**Optimisation du stockage d’un cloud avec une IA**

Récupération des données pour le training :

Les données dont on avait parlé sont en **local** ou à partir d’un **cloud**? Il faudra adapter le programme afin d’accéder et parcourir les données afin de récupérer les différents variables qu’on utilise .

Dataset généré a partir d’un programme python :

dataset=["taille","extension",

"jour\_crea","mois\_crea","annee\_crea",

"jour\_modif","mois\_modif","annee\_modif",

"jour\_acces","mois\_acces","annee\_acces",

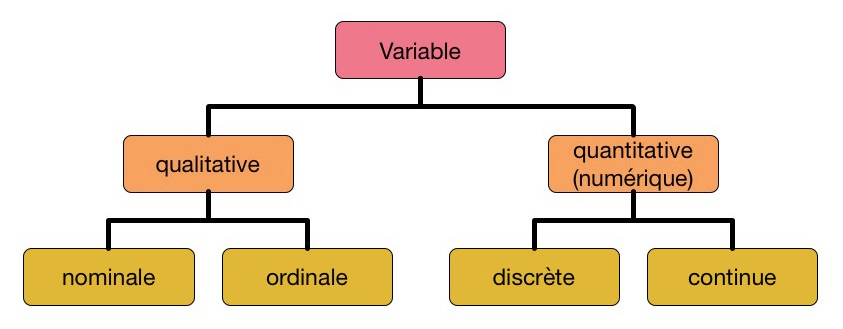
"duree\_modif\_acces","duree\_modif\_crea",

"categ\_taille","categ\_ext",

"categ\_date\_crea","categ\_date\_modif",

"categ\_date\_acces"]

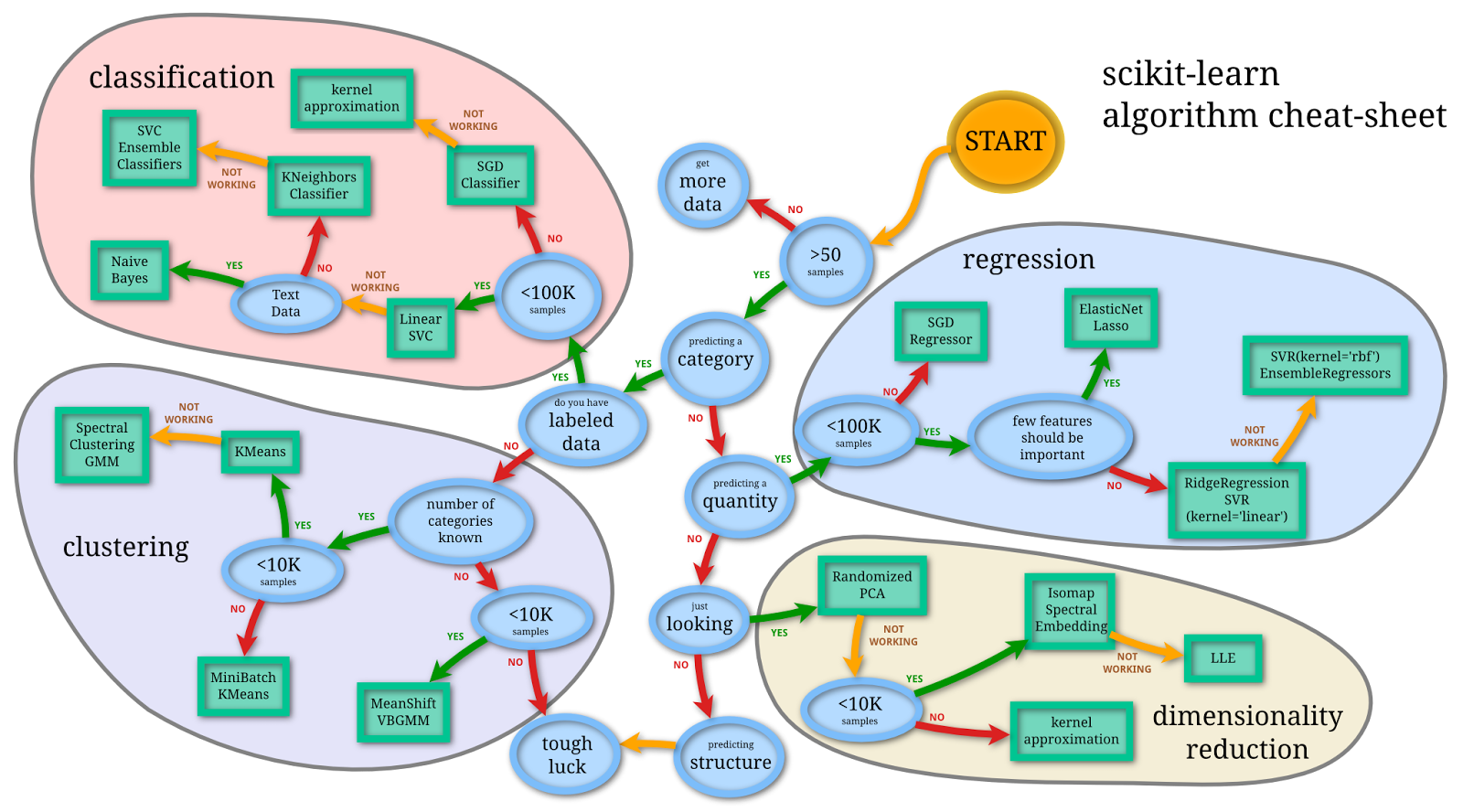
Est-ce judicieux d’utiliser les extensions et si oui comment (**variable** qualitative nominale)?

Figure 1: Type de variables

Division du dataset  et cross validation:

Train\_dataset 80 % et test\_dataset 20%

Choix model de machine learning :

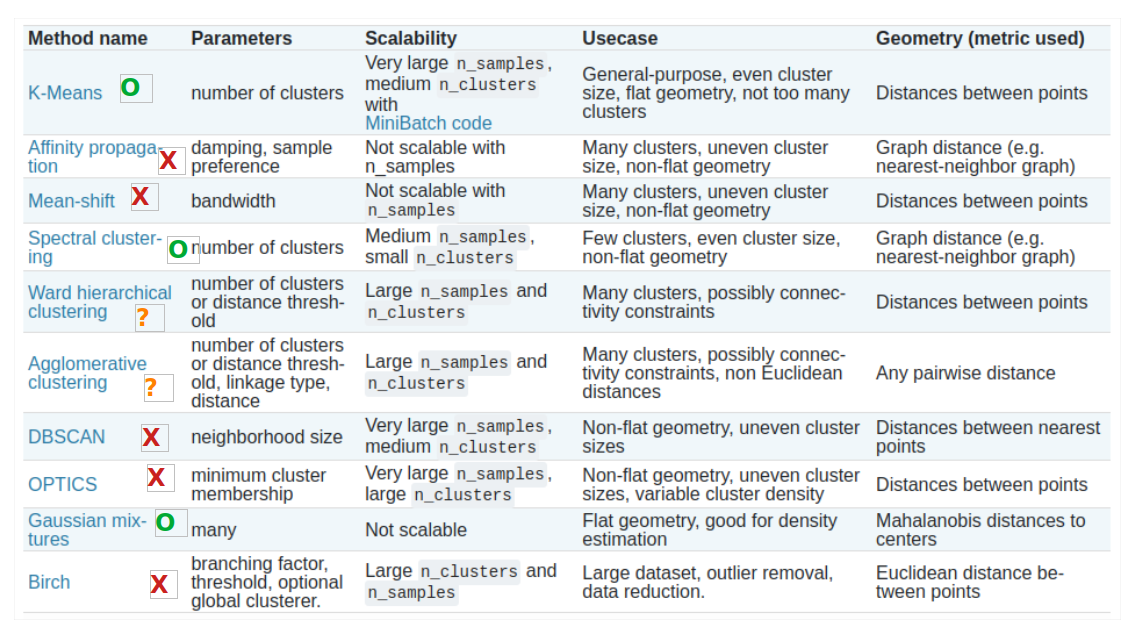
Figure 2: Carte des algorithmes de scikit-learn

On a des données **structurées** : tableau

On a veut prédire si oui ou non le fichier doit être supprimé donc 2 catégories

Nos données n’ont pas de labels donc on est apprentissage **non** **supervisé**

Donc on va s’orienter vers les modèles de **clustering**

Figure 3: Principaux modèles de clustering

Sur la carte des algorithmes de scikit-learn on voit que dans notre cas il est conseillé d’essayer un modèle K-means puis si ce n’est pas satisfaisant utiliser les modèles Spectral clustering ou Gaussian mixtures.

Sites des programme dont on va s’inspirer pour créer les modèles :

K-means :

<https://www.altoros.com/blog/using-k-means-clustering-in-tensorflow/>

K-means, Agglomerative clustering, hierarchical clustering:

[https://github.com/PacktPublishing/Learning-Path-Machine-learning-with-Tensorflow-and-Scikit-Learn/blob/master/Module%2001/PythonMachineLearningSecondEdition\_Code/Chapter11/ch11.py](https://github.com/PacktPublishing/Learning-Path-Machine-learning-with-Tensorflow-and-Scikit-Learn/blob/master/Module 01/PythonMachineLearningSecondEdition_Code/Chapter11/ch11.py)

Spectral clustering :

https://www.kdnuggets.com/2020/05/getting-started-spectral-clustering.html

Gaussian mixtures :

https://www.tensorflow.org/probability/examples/Bayesian\_Gaussian\_Mixture\_Model

**RECHERCHES :**

Installation Tensorflow avec une carte graphique AMD Radeon.

<https://lecrabeinfo.net/installer-ubuntu-20-04-lts-le-guide-complet.html>

<https://rocmdocs.amd.com/en/latest/Installation_Guide/Installation-Guide.html>

<https://github.com/ROCmSoftwarePlatform/tensorflow-upstream/blob/develop-upstream/rocm_docs/tensorflow-install-basic.md>

<https://github.com/ROCmSoftwarePlatform/tensorflow-upstream>

https://www.tensorflow.org/install/pip

Méthodes de path:

<https://docs.python.org/fr/3/library/pathlib.html>

<https://webdevdesigner.com/q/how-to-get-file-creation-modification-date-times-in-python-47587/>

Cours en français sur le machine learning :

<https://www.youtube.com/watch?v=P6kSc3qVph0&list=PLO_fdPEVlfKqMDNmCFzQISI2H_nJcEDJq&index=20>

Cours sur clustering :

<http://www.vincentlemaire-labs.fr/cours/2.2-ApprentissageNonSupervise.pdf>

<https://scikit-learn.org/stable/modules/clustering.html>

<https://openclassrooms.com/fr/courses/4379436-explorez-vos-donnees-avec-des-algorithmes-non-supervises/4379551-decouvrez-l-interet-des-algorithmes-de-clustering>

Vérifier que tensorflow est bien installé :

pip list | grep tensorflow && pip uninstall -y tensorflow

Activer et désactiver un environnement virtuel :

source ./venv/bin/activate # sh, bash, or zsh

deactivate

Si on a assez de temps :

Utiliser les fichiers de la corbeille pour donner des poids aux variables.