Spécifications techniques de « Data_base.py »

Introduction

Ce programme permet de générer une base de données afin d'entrainer la random forest pour améliorer le modèle de prédiction du processus de Hawkes.

Architecture technique du script « data_base.py »

On va lire les messages du topic « cascade_properties », contenant les prédictions d'une cascade partielles (type : « parameters ») et la longueur réelle de la cascade (type : « size ») et les stockés dans un data frame. Quand la longueur du data frame atteint une certaine valeur, on sauvegarde cette data frame sous la forme d'un fichier .csv pour être utilisé par « random_forest ».

Les attributs sauvegardés dans le data frame sont : beta, G1, n* et n_par.

Un fichier de test : « test_data_base.py » existe pour vérifier en cas de modification de ce script que celui-ci est toujours opérationnel.

Architecture fonctionnelle du script « data_base.py »

Le script est composé de 10 fonctions python :

1. type_message : permettant de récupérer le type d'un message Kafka
2. id_message : permettant de récupérer l'identifiant d'un message Kafka
3. p_message : permettant de récupérer le paramètre p d'un message Kafka
4. beta_message : permettant de récupérer le paramètre beta d'un message Kafka
5. G1_message : permettant de récupérer le paramètre G1 d'un message Kafka
6. calcul_n_star : permettant de calculer n*
7. n_tot_message : permettant de récupérer la longueur totale d'une cascade d'un message Kafka
8. n_partiel : permettant de récupérer la longueur d'une cascade au bout de 600s d'une cascade Kafka
9. coeff : permettant de calculer la valeur de W
10. creation_db : pour lancer la création d'une database

La création d'une data frame se lance avec la commande suivante : python3 Data_Base.py creation_db chemin, taille_df

type_message

Variable d'entrée : • liste : un message Kafka transformer en liste
Variable de sortie : • type_m : l'attribut « type » d'un message Kafka
Algorithme : 1. On récupère l'attribut « type » du message Kafka

Id_message

Variable d'entrée : • liste : un message Kafka transformer en liste
Variable de sortie : • id_m : l'attribut « type » d'un message Kafka
Algorithme : 1. On regarde si le type du message est « parameters »

2. On récupère l'attribut « identifiant » du message Kafka

P_message

Variable d'entrée :
- liste : un message Kafka transformer en liste

Variable de sortie :
- p_m : l'attribut « p » d'un message Kafka

Algorithme :
1. On regarde si le type du message est « parameters »
2. On récupère l'attribut « p » du message Kafka

Beta_message

Variable d'entrée :
- liste : un message Kafka transformer en liste

Variable de sortie :
- beta_m : l'attribut « beta » d'un message Kafka

Algorithme :
1. On regarde si le type du message est « parameters »
2. On récupère l'attribut « beta » du message Kafka

G1_message

Variable d'entrée :
- liste : un message Kafka transformer en liste

Variable de sortie :
- id_m : l'attribut « G1 » d'un message Kafka

Algorithme :
1. On regarde si le type du message est « parameters »
2. On récupère l'attribut « G1 » du message Kafka

Calcul_n_star

Variables d'entrées :
- p
- alpha
- mu

Variable de sortie :
- n_star : une variable explicative du modèle de la random forest, $n^* = p * \mu * \frac{\alpha-1}{\alpha-2}$

Algorithme :

N_tot_message

Variable d'entrée :
- liste : un message Kafka transformer en liste

Variable de sortie :
- ntot : l'attribut « n_tot » d'un message Kafka

Algorithme :
1. On regarde si le type du message est « size »
2. On récupère l'attribut « n_tot » du message Kafka

N_partiel

Variable d'entrée :
- liste : un message Kafka transformer en liste

Variable de sortie :
- n_par : l'attribut « n_par » d'un message Kafka

Algorithme :

1. On regarde si le type du message est « parameters »
2. On récupère l'attribut « n_par » du message Kafka

Coeff

Variables d'entrées :

- n_tot
- n_partiel
- G1
- n_star

Variable de sortie :

- W : la variable à apprendre, $W = (n_{tot} - n_{partiel}) * \frac{1-n^*}{G_1}$

Algorithme :

Creation_db

Variable d'entrée :

- chemin : le chemin pour écrire la database
- taille_df : la taille du dataframe

Variable de sortie :

Algorithme :

1. On transforme le message Kafka en une liste
2. Si le message a pour type « parameters »
    a. On ajoute dans un dataframe les variables « beta », « n* », « G1 » et « nobs »
    b. On choit comme index, l'identifiant du message Kafka
3. Si le message a pour type « size »
    a. On calcul W
    b. On l'ajoute dans le dataframe a la ligne correspondante
4. Quand la taille du dataframe atteint celle souhaité, on écrit un fichier .csv dans le chemin d'entrée