

Automatisation d'un Pipeline ETL

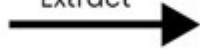
Pipeline ETL

ETL Process

Sourced Systems

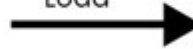


Extract



Transform

Load



Destination



Web scraping

- Collecte automatique de données
- Suivi de l'actualité
- Automatiser la navigation web

Critères de comparaison entre les bibliothèques

Simplicité du code

Performance : capacité à traiter des gros volumes de données

Support Javascript

Gestion des erreurs

Processus de web scraping

- 1) Identifier l'emplacement exacte de la source de données
- 2) Envoyer une requete pour collecter le contenu brut
- 3) Stocker les données extraites

BeautifulSoup

Extraction des données statiques HTML et XML
Nettoyage de contenu HTML : conservation du
texte propre pour la préparation des données à
l'analyse du texte

Scrapy

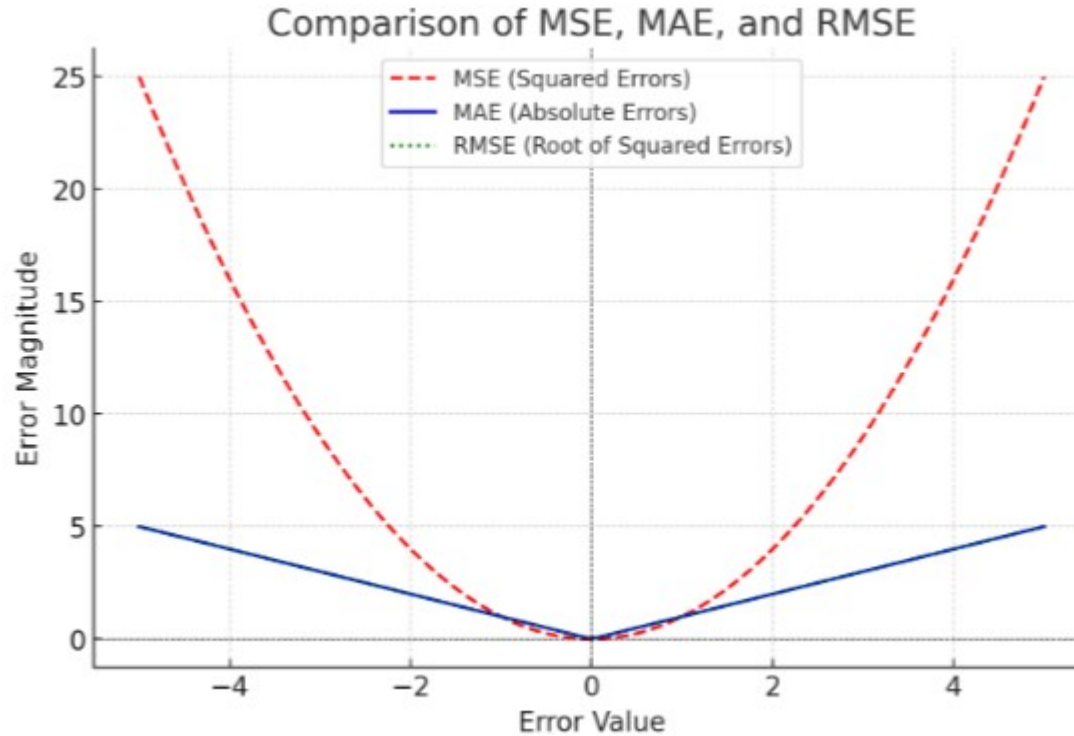
Framework Python open source qui permet :

- collecte de bases de données depuis des sites web
- automatisation de la navigation
- gestion de plusieurs pages

Selenium

- Interaction avec des pages dynamiques
automatisation complète
- Remplissage et soumission de formulaires
- Téléchargement de fichiers

Métriques mae / mse



MAE : Mean Absolute Error

La moyenne des distances absolues (distance de Manhattan) entre les valeurs réelles y_i et les valeurs prédites \hat{y}_i

The diagram illustrates the Mean Absolute Error (MAE) formula with the following components and annotations:

- Formula:** $MAE = \frac{1}{n} \sum |y - \hat{y}|$
- Annotations:**
 - A blue box around $\frac{1}{n}$ is labeled "Divide by the total number of data points".
 - A green box around y is labeled "Actual output value".
 - An orange box around \hat{y} is labeled "Predicted output value".
 - A bracket under the absolute value term $|y - \hat{y}|$ is labeled "The absolute value of the residual".
 - The word "Sum of" is written below the summation symbol \sum .

MSE : Mean squared error

La différence moyenne au carré entre les valeurs estimée \hat{y}_i et les valeur réelles y_i
chaque erreur est élevée au carré

$$MSE = \frac{1}{n} \sum \underbrace{\left(y - \hat{y} \right)}_{\substack{\text{The square of the difference} \\ \text{between actual and} \\ \text{predicted}}}^2$$

cron

- automatiser l'exécution régulière du pipeline ETL à une fréquence définie.

```
GNU nano 6.2 /tmp/crontab.iaGwQg/crontab
## Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
00 08 * * * /home/sara/miniconda3/bin/python3 /home/sara/Documents/GitHub/Pipeline/main.py >> /home/sara/Documents/GitHub/Pipeline/main.log 2>&1
```

