Mode d'emploi du projet

Rania Bentabe

8 juin 2025

1. Prérequis logiciels

- Python 3.7+
- `cx_Oracle` (pour se connecter à Oracle)
- `pandas` et `numpy` (pour le traitement CSV et la génération de données)
- JupyterLab (pour exécuter les notebooks)
- SQLLoader (fourni avec le client Oracle)
- Oracle Instant Client 21.17 (bibliothèques OCI)

Note :

- Les scripts ont été testés sous Linux.
- Les chemins doivent être adaptés selon votre système.

2. Installation du client Oracle (`cx_Oracle`)

1. Télécharger l'Instant Client 21.17 : `https://www.oracle.com/fr/database/technologies/instant-client/linux-x86-64-downloads.html`
2. Extraire dans (pour mon cas) :

```
~/Documents/Stage/ProjetStage[BentabeRania]/Dictionnaire/
   oracle_client/
```

3. Installer `cx_Oracle`, `pandas` et `numpy` :

```
cd ~/Documents/Stage/ProjetStage[BentabeRania]/Dictionnaire
python3 -m pip install cx_Oracle pandas numpy
```

3. Génération du dictionnaire de données Oracle

Objectif : extraire la structure des tables Oracle et générer un CSV.

1. Se placer dans le dossier :

```
cd ~/Documents/Stage/ProjetStage[BentabeRania]/Dictionnaire
```

2. Exporter la variable d'environnement :

```
export LD_LIBRARY_PATH=~/Documents/Stage/ProjetStage[
   BentabeRania]/Dictionnaire/oracle_client/instantclient_21_17:
   $LD_LIBRARY_PATH
```

3. Exécuter :

```
python3 dict.py
```

Résultat attendu :

— Message `Connexion réussie à Oracle !!!`

— Fichier `dictionnaire_donnees.csv` généré

4. Installation de JupyterLab sur Ubuntu

JupyterLab est l'interface recommandée pour exécuter les notebooks Python (.ipynb). Voici comment l'installer sur Ubuntu :

1. Mettre à jour les paquets :

```
sudo apt update
```

2. Installer `pip` et `Python3` (si ce n'est pas déjà fait) :

```
sudo apt install python3-pip python3-dev
```

3. Installer JupyterLab avec `pip` :

```
pip3 install jupyterlab
```

4. Lancer JupyterLab depuis un terminal :

```
jupyter lab
```

Une interface s'ouvre dans le navigateur (généralement sur `http://localhost:8888`).

5. Traitement des données Kaggle (Data.ipynb)

Objectif : nettoyer le fichier `cleansingWine.csv` et générer 3 fichiers CSV.

1. Ouvrir JupyterLab :

```
cd ~/Documents/Stage/ProjetStage[BentabeRania]
jupyter lab
```

2. Ouvrir le notebook `Data.ipynb` puis exécuter toutes les cellules avec `Shift + Entrée`.

3. Trois fichiers CSV sont générés :
   — `vinR.csv`
   — `producteurR.csv`
   — `recolteR.csv`

Alternative : Si vous ne souhaitez pas relancer le notebook, les fichiers sont déjà fournis dans le dossier.

6. Génération de 1 000 000 de lignes de récolte

Objectif : enrichir recolteR.csv en mémoire et créer un fichier .dat pour SQL*Loader.

1. Installer les dépendances (si ce n'est pas déjà fait) :

```
cd ~/Documents/Stage/ProjetStage[BentabeRania]
python3 -m pip install pandas numpy
```

2. Dans JupyterLab, ouvrir le notebook `Script.ipynb`.
3. Exécuter toutes les cellules. Les fichiers produits sont :
   — `recolte_augmente_1M.dat`
   — `recolte_augmente_1M_YYYYMMDD_HHMMSS.csv`

7. Chargement dans Oracle avec SQL*Loader

Objectif : insérer 1 000 000 de lignes dans la table `recolteR`.

1. Utiliser le fichier de contrôle `recolte.ctl` fourni.
2. Exécuter :

```
sqlldr userid=MONUSER/MOTDEPASSE \
  control=recolte.ctl \
  log=recolte.log \
  bad=recolte.bad \
  discard=recolte.dsc \
  readsize=10485760 \
  bindsize=10485760 \
  rows=10000 \
  errors=10
```

3. Après chargement (vérification post-chargement) :

```
DELETE FROM RECOLTER r
WHERE ROWID NOT IN (
  SELECT MIN(ROWID)
  FROM RECOLTER
  GROUP BY Nprod, Nvin, ANNEE
);
```

```
COMMIT;

SELECT COUNT(*) FROM RECOLTER;
```

8. Schéma global d'exécution

```
ProjetStage[BentabeRania]/
       Dictionnaire/
             dict.py
             oracle_client/instantclient_21_17/
             dictionnaire_donnees.csv
       cleansingWine.csv
       Data.ipynb
       vinR.csv
       producteurR.csv
       recolteR.csv
       Script.ipynb
       recolte_augmente_1M.dat
       recolte_augmente_1M_20250605_173431.csv
       recolte.ctl
       [autres scripts   ventuels  ]
```

9. Conseils rapides

— Toujours `TRUNCATE` la table Oracle avant un nouveau chargement.

— Vérifier la variable `LD_LIBRARY_PATH` à chaque session.

— Examiner les fichiers `.bad` et `.log` après chargement.

— Adapter les paramètres `readsize`, `bindsize` selon votre machine.