20,0;vingt;C
```

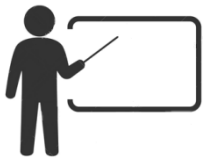



Q13. Comment avez-vous procédé pour répondre à la question précédente ? Quels ont été vos critères pour nettoyer les données ?

Bon procédons au nettoyage de ces données :

Script_Machine_Learning_1. Modifiez le script load_csv_essai_1.py pour réaliser le nettoyage des données. On ne conservera que des groupes complets contenant donc des valeurs pour les features A, B, C et D.

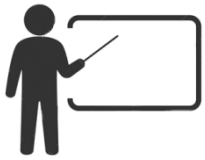


Q14. A ce stade toutes les anomalies ont-elles été traitées ?

2.3 Conversion des valeurs numériques

Les valeurs numériques fournies dans les data : feature A et B sont présentes dans le fichier csv sous un format texte. Il faut donc les convertir. Pour rappel les fonctions *int('chaine')* et *float('chaine')* réalisent les conversions correctement d'une chaine vers un entier ou un flottant, sous condition que la chaine de caractère ai un format correct.

Script_Machine_Learning_2. Complétez votre script précédent en convertissant les deux premiers champs de valeurs feature A et feature B en valeurs numériques respectivement entières et flottants.



3 Mise à l'échelle des données

Pour illustrer cette étape nous allons utiliser le fichier de données : Fichier_C.csv

Fichier_C.csv

```
1  Feature A;Feature B;Feature C
2  1.1;120;A
3  1.5;80;B
4  2.2;65;C
5  1.3;88;A
6  0.8;160;B
7  0.2;42;C
```

Nous observons dans ce fichier deux caractéristiques numériques Feature A et Feature B qui permettent de classifier les données en classes A, B et C.

Les algorithmes de Machine Learning travaillent sur les données numériques, ici les deux premières colonnes. Mais alors :

Q15. Que peut-on dire des valeurs respectives des deux colonnes ?

Q16. A votre avis cela va-t-il poser un problème dans le calcul ?

Pour solutionner ce problème un prétraitement des données est réalisé. Cette mise à l'échelle peut être faite de deux manières différentes : une normalisation ou une standardisation, les deux techniques sont possibles et ont des avantages différents.

3.1 Normalisation

Cette technique fonctionne en recombinant les valeurs avec les minimum et maximum pour chacune des colonnes selon la relation suivante :

$$\text{scaled value} = \frac{value - min}{max - min}$$

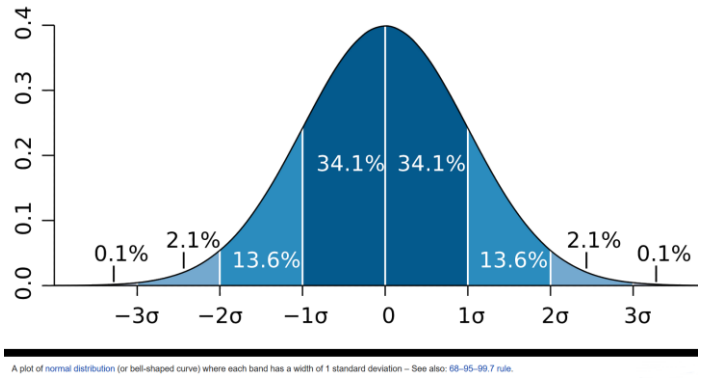
Script_Machine_Learning_3. Réaliser la normalisation des données.

Q17. Après normalisation quelle est l'intervalle des valeurs possibles pour chacune des colonnes ?

Q18. Avons-nous réussi avec cette technique à résoudre les inquiétudes soulevées dans la question 7 ?

3.2 Standardisation

La standardisation des données est une technique qui réparti la distribution des données selon une loi normale centrée réduite. Celle ci est centrée sur l'abscisse 0 et l'écart-type est par construction égal à 1. La représentation appelée courbe de Gauss ou courbe en cloche est donnée ci-dessous :



Interprétation de la courbe en cloche[3] La courbe de Gauss expliquée aux enfants.pdf

Étapes du calcul :

a) Calculer la moyenne pour chaque paramètre :

$$\text{mean} = \frac{\sum_{i=1}^{n} values_i}{count(values)}$$

[3] Pour approfondir les notions mathématiques sous-jacentes voir ici :
https://commentprogresser.com/statistique-distribution-loi-normale.html
http://www.jybaudot.fr/Probas/loinormale.html
http://www.jybaudot.fr/Stats/moyenne.html
http://www.jybaudot.fr/Probas/centreereduite.html
https://www.youtube.com/watch?v=8wjwbCxM7G0
https://www.youtube.com/watch?v=SfVuKV4TrGI

b) Calculer l'écart type (standard deviation) :

$$\text{standard deviation} = \sqrt{\frac{\sum_{i=1}^{n}(value_i - mean)^2}{count(values) - 1}}$$

c) Standardiser les données :

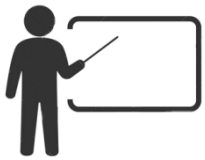
$$\text{standardized\_value}_i = \frac{value_i - mean}{stdev}$$

Mise en œuvre :

La table de données utilisées :

Feature A	Feature B	Feature C
1.10	120.00	A
1.50	80.00	B
2.20	65.00	C
1.30	88.00	A
0.80	160.00	B
0.20	42.00	C

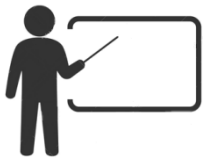
Script_Machine_Learning_4. Calculer la moyenne de chacune des colonnes numériques du fichier : Fichier_C.csv



Résultats attendus :

```
Moyenne colonne   0 :   1.183
Moyenne colonne   1 :  92.500
```

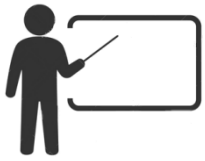
Script_Machine_Learning_5. Calculer l'écart type de chacune des colonnes numériques du fichier.



Résultats attendus :

```
Ecart type colonne   0 :   0.674
Ecart type colonne   1 :  41.942
```

Script_Machine_Learning_6. Réaliser la standardisation des données.

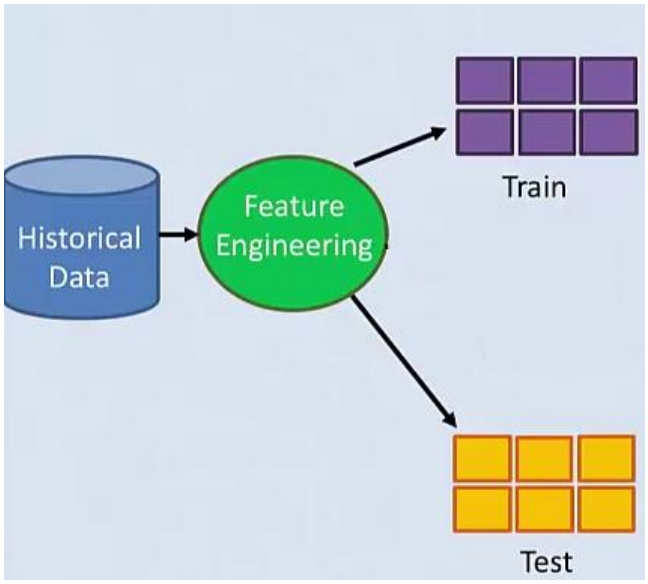


Résultats attendus :

Feature A	Feature B	Feature C
-0.124	0.656	A
0.470	-0.298	B
1.509	-0.656	C
0.173	-0.107	A
-0.569	1.609	B
-1.460	-1.204	C

4 Séparations des données

Pour pouvoir tester les performances d'un algorithme de Machine Learning nous allons séparer le jeu de données en deux groupes, le premier *'train'* servira à trouver les bons paramètres du modèle. Le second *'test'* qui n'a pas servi à construire le modèle sera utilisé pour mesurer les performances de celui-ci.



La proportion entre les deux sous-groupes est ajustée avec un coefficient split. Un script fonctionnel vous est donné : Script_test_split_a_commenter.py

Q19. Analyser le script proposé et compléter les commentaires.

5 Mesure des performances

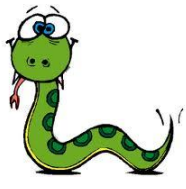
Nous nous intéressons ici à la mesure des performances de nos résultats. Cela est possible puisque noue entrainons nos algorithmes de Machine Learning sur des jeux de donnée déjà classifiés. Il suffit in fine de comparer la sortie obtenue par le calcul avec celle prévue dans le jeu de données et de comparer les deux.

Il est alors possible d'en déduire quatre indicateurs de performances.

5.1 Indicateur de précision

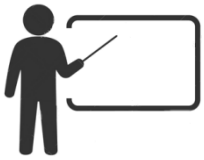
Cet indicateur le plus simple calcul le pourcentage de réponses correctes sur l'ensemble des résultats.






$$\text{accuracy} = \frac{\text{correct predictions}}{\text{total predictions}} \times 100$$

Script_Machine_Learning_7. Compléter la fonction calcul precision du script : Script_Calcul_Precision_A_Completer.py



Un exemple de résultats attendus :

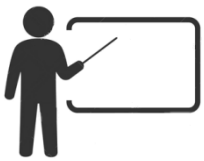
```
Valeurs réelles :  [0, 0, 0, 0, 0, 1, 1, 1, 1, 1]
valeurs_prevues :  [0, 1, 0, 0, 0, 1, 0, 1, 1, 1]
Précision des résultats :  80.0  %
```

5.2 Utilisation de l'erreur absolue moyenne MAE

Lorsque les résultats sont des nombres réels il est plus efficace de travailler avec la valeur absolue de l'écart entre la valeur réelle et la valeur estimée. C'est le Mean Absolute Error ou MAE :

$$MAE = \frac{\sum_{i=1}^{n} abs(predicted_i - actual_i)}{\text{total predictions}}$$

Script_Machine_Learning_8. Compléter le script précédent pour ajouter le calcul de l'erreur moyenne absolue.

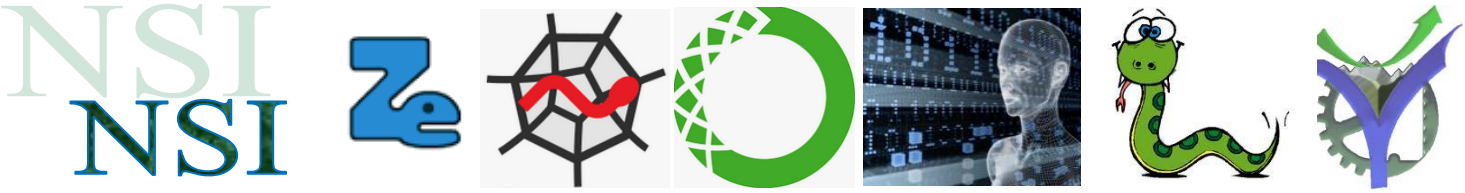


Un exemple de résultats attendus :

```
Données réelles  :  [0.1, 0.2, 0.3, 0.4, 0.5]
Données prédites :  [0.11, 0.19, 0.29, 0.41, 0.5]
Erreur absolue moyenne :  0.0080
```

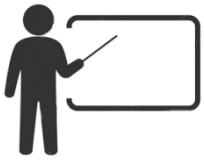
5.3 Erreur quadratique moyenne RMSE

Une autre façon de calculer des erreurs est le calcul Root Mean Squared Error ou calcul de l'erreur quadratique moyenne :



$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(predicted_i - actual_i)^2}{\text{total predictions}}}$$

Script_Machine_Learning_9. Compléter le script précédent pour calculer l'erreur quadratique moyenne.



Un exemple de résultats attendus :

```
Données réelles  :  [0.1, 0.2, 0.3, 0.4, 0.5]
Données prédites :  [0.11, 0.19, 0.29, 0.41, 0.5]
Erreur quadratique moyenne :  0.0089
```

6 Un outil de visualisation des résultats : la matrice de confusion[4]

6.1 Définition

Une Confusion Matrix est un résumé des résultats de prédictions sur un problème de classification. Les prédictions correctes et incorrectes sont mises en lumière et réparties par classe. Les résultats sont ainsi comparés avec les valeurs réelles.

Cette matrice permet de comprendre de quelle façon le modèle de classification est confus lorsqu'il effectue des prédictions. Ceci permet non seulement de savoir quelles sont les erreurs commises, mais surtout le type d'erreurs commises. Les utilisateurs peuvent les analyser pour déterminer quels résultats indiquent comment les erreurs sont commises.

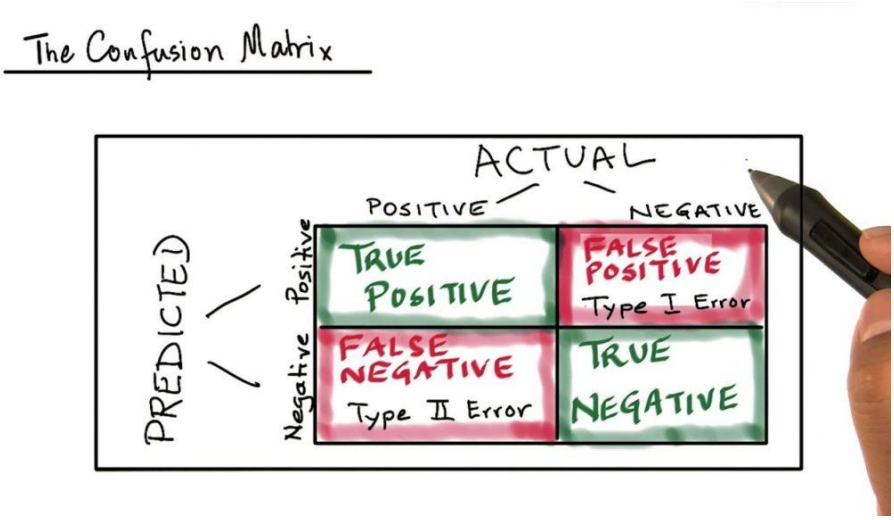
Outre le Machine Learning, les matrices de confusion sont aussi utilisées dans le domaine des statistiques, du Data Mining et de l'intelligence artificielle. De manière générale, elles permettent d'analyser des données statistiques plus rapidement et de rendre les résultats plus simples à déchiffrer via la Data Vizualisation. Elles offrent l'opportunité d'analyser les erreurs dans les statistiques, le forage de données, ou même les examens médicaux.

6.2 Comment calculer une matrice de confusion

Pour calculer une matrice de confusion, il est nécessaire de disposer d'un ensemble de données de test (test dataset) ou d'un ensemble de données de validation (validation dataset) avec les valeurs de résultat attendues. On fait ensuite une prédiction pour chaque ligne du test dataset .

[4] Paragraphe rédigé à partir de https://www.lebigdata.fr/confusion-matrix-definition

A partir des résultats escomptés et des prédictions, la matrice indique le nombre de prédictions correctes pour chaque classe et le nombre de prédictions incorrectes pour chaque classe organisées en fonction de la classe prédite. Chaque ligne du tableau correspond à une classe prédite, et chaque colonne correspond à une classe réelle.



6.3 Interpréter une matrice de confusion

Pour bien comprendre le fonctionnement d'une matrice de confusion, il convient de bien comprendre les quatre terminologies principales : TP, TN, FP et FN. Voici la définition précise de chacun de ces termes :

- ***TP (True Positives) :*** *les cas où la prédiction est positive, et où la valeur réelle est effectivement positive. Exemple : le médecin vous annonce que vous êtes enceinte, et vous êtes bel et bien enceinte.*

- ***TN (True Negatives) :*** *les cas où la prédiction est négative, et où la valeur réelle est effectivement négative. Exemple : le médecin vous annonce que vous n'êtes pas enceinte, et vous n'êtes effectivement pas enceinte.*

- ***FP (False Positive) :*** *les cas où la prédiction est positive, mais où la valeur réelle est négative. Exemple : le médecin vous annonce que vous êtes enceinte, mais vous n'êtes pas enceinte.*

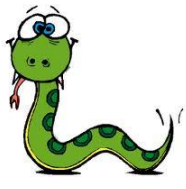
- ***FN (False Negative) :*** *les cas où la prédiction est négative, mais où la valeur réelle est positive. Exemple : le médecin vous annonce que vous n'êtes pas enceinte, mais vous êtes enceinte.*

6.4 Un exemple

Reprenons les résultats calculé précédemment :

```
Valeurs réelles :  [0, 0, 0, 0, 0, 1, 1, 1, 1, 1]
valeurs_prevues :  [0, 1, 0, 0, 0, 1, 0, 1, 1, 1]
```






Nous pouvons extraire les informations suivantes :

Valeur réelle 0 on obtient **prévue 0** : 4 (correct) **prévue 1** : 1 (erroné)

Valeur réelle 1 on obtient **prévue 0** : 1 (erroné) **prévue 1** : 4 (correct)

On peut colorer les résultats pour s'y retrouver plus facilement :

```
Valeurs réelles :  [0, 0, 0, 0, 0, 1, 1, 1, 1, 1]
valeurs_prevues :  [0, 1, 0, 0, 0, 1, 0, 1, 1, 1]
```

On peut tracer la matrice :

		Valeurs réelles 0	Valeurs réelles 1
Valeurs	0	4	1
Prédites	1	1	4

Q20. Donner la matrice de confusion du même exemple de data d'entrée mais dans le cas où il n'y a aucune erreur :

		Valeurs réelles 0	Valeurs réelles 1
Valeurs	0		
Prédites	1		

Q21. Déterminer la matrice de confusion pour le cas ci-dessous :

```
valeurs_relles   = [2,0,0,0,0,1,1,1,1,1,2,2]
valeurs_prevues  = [1,0,1,0,2,1,0,1,1,1,2,2]
```

		Valeurs réelles 0	Valeurs réelles 1	Valeurs réelles 2
Valeurs	0			
Prédites	1			
	2			