**Erreur 1 :**

ModuleNotFoundError: No module named 'drain3'

**Solution 1 :**

$ sudo apt install virtualenv python3-virtualenv -y

$ virtualenv -p /usr/bin/python3 testpip

$ source testpip/bin/activate

$ pip3 install drain3

**Erreur 2 :**

config file not found: drain3.ini

**Solution 2 :**

Ajouter dans le script python le code suivant :

*# Configuration de Drain3*

*config = TemplateMinerConfig()*

*# Définir les paramètres directement*

*config.snapshot\_interval\_minutes = 1*

*config.persist\_snapshot = True*

*config.persist\_state = True*

*config.log\_level = logging.INFO*

*config.tree\_max\_depth = 4*

*config.min\_similarity\_threshold = 0.4*

*config.max\_clusters = 10000*

**Erreur 3 :**

KeyError: 'template'

**Solution 3:**

Vérification de la clé 'template' :

* Avant d'essayer d'accéder à result['template'], une vérification est effectuée pour voir si la clé 'template' existe dans le résultat.
* Si la clé n'existe pas, un avertissement est enregistré.

Vérification de l'attribut 'clusters' :

* Une vérification similaire est faite pour l'attribut 'clusters' dans template\_miner.drain.
* Si l'attribut n'existe pas, un avertissement est enregistré.

**Erreur 4 :**

AttributeError: 'TemplateMiner' object has no attribute 'persistence'

**Solution 3:**

Suppression de l'appel à template\_miner.persistence.save\_state() :

L'appel à save\_state est maintenant fait directement sur l'objet drain de template\_miner car c'est là que l'état des clusters est maintenu.

Sauvegarde de l'état des clusters :

L'appel à save\_state sur template\_miner.drain assure que l'état actuel des clusters est sauvegardé correctement

**Erreur 5 :**

TemplateMiner.save\_state() missing 1 required positional argument: 'snapshot\_reason'

**Solution 5 :**

Ajout de l'argument snapshot\_reason :

L'appel à template\_miner.save\_state inclut maintenant l'argument "Periodic save" pour spécifier la raison de l'enregistrement de l'état.

**Erreur 6 :**

TypeError: 'LogCluster' object is not subscriptable

**Solution 6**

Accès correct aux attributs du résultat :

Vérifiez l'accès aux attributs du résultat en utilisant result['template'] uniquement lorsque cela est nécessaire.

Affichage du nombre de clusters :

Utilisez logger.info(f"Clusters: {len(template\_miner.drain.clusters)}") pour afficher le nombre de clusters sans essayer de souscrire l'objet LogCluster.

Exécution du Script

**Erreur 7 :**

cluster\_id = result["cluster\_id"] ~~~~~~^^^^^^^^^^^^^^ TypeError: 'LogCluster' object is not subscriptable

**Solution 7 :**

Accéder à l'attribut cluster\_id : Si result est une instance de la classe LogCluster, il est probable que cluster\_id soit un attribut de cette classe. Vous devrez donc probablement accéder à cet attribut comme suit :

*cluster\_id = result.cluster\_id*

**Erreur 8 :**

ModuleNotFoundError: No module named 'matplotlib'

**Solution 8 :**

Pip3 install matplotlib

**Erreur 9 :**

Failed to extract font properties from /usr/share/fonts/truetype/noto/NotoColorEmoji.ttf: In FT2Font: Can not load face (unknown file format; error code 0x2)

**Solution 9**

sudo apt-get install fonts-dejavu fonts-dejavu-core fonts-dejavu-mono

sudo apt-get install ttf-mscorefonts-installer

sudo apt-get install fontconfig

**Erreur 10**

IndentationError: unexpected indent

**Solution 10**

Vérifier indentation de code

**Erreur 11**

TypeError: not all arguments converted during string formatting !

**Solution 11**

Vérification de la manipulation des variables et leurs conversions en string

**Erreur 12**

AttributeError: 'TemplateMiner' object has no attribute 'events'

**Solution 12**

Inspecter les attributs et méthodes disponibles dans template\_miner

**Erreur 13**

TypeError: unsupported operand type(s) for +: 'LogCluster' and 'LogCluster'

**Solution 13**

Vérifier les opérations d’additions entre deux objets Clusters

**Erreur 14**

ValueError: Only one class present in y\_true. ROC AUC score is not defined in that case.

**Solution 14**

**Erreur 15**

ValueError: The least populated class in y has only 1 member, which is too few. The minimum number of groups for any class cannot be less than 2.

**Solution 15**

**Erreur 15**

J'ai eu cette erreur :

TypeError: '>' not supported between instances of 'str' and 'int'

dans cette ligne de code :

failure\_events = event\_matrix\_df.columns[event\_counts > 0]

Stp. Corriges. Ce code

**Erreur 16**

J'ai eu cette erreur :

IndexError: list index out of range

dans cette ligne de code :

event\_numbers = [int(event.split('\_')[1]) for event in failure\_events]

~~~~~~~~~~~~~~~~^^^

Stp. Corriges. Ce code :

# Afficher un diagramme à barres des événements de panne

def plot\_failure\_distribution(event\_counts):

failure\_events = event\_counts.index

counts = event\_counts.values

plt.figure(figsize=(10, 6))

plt.bar(failure\_events, counts)

plt.xlabel('Événements')

plt.ylabel('Nombre d\'occurrences')

plt.title('Distribution des pannes')

# Extraire les numéros d'événements à partir des chaînes (par exemple, 'Event\_1' -> 1)

event\_numbers = [int(event.split('\_')[1]) for event in failure\_events]

# Sélectionner les événements multiples de 5 (au lieu de 10)

event\_labels = [f'Event\_{number}' for number in event\_numbers if number % 5 == 0]

event\_positions = [failure\_events[i] for i, number in enumerate(event\_numbers) if number % 5 == 0]

# Définir les étiquettes de l'axe x à des multiples de 5

plt.xticks(event\_positions, event\_labels, rotation=45)

plt.show()

# Afficher les statistiques des pannes

plot\_failure\_distribution(event\_counts)