Les partitions

On a vu dans l'exercice précédent que si on connecte deux consumer au même topic, ils recevaient tous les deux tous les messages.

Or si on veut faire du Machine learning en production, on ne veut pas que deux consumer fasse les prédictions sur deux messages identiques.

On veut qu'un message ne soit traité qu'une fois.

Pour cela on peut utiliser la notion de group et de partition.

Exercice 1 - Partitions

Avant de pouvoir d'utiliser les partitions, il faut les créer. Une partition se crée pour un topic donné.

Pour cela on va utiliser la classe KafkaAdminClient

1. Importer la classe KafkaAdmin client (lien doc) et la classe NewPartitions
2. créer une instance de KafkaAdminClient et stocker dans la variable admin_client
3. Créer un dictionnaire nommé topic_partitions
4. Créer une variable topic avec votre nom de famille ou son abbréviation
5. Créer une instance de la classe NewPartitions (from kafka.admin imoprt NewPartitions) en spécifiant l'argument total_count=2
6. Ajouter au dictionnaire une clé avec le nom du topic qui contient l'instance de la classe NewPartitions
7. Avec la méthode admin_client.create_partitions créer les partitions pour votre topic

Exercice 2 - Utilisation des partitions

Maintenant qu'on a créé plusieurs partitions pour notre topic on va pouvoir dire à chaque consumer de se connecter à une partition différente.

1. Enlever l'argument dans le constructeur qui permet de spécifier le nom du topic. On va le faire d'une autre manière
2. Reprendre votre code qui permet de créer un consumer et qui affiche un message et spécifier l'argument group_id="grp1" à la création d'un consumer
3. Importer la classe TopicPartition instancier la classe avec le nom du topic et la partition valant 0
4. Avec la méthode assign du consumer faire en sorte d'assign l'instance TopicPartition au consumer
5. Lancer le consumer

6. Lancer un second consumer afin qu'il soit sur le même group-id et le même topic mais avec une partition différente.
7. Dans le fichier Producer.py faire en sorte de spécifier l'argument partition avec un nombre aléatoire entre 0 et 1.
8. Envoyer plusieurs messages et vérifier que seulement un des deux consumer reçoit les messages selon où le producer envoie son message