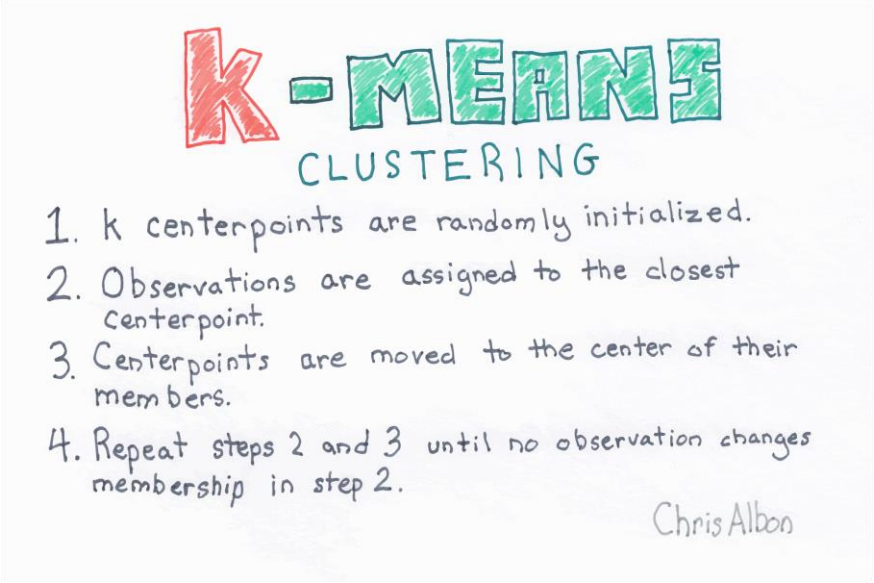
Algorithmes de Machine Learning

Etape 2

Implémentez votre propre algorithme de clustering k-means



There are no funny pictures of clustering
Source : [Chris Albon]

Objectifs de l'activité

- Comprendre la mise en œuvre du clustering k-means
- Pratiquez l'écriture de votre python et algorithme en créant votre propre module k-means!

Compétences

- Mettre en œuvre un algorithme k-means pour un apprentissage non supervisé

Modalité pédagogique

- Lisez le livre *Hands-on Machine Learning*, Chapitre 9, Introduction : p.235 - 248. Jetez un œil aux autres ressources ci-dessous.

- Assurez-vous de bien comprendre les étapes que l'algorithme k-means utilise pour trouver les étiquettes de cluster
- Vous allez créer un module qui utilise k-means pour affecter des étiquettes à un ensemble de données en utilisant uniquement des bases python et NumPy. Les données sont bidimensionnelles et au format d'un tableau NumPy.
    - 🟡 Faites un premier aperçu de votre algorithme (sur papier si nécessaire !). Assurez-vous d'écrire un pseudo-code pour votre algorithme dans un script python - avant de faire du code ! (Ça va vraiment aider !)
    - 🟡 Pensez aux différents éléments de python dont vous aurez besoin et à leur emplacement dans votre pseudo-code (par exemple, quels types de données, boucles, opérations, vous devrez peut-être utiliser). Si vous n'êtes pas sûr de leur fonctionnement, pensez à passer du temps dans les prochains jours à apprendre comment ils fonctionnent lorsque vous devez les utiliser. Cela aidera vos compétences de codage à long terme. Si vous avez besoin d'aide pour des idées, demandez à l'instructeur.
    - 🟡 Assurez-vous de commenter votre code au fur et à mesure, afin que les autres (et vous à l'avenir !) puissent le comprendre. Si votre code final n'est pas bien commenté, vous ne passerez pas la compétence.
    - 🟡 Testez votre algorithme sur les différents datasets. Votre algorithme doit fournir des étiquettes pour chacun des points de données.
    - 🟡 Créez une fonction pour tracer les données fournies et les centroïdes de cluster. Colorez les points de données de chaque cluster avec des couleurs distinctes.
- Utilisez votre module pour trouver le nombre optimal de clusters adapté aux datasets.

Ressources

- Clustering dans le chapitre du livre scikit-learn (voir chapitre 9)
  https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/
- Brève explication des k-means
  https://www.youtube.com/watch?v=_aWzGGNrcic
- K-means interactif
  https://www.naftaliharris.com/blog/visualizing-k-means-clustering/
- Introduction à Spyder IDE
  https://www.youtube.com/watch?v=zYNRgVimU3Q

Livrables

- Script / notebook python qui contient :
    - 🟡 Une implémentation de l’algorithme k-means
    - 🟡 Un tracé du résultat sur les données fournies (décrit ci-dessus)

Pour aller plus loin

- Assurez-vous que votre code est conforme aux directives de bonnes pratiques (comme PEP8)
  https://www.python.org/dev/peps/pep-0008/
  https://www.python.org/dev/peps/pep-0257/
  https://stackoverflow.com/questions/356161/python-coding-standards-best-practices

- Augmentez le nombre de points dans l'ensemble de données. Reportez le temps nécessaire à l'exécution de votre module. Comparez-le à l'implémentation de k-means de scikit-learn.
- Pensez à refactoriser votre code (réécriture) afin qu'il soit plus simple à lire ou qu'il s'exécute plus rapidement (les deux peuvent être importants !).
- Pensez à améliorer la méthode d'initialisation aléatoire de votre algorithme (voir page 243 du *Hands-on Machine Learning*) pour éviter des résultats sous-optimaux.
- Faites de votre module une classe qui peut être importée et implémentée de la même manière que scikit-learn.
  https://www.youtube.com/watch?v=ZDa-Z5JzLYM