

## Spécifications techniques de « Data\_base.py »

### Introduction

Ce programme permet de générer une base de données afin d'entraîner 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

### type\_message

|   |
|---|
| Variable d'entrée :   |
| <ul style="list-style-type: none"><li>• liste : un message Kafka transformer en liste</li></ul>     |
| Variable de sortie :  |
| <ul style="list-style-type: none"><li>• type_m : l'attribut « type » d'un message Kafka</li></ul>   |
| Algorithme :  |
| <ol style="list-style-type: none"><li>1. On récupère l'attribut « type » du message Kafka</li></ol> |

### Id\_message

|   |
|---|
| Variable d'entrée :   |
| <ul style="list-style-type: none"><li>• liste : un message Kafka transformer en liste</li></ul>   |
| Variable de sortie :  |
| <ul style="list-style-type: none"><li>• id_m : l'attribut « type » d'un message Kafka</li></ul>   |
| Algorithme :  |
| <ol style="list-style-type: none"><li>1. On regarde si le type du message est « parameters »</li><li>2. On récupère l'attribut « identifiant » du message Kafka</li></ol> |

### P\_message

|   |
|---|
| Variable d'entrée :   |
| <ul style="list-style-type: none"><li>• liste : un message Kafka transformer en liste</li></ul>   |
| Variable de sortie :  |
| <ul style="list-style-type: none"><li>• p_m : l'attribut « p » d'un message Kafka</li></ul>   |
| Algorithme :  |
| <ol style="list-style-type: none"><li>1. On regarde si le type du message est « parameters »</li><li>2. On récupère l'attribut « p » du message Kafka</li></ol> |

### Beta\_message

|  |
|--|
| Variable d'entrée :  |
| <ul style="list-style-type: none"><li>• liste : un message Kafka transformer en liste</li></ul>  |
| Variable de sortie :   |
| <ul style="list-style-type: none"><li>• beta_m : l'attribut « beta » d'un message Kafka</li></ul>  |
| Algorithme :   |
| <ol style="list-style-type: none"><li>1. On regarde si le type du message est « parameters »</li><li>2. On récupère l'attribut « beta » du message Kafka</li></ol> |

### G1\_message

|  |
|--|
| Variable d'entrée :  |
| <ul style="list-style-type: none"><li>• liste : un message Kafka transformer en liste</li></ul>  |
| Variable de sortie :   |
| <ul style="list-style-type: none"><li>• id_m : l'attribut « G1 » d'un message Kafka</li></ul>  |
| Algorithme :   |
| <ol style="list-style-type: none"><li>1. On regarde si le type du message est « parameters »</li><li>2. On récupère l'attribut « G1 » du message Kafka</li></ol> |

### Calcul\_n\_star

|   |
|---|
| Variables d'entrées :   |
| <ul style="list-style-type: none"><li>• p</li><li>• alpha</li><li>• mu</li></ul>  |
| Variable de sortie :  |
| <ul style="list-style-type: none"><li>• n_star : une variable explicative du modèle de la random forest, <math>n^* = p * \mu * \frac{\alpha-1}{\alpha-2}</math></li></ul> |
| Algorithme :  |

### N\_tot\_message

|   |
|---|
| Variable d'entrée :   |
| <ul style="list-style-type: none"><li>• liste : un message Kafka transformer en liste</li></ul>   |
| Variable de sortie :  |
| <ul style="list-style-type: none"><li>• ntot : l'attribut « n_tot » d'un message Kafka</li></ul>  |
| Algorithme :  |
| <ol style="list-style-type: none"><li>1. On regarde si le type du message est « size »</li><li>2. On récupère l'attribut « n_tot » du message Kafka</li></ol> |

### N\_partiel

|  |
|--|
| Variable d'entrée :  |
| <ul style="list-style-type: none"><li>• liste : un message Kafka transformer en liste</li></ul>        |
| Variable de sortie :   |
| <ul style="list-style-type: none"><li>• n_par : l'attribut « n_par » d'un message Kafka</li></ul>      |
| Algorithme :   |
| <ol style="list-style-type: none"><li>1. On regarde si le type du message est « parameters »</li></ol> |

|  |
|--|
| 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

|                      |
|----------------------|
| 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 choisit comme index, l'identifiant du message Kafka
3. Si le message a pour type « size »
  - a. On calcule W
  - b. On l'ajoute dans le dataframe à la ligne correspondante
4. Quand la taille du dataframe atteint celle souhaitée, on écrit un fichier .csv dans le chemin d'entrée