

Les algorithmes

Commençons par le code binaire puisque que c'est tout ce qu'un ordinateur comprend

Une information binaire ne peut qu'avoir deux états (ouvert ou fermé)

On le symbolise par 0 ou 1

La semaine dernière

Nous avons parlé abondamment des données.

Lorsque j'écris 9562,

9562, c'est $9000 + 500 + 60 + 2$.

Allons plus loin

9000, c'est 9×1000 , parce que le 9 est le quatrième chiffre en partant de la droite

500, c'est 5×100 , parce que le 5 est le troisième chiffre en partant de la droite

60, c'est 6×10 , parce que le 6 est le deuxième chiffre en partant de la droite

2, c'est 2×1 , parce que le 2 est le premier chiffre en partant de la droite

On peut encore écrire ce même
nombre d'une manière
légèrement différente.

$$9\ 562 = (9 \times 10 \times 10 \times 10) + (5 \times 10 \times 10) + (6 \times 10) + (2)$$

En base 10, ça donne

$$9\ 562 = 9 \times 10^3 + 5 \times 10^2 + 6 \times 10^1 + 2 \times 10^0$$

Pourquoi la base 10?

Les babyloniens avaient en leur temps adopté la base 60 (dite sexagésimale).

60 étant un nombre divisible par beaucoup d'autres (c'est pour cette raison qu'il avait été choisi), on pouvait, rien qu'en regardant le dernier chiffre, savoir si un nombre était divisible par 2, 3, 4, 5, 6, 10, 12, 15, 20 et 30.

Alors qu'en base 10, nous ne pouvons immédiatement répondre à la même question que pour les diviseurs 2 et 5.

La base sexagésimale a certes disparu

Mais Babylone nous a laissé en héritage sa base sexagésimale dans la division du cercle en soixante parties (pour compter le temps en minutes et secondes),

et celle en 6 x 60 parties (pour les degrés de la géométrie et de l'astronomie).

Alors, pourquoi avons-nous
adopté la base décimale ?

Nos doigts !

Les ordinateurs sont en base 2 (binaire)

Les ordinateurs, eux, comme on l'a vu, ont un dispositif physique fait pour stocker (de multiples façons) des informations binaires

Une information binaire

s'appelle un bit (en anglais... bit).

Un groupe de huit bits s'appelle un octet (en anglais, byte)

Cela signifie qu'un octet peut
servir à coder 256 nombres
différents :

ce peut être la série des nombres entiers de 1 à 256, ou de 0 à 255, ou
de -127 à $+128$.

Si l'on veut coder des nombres plus grands que 256, ou des nombres négatifs, ou des nombres décimaux,

on va donc être contraint de mobiliser plus d'un octet

En effet, avec deux octets, on a $256 \times 256 = 65\,536$ possibilités.

En utilisant trois octets, on passe à $256 \times 256 \times 256 = 16\,777\,216$ possibilités

Prenons un octet au hasard :

1 1 0 1 0 0 1 1

D'après les principes vus plus haut, ce nombre représente en base dix, en partant de la gauche :

$$1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 =$$

$$1 \times 128 + 1 \times 64 + 1 \times 16 + 1 \times 2 + 1 \times 1 =$$

$$128 + 64 + 16 + 2 + 1 =$$

$$211$$

Inversement, comment traduire un nombre décimal en codage binaire ?

Prenons, par exemple, 186.

Dans 186, on trouve 1×128 , soit 1×2^7 . Je retranche 128 de 186 et j'obtiens 58.

Dans 58, on trouve 0×64 , soit 0×2^6 . Je ne retranche donc rien.

Dans 58, on trouve 1×32 , soit 1×2^5 . Je retranche 32 de 58 et j'obtiens 26.

Dans 26, on trouve 1×16 , soit 1×2^4 . Je retranche 16 de 26 et j'obtiens 10.

Dans 10, on trouve 1×8 , soit 1×2^3 . Je retranche 8 de 10 et j'obtiens 2.

Dans 2, on trouve 0×4 , soit 0×2^2 . Je ne retranche donc rien.

Dans 2, on trouve 1×2 , soit 1×2^1 . Je retranche 2 de 2 et j'obtiens 0.

Dans 0, on trouve 0×1 , soit 0×2^0 . Je ne retranche donc rien.

Il ne me reste plus qu'à reporter ces différents résultats (dans l'ordre !) pour reconstituer l'octet.

J' écris alors qu'en binaire, 186 est représenté par :

1 0 1 1 1 0 1 0

Un algorithme, c'est une suite d'instructions, qui une fois exécutée correctement, conduit à un résultat donné.

Le big data, c'est quoi?

Dans un délai de 30 secondes, on envoi:

Pendant ce petit laps de temps, environ 80 millions d'emails,

6 millions de SMS,

16 millions de messages Whatsup,

140 000 tweets ont été envoyés dans le monde.

Et près de 60 millions de mots traduits sur Google Translate.

Ces chiffres vertigineux sont en croissance quasi exponentielle.

Source: <https://www.linkedin.com/learning/les-fondements-de-l-intelligence-artificielle/definir-l-ia>

Les fondamentaux de tous les algorithmes et de tous les modèles de l'intelligence artificielle sont principalement basés sur les mathématiques au sens large.

En effet, avant l'avènement de l'intelligence artificielle, les scientifiques et les experts se basaient sur les mathématiques pour modéliser et résoudre les problèmes complexes.

Ils utilisaient essentiellement la recherche opérationnelle avec ses deux composantes essentielles, à savoir

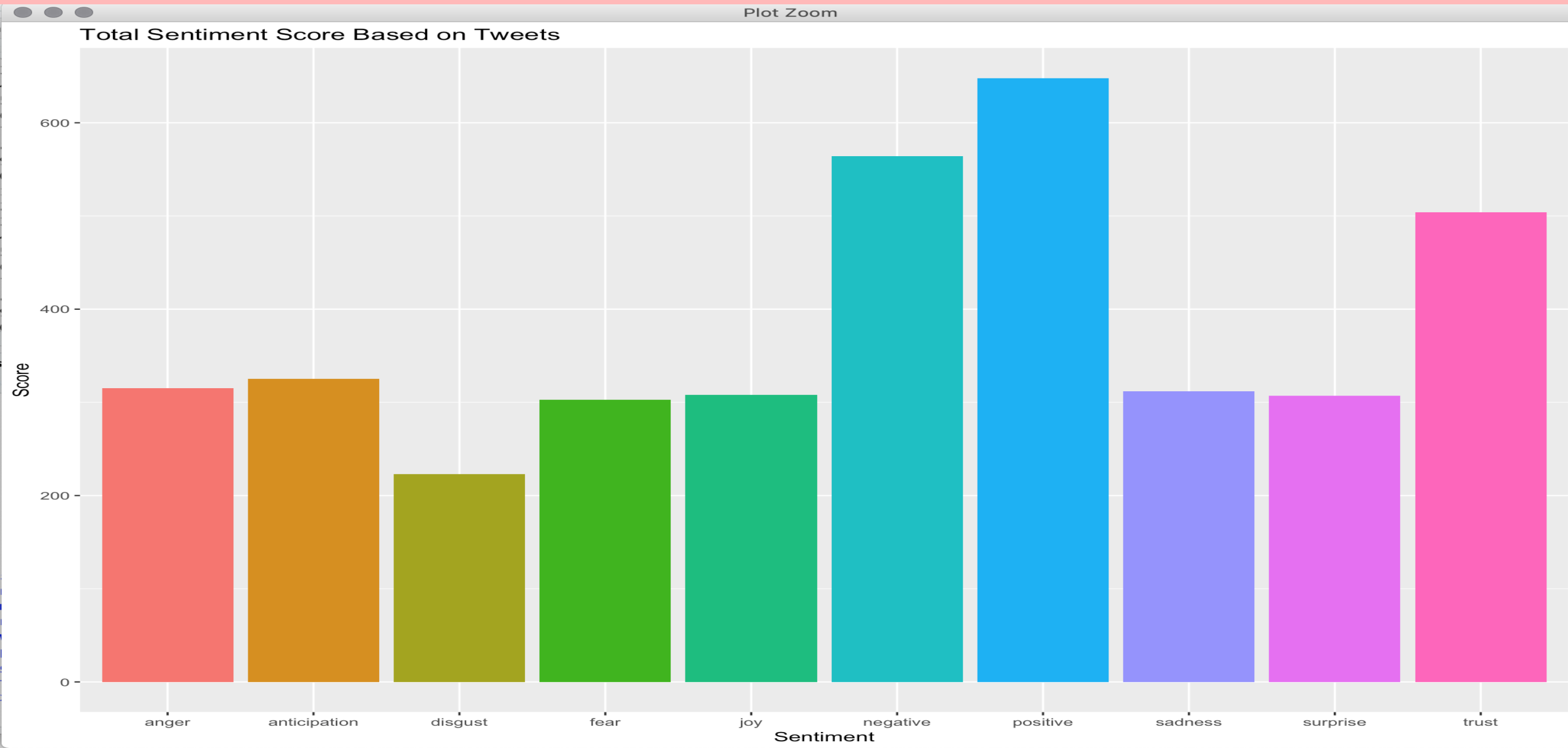
la théorie des graphes et

la programmation linéaire,

l'algèbre et l'analyse mathématique,

les probabilités et statistiques,

Analyse des sentiments



Le marché de l'IA

En 2017, il existait déjà quatre milliards d'objets connectés qui utilisaient des fonctionnalités d'intelligence artificielle.

Les cabinets spécialisés dans l'étude des marchés estiment que en 2020, le nombre d'objets connectés et reliés à l'IA va atteindre les sept milliards et que à la même année, 90% des services à la clientèle seront réalisés avec l'aide de l'IA et 85% des interactions entre une entreprise et ses clients seront effectuées ou du moins initiées sans intervention humaine.

En 2025, le marché de l'intelligence artificielle va représenter 90 milliards de dollars de l'économie mondiale. En 2030, 35% des emplois dans le monde seront entièrement ou partiellement automatisés.

Algorithme de classification.

Naive Bayes

Les domaines d'application du classifieur bayésien naïf sont les suivants :

traitement automatique du langage naturel,
détection de spams dans vos logiciels de courrier électronique,
système de recommandation d'achat,
et également analyse de sentiments dans les enquêtes ou dans les commentaires d'achat d'articles.

Questions?

Merci!

À mardi prochain sur Meetup:

<https://meet.google.com/zoa-xsnw-thr>

Références:

Christophe Darmangeat, ALGORITHMIQUE ET PROGRAMMATION POUR
NON-MATHEUX, Dunod, 2008

Stuart J. Russell and Peter Norvig,
Artificial Intelligence
A Modern Approach,
Third Edition,
PRENTICE HALL ,2009

Références:

<https://www.linkedin.com/learning/les-fondements-de-l-intelligence-artificielle/comprendre-le-marche-de-l-ia>