

Fichier Python consumer pour détection d'erreur

```
## Bibliotheque
from kafka import KafkaConsumer
from kafka.structs import TopicPartition

from sklearn.ensemble import IsolationForest
import pandas as pd

from sklearn.cluster import DBSCAN
import time

import pickle
listeDonnee = []
pickle.dump( listeDonnee, open( "saveLISTEDONNEE_DBSCAN.txt", "wb" ) )

## Fonction pour convertir une liste en DataFrame
def convert_list_database(liste):
    tableau = []
    for i in range(len(liste)-1):
        tableau.append(liste[i])
    # return tableau
    return pd.DataFrame(tableau, columns=['c1'])

## Variable
nomTopic = 'humidite'
bootstrap_servers = 'localhost:9092'
topic = TopicPartition(nomTopic, 0)

## Programme
## Création d'un consumer
consumer = KafkaConsumer(bootstrap_servers = bootstrap_servers)

## Assignation au topic que l'on va lire
consumer.assign([topic])

## Permet d'aller à la fin de la partition
consumer.poll()
consumer.seek_to_end(topic)

## Création de la FILE
taille_File = 20 +1
file = []

## Boucle des messages
for msg in consumer:
    ## Décodage du message, ici en UTF-8
    msg = msg.value.decode("utf-8")

    ##Partie detection

    ## Partie de détection d'erreur
    if len(file) > taille_File:
        file.append(msg)
        file.pop(0)
```

```
print("Erreur ?")
#etat allant de -1 à 1

## on met ci-dessous la fonction de détection d'anomalies
etat = detectionAnomaly(file, taille_File)

## Partie d'initialisation
else:
    print("Initialisation", round(len(file)/taille_File,2)) # Pourcentage ?
    file.append(msg)
    # etat 2 correspond à l'initialisation
    etat = -2

## Partie pour sauvegarder les erreurs dans un fichiers externes (optionnel)
temps = time.time()
valeur = msg

donnee = (valeur,temps,etat)
# sauvegarde des données dans un fichier de sauvegarde !consomme du temps si le fichier est trop gr
and
listeDonnee = pickle.load( open( "saveLISTEDONNEE_DBSCAN.txt", "rb" ) )
listeDonnee.append(donnee)
pickle.dump( listeDonnee, open( "saveLISTEDONNEE_DBSCAN.txt", "wb" ) )

## Aller à la fin de la partition
consumer.poll()
consumer.seek_to_end(topic)
```