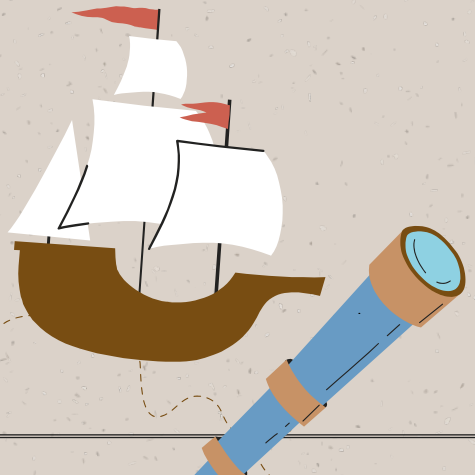
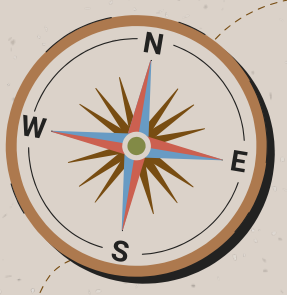
Machine Learning technique

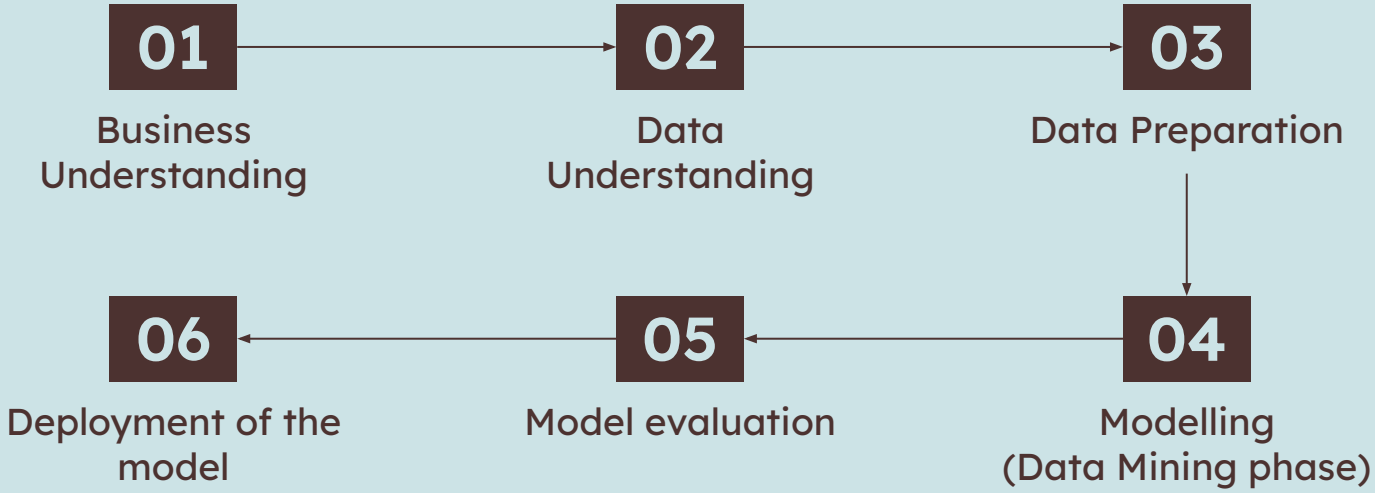
Nathan Hoche



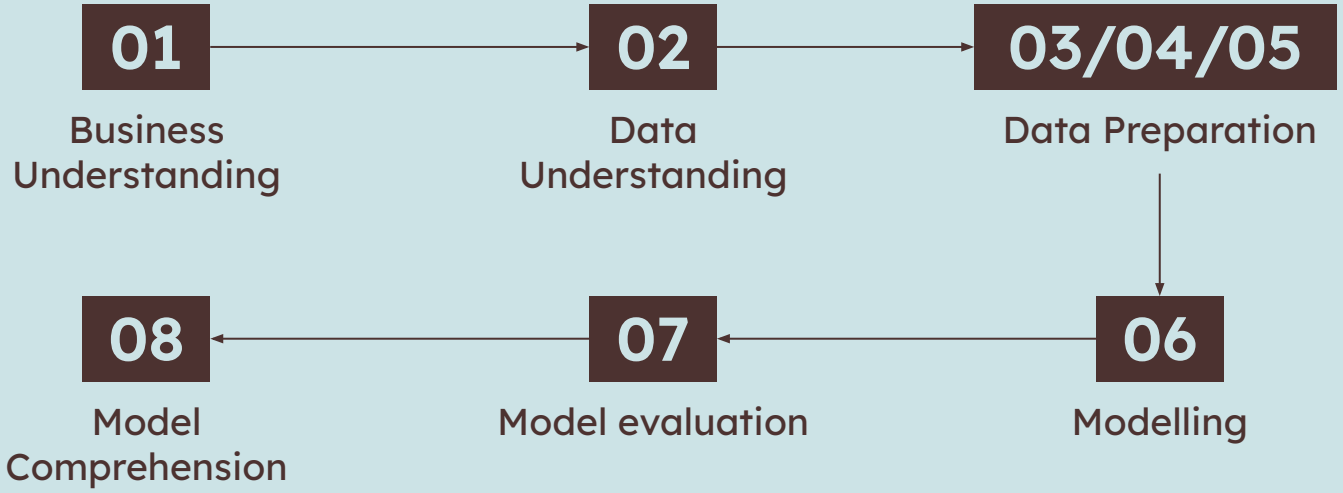


CRISP-DM Process Model

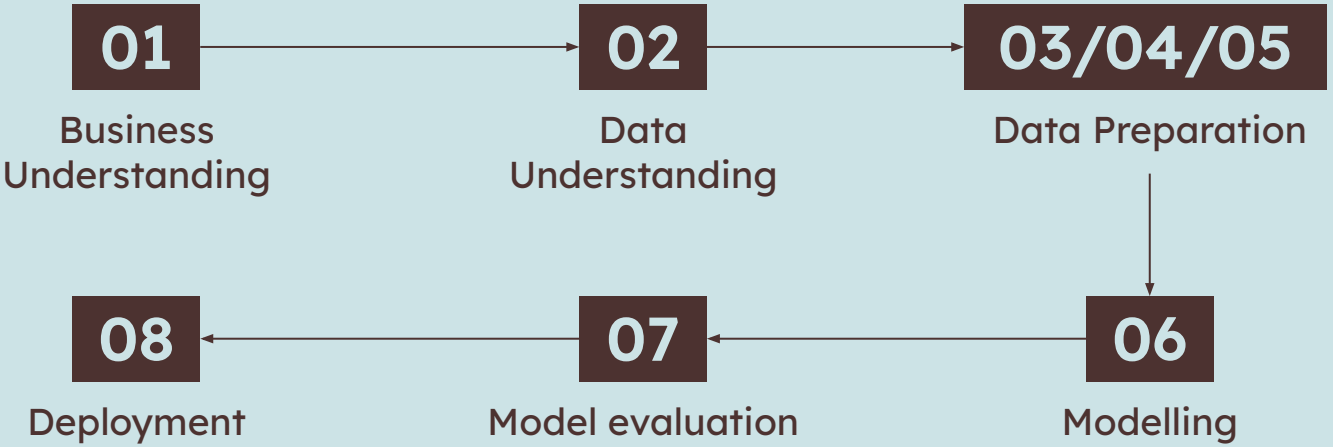
CRoss-Industry Standard Process for Data Mining



Adaptation pour la Data Science



Adaptation pour le Machine Learning



01

Data Understanding

Techniques - Association Rules

ID	Items
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}
...	...

market basket transactions

{Diapers, Beer} Example of a frequent itemset

{Diapers} → {Beer} Example of an association rule

Objectif: Trouver les corrélations entre les éléments

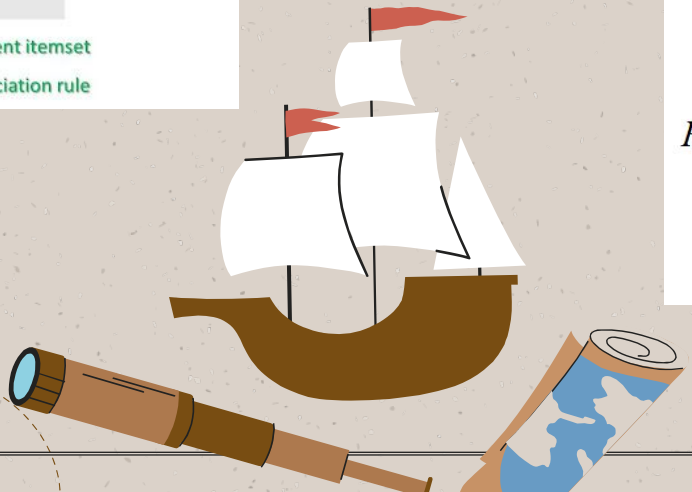
Techniques:

$$Rule: X \Rightarrow Y$$

$$Support = \frac{frq(X,Y)}{N}$$

$$Confidence = \frac{frq(X,Y)}{frq(X)}$$

$$Lift = \frac{Support}{Supp(X) \times Supp(Y)}$$



02

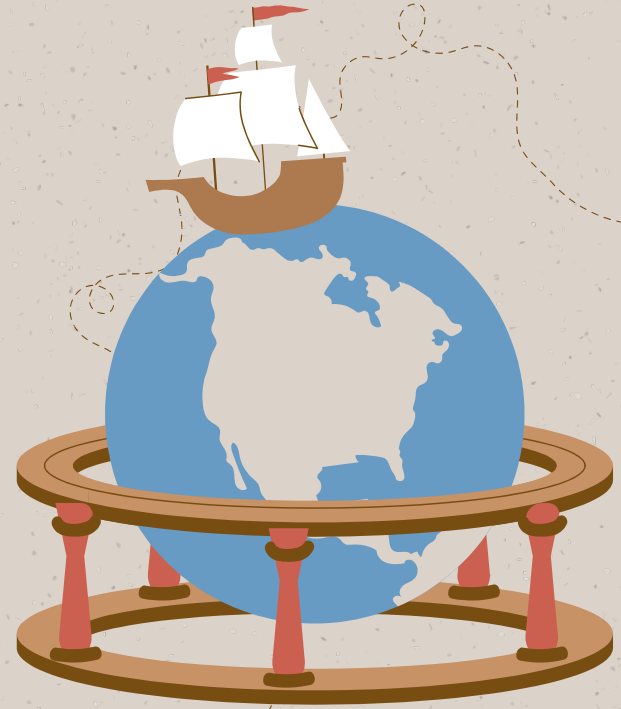
Data Preparation

Problème - Fairness

IMAGE NET (1,2 million d'images), mais :
- 45 % des images proviennent des États-Unis -> 4 % de la population mondiale
- 3% des images proviennent de l'Inde et de la Chine -> 36% de la population mondiale

Pour les algorithmes de vision par ordinateur -> la mariée est habillée en blanc -> ne reconnaît pas la mariée indienne habillée en rouge

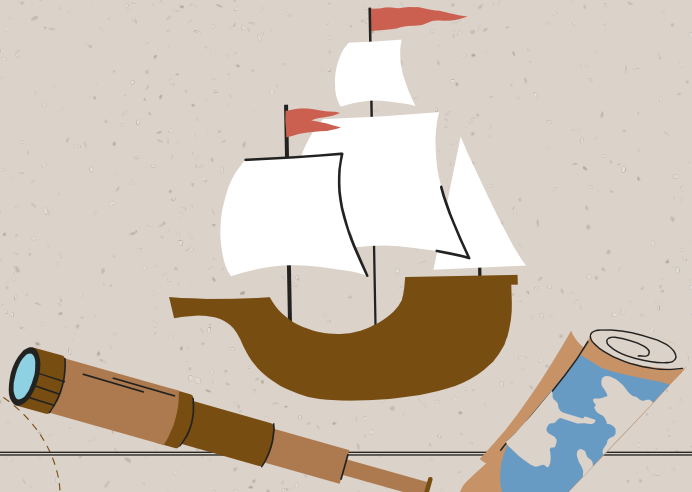
Biobanque britannique (500 000 participants), mais :
- 5% ont une maladie cardiovasculaire -> mais c'est 10% de la population en général



Problème - Cas dupliquer

Méthodes:

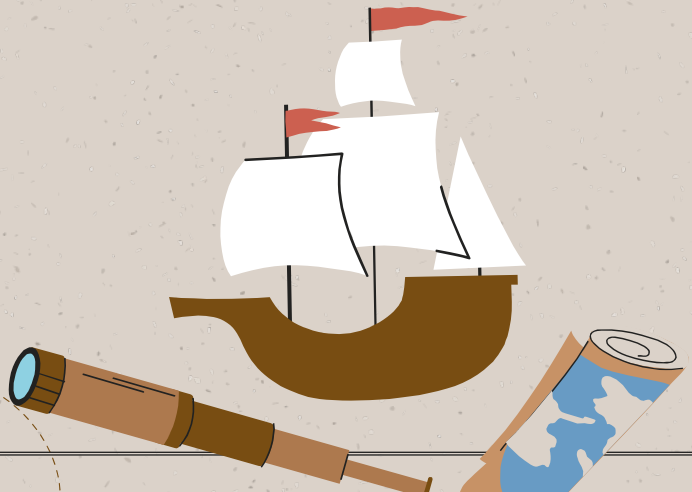
1. Suppression des attributs sensibles (ex: Username, ID, postal Code, ...)
2. Data massaging (Sélection des cas les plus représentatifs de chaque classe)
3. Reweighting (Assignez un poids à chaque cas en fonction de leur proportion dans le dataset)



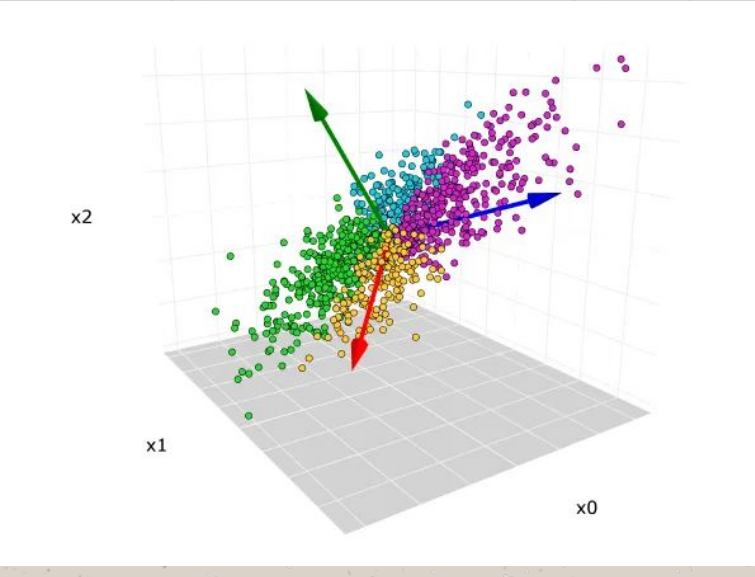
Problème - Classe Déséquilibrée

Méthodes:

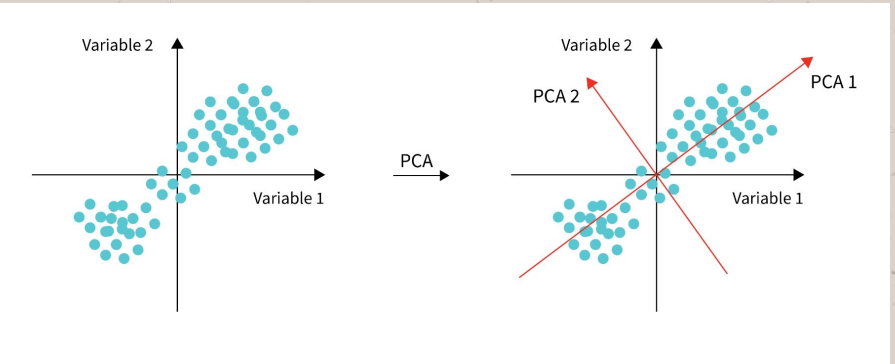
1. Under-Sampling (Suppression aléatoire de cas de la classe majoritaire)
2. Over-Sampling (Dupliquer les cas des classes majoritaire ou en générer des nouveaux)



Problème - Difficulté de classification

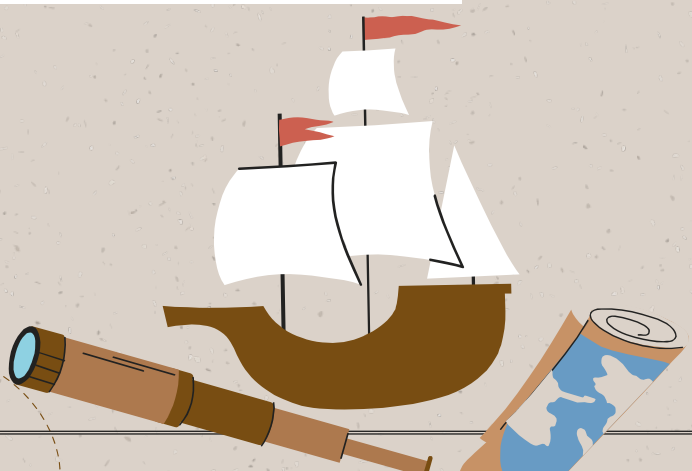


Technique - PCA



Objectif: Recrée des features plus intéressantes pour la classification

Méthode: Déplacement d'axes dans l'objectif de réduire la "variance" entre les attributs

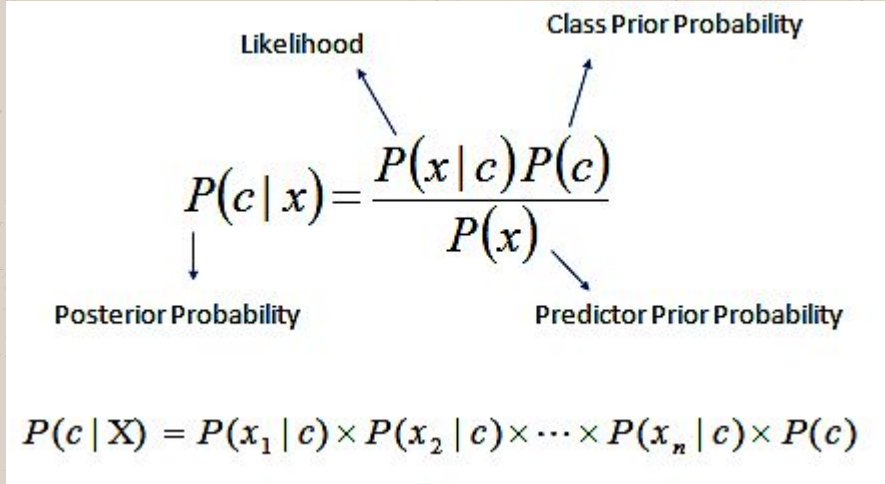


03

Modelling

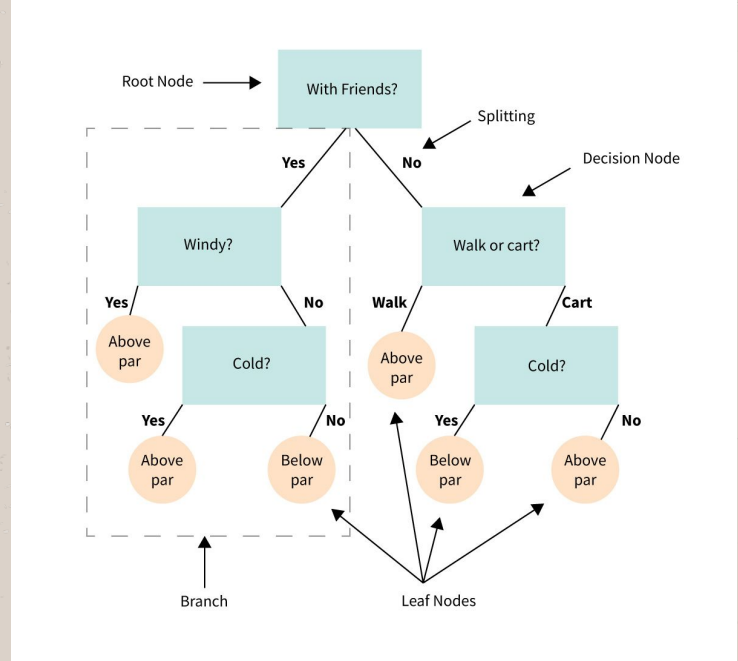
Probabilistic method: Naïve Bayes

Méthode: Calculer la probabilité de chaque classe en fonction de la valeur des attributs connu



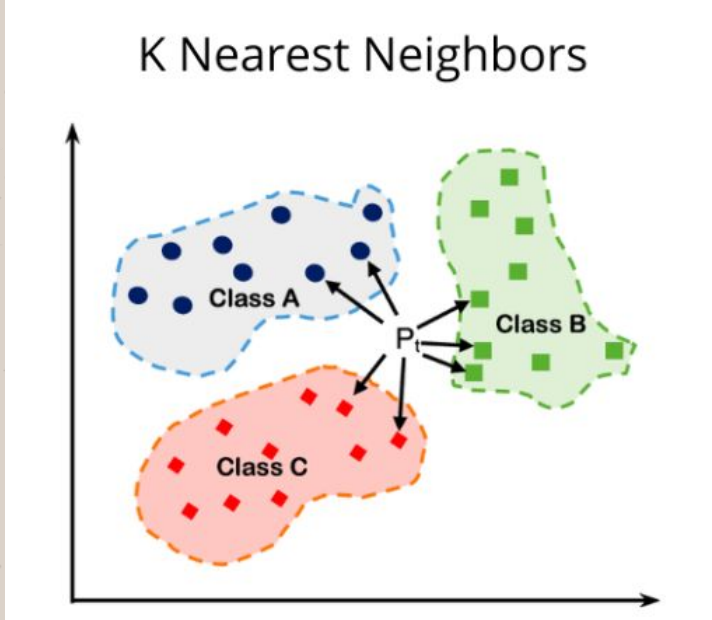
Conditional method: Decision Tree

Méthode: Création d'un arbre décisionnel afin d'identifier la classe



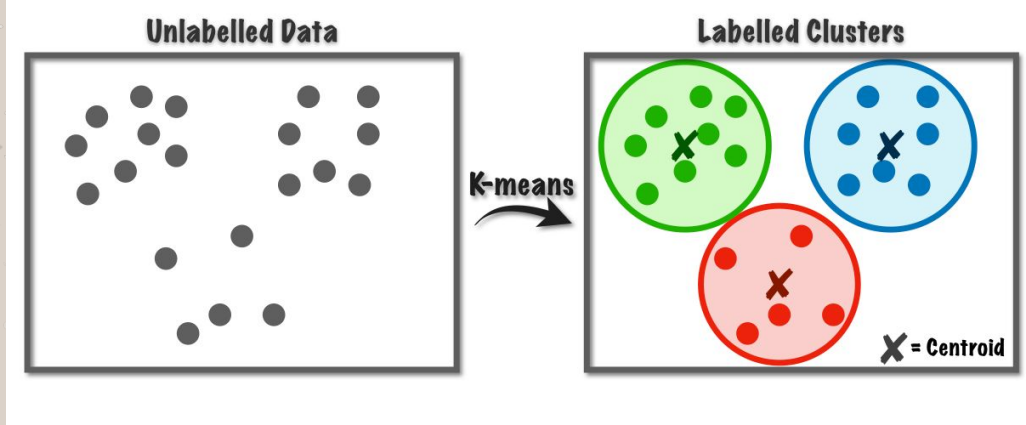
Distance method: KNN

Méthode: Calcul des distances entre chaque élément des cas connus, afin d'identifier la classe la plus proche de X



Clustering method: K-Means

Méthode: Mise en place de cluster (correspondant chacun à une classe). La comparaison entre le point X et le centroid des clusters permet de définir la classe de X.



04

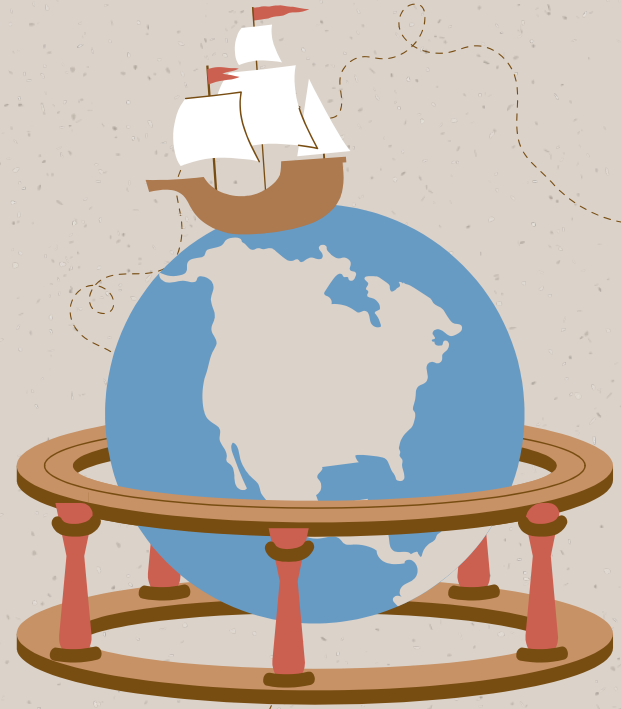
Model Evaluation

Comment savoir si notre modèle est valide?

Problème - Mesurer la performance

Calcul de l'Accuracy (nombre de bonne réponse sur le nombre de réponse).

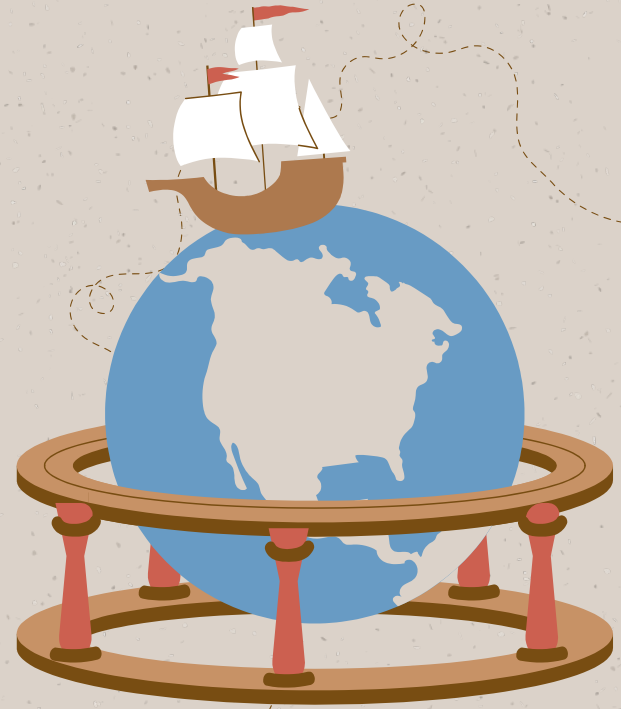
Peut être combiner avec des alternatives: precision, recall, F-measure



Problème - Vérifier la Généralité

Lors de la phase d'apprentissage, il y a un risque que l'agencement des données soient avantageuse. Le modèle n'est pas tester sur les cas compliqué.

Pour l'identifier, il faut utiliser le K-fold cross validation.



05

Pratique

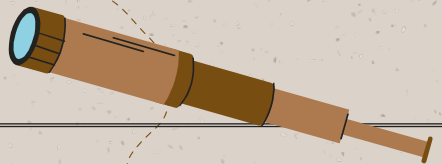
Iris Dataset

Groupe: 3-5 personnes

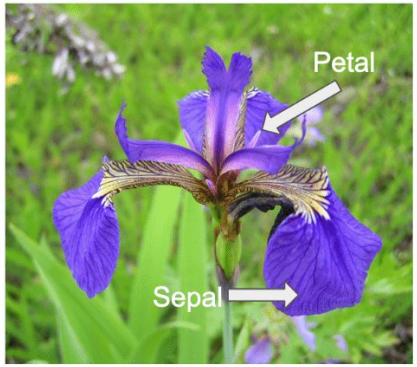
Objectif: Mise en place d'un modèle pour identifier le type de la plante

Workshop:https://github.com/nathan-hoche/Workshop-2024/blob/main/Track-IA/02-Clustering.ipynb

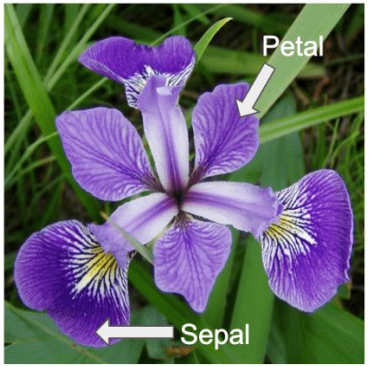
Aide:https://scikit-learn.org/stable/auto_examples/datasets/plot_iris_dataset.html



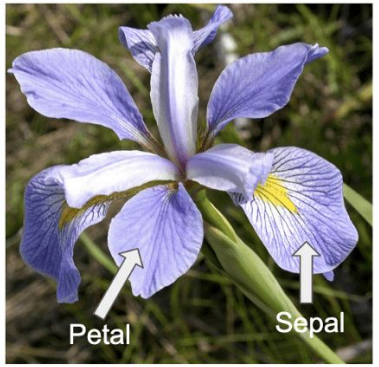
Iris setosa



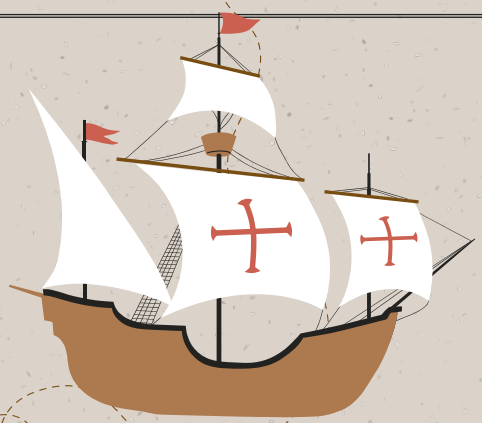
Iris versicolor

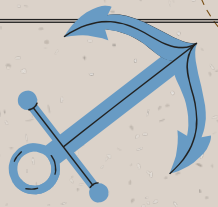


Iris virginica



Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5	3.4	1.5	0.2	Iris-setosa
9	4.4	2.9	1.4	0.2	Iris-setosa
10	4.9	3.1	1.5	0.1	Iris-setosa
11	5.4	3.7	1.5	0.2	Iris-setosa
12	4.8	3.4	1.6	0.2	Iris-setosa
13	4.8	3	1.4	0.1	Iris-setosa
14	4.3	3	1.1	0.1	Iris-setosa
15	5.8	4	1.2	0.2	Iris-setosa
16	5.7	4.4	1.5	0.4	Iris-setosa
17	5.4	3.9	1.3	0.4	Iris-setosa
18	5.1	3.5	1.4	0.3	Iris-setosa
19	5.7	3.8	1.7	0.3	Iris-setosa





Conclusions

scikit-learn

Machine Learning in Python

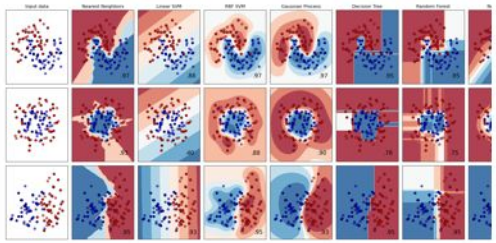
Getting Started | Release Highlights for 1.4 | GitHub

- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.
Algorithms: Gradient boosting, nearest neighbors, random forest, logistic regression, and more...

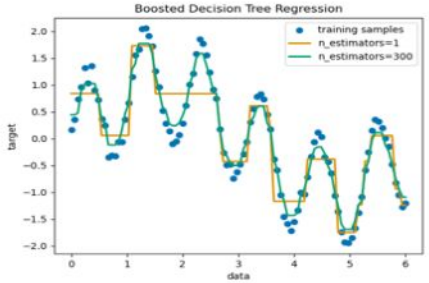


Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: Gradient boosting, nearest neighbors, random forest, ridge, and more...

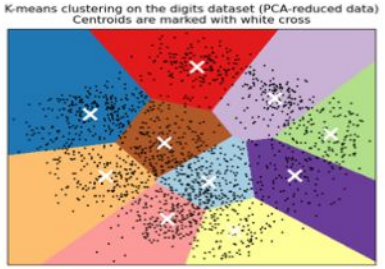


Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, HDBSCAN, hierarchical clustering, and more...



Examples

Dimensionality reduction

Model selection

Preprocessing

Browse State-of-the-Art

12,384 benchmarks 4,674 tasks 117,448 papers with code

Computer Vision

Semantic Segmentation

280 benchmarks
4896 papers with code

Image Classification

455 benchmarks
3591 papers with code

Object Detection

332 benchmarks
3506 papers with code

Contrastive Learning

1 benchmark
1970 papers with code

Image Generation

430 benchmarks
1766 papers with code

See all 1661 tasks

Natural Language Processing

Language Modelling

62 benchmarks
3884 papers with code

Translation

7 benchmarks
3086 papers with code

Question Answering

239 benchmarks
2610 papers with code

Machine Translation

95 benchmarks
2069 papers with code

Text Generation

266 benchmarks
1367 papers with code

Avez-vous des questions?