



Université Chouaib Doukkali
Ecole Nationale des Sciences Appliquées d'El Jadida
Département Télécommunications, Réseaux et Informatique
Module : **MLops**



Filière : **SDIA**
Niveau : **2^{ème} Année**

LAB 1 :

**Lab 1 : Du notebook au mini-système
production-ready**

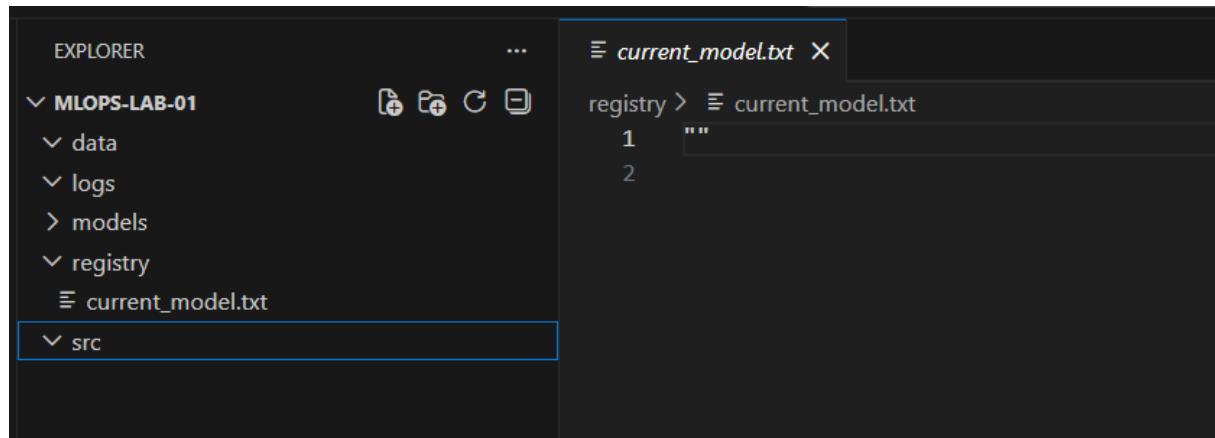
Réalisé Par :
Wassima RAICHI

Année Universitaire : 2025/2026

Étape 1 : Initialiser la structure du projet

```
C:\Users\hp\OneDrive\Bureau\TP MLops>mkdir mlops-lab-01
C:\Users\hp\OneDrive\Bureau\TP MLops>cd mlops-lab-01
C:\Users\hp\OneDrive\Bureau\TP MLops\mlops-lab-01>mkdir data
C:\Users\hp\OneDrive\Bureau\TP MLops\mlops-lab-01>mkdir models
C:\Users\hp\OneDrive\Bureau\TP MLops\mlops-lab-01>mkdir registry
C:\Users\hp\OneDrive\Bureau\TP MLops\mlops-lab-01>mkdir logs
C:\Users\hp\OneDrive\Bureau\TP MLops\mlops-lab-01>mkdir src
C:\Users\hp\OneDrive\Bureau\TP MLops\mlops-lab-01>echo "" > registry\current_model.txt
C:\Users\hp\OneDrive\Bureau\TP MLops\mlops-lab-01>tree
Structure du dossier
Le numéro de série du volume est 6CED-D509
C:.
├── data
├── logs
├── models
└── registry
    └── current_model.txt

C:\Users\hp\OneDrive\Bureau\TP MLops\mlops-lab-01>
```



Étape 2 : Préparer l'environnement Python

```
C:\Users\hp\OneDrive\Bureau\TP MLops\mlops-lab-01>python -m venv venv_mlops
```



```
PS C:\Users\hp\OneDrive\Bureau\TP MLops\mlops-lab-01> .\venv_mlops\Scripts\Activate.ps1
(venv_mlops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mlops-lab-01> |
```



```
(venv_mlops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mlops-lab-01> pip install --upgrade pip
Requirement already satisfied: pip in c:\users\hp\onedrive\bureau\tp mllops\mllops-lab-01\venv_mllops\lib\site-packages (20.1)
Collecting pip
  Obtaining dependency information for pip from https://files.pythonhosted.org/packages/44/3c/d717024885424591d5376226e836c2d5293ce2011523c9de23ff7bf068/pip-25.3-py3-none-any.whl.metadata
    Using cached pip-25.3-py3-none-any.whl.metadata (4.7 kB)
  Using cached pip-25.3-py3-none-any.whl (1.8 MB)
ERROR: To modify pip, please run the following command:
C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01\venv_mllops\Scripts\python.exe -m pip install --upgrade pip

[notice] A new release of pip is available: 23.2.1 → 25.3
[notice] To update, run: python.exe -m pip install --upgrade pip
```

```
(venv_mllops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01> pip install pandas numpy scikit-learn fastapi uvicorn
Collecting pandas
  Obtaining dependency information for pandas from https://files.pythonhosted.org/packages/8e/59/712db1d7040520de7a4965df15b774348980e6df45c129b8c64d0dbe74ef/pandas-2.3.3-cp311-cp311-win_amd64.whl.metadata
    Using cached pandas-2.3.3-cp311-cp311-win_amd64.whl.metadata (19 kB)
Collecting numpy
  Obtaining dependency information for numpy from https://files.pythonhosted.org/packages/aa/44/9fe81aeldcc29c531843852e2874080dc441338574ccc4306b39e2ff6e59/numpy-2.3.5-cp311-cp311-win_amd64.whl.metadata
    Using cached numpy-2.3.5-cp311-cp311-win_amd64.whl.metadata (60 kB)
Collecting scikit-learn
  Obtaining dependency information for scikit-learn from https://files.pythonhosted.org/packages/89/3c/45c352094cf6a60050bcb967b1faf246b22e93cb459f2f907b600f2ceda5/scikit_learn-1.8.0-cp311-cp311-win_amd64.whl.metadata
    Using cached scikit_learn-1.8.0-cp311-cp311-win_amd64.whl.metadata (11 kB)
Collecting fastapi
  Obtaining dependency information for fastapi from https://files.pythonhosted.org/packages/3e/57/aa70121b5008f44031be645a61a7c4abc24e0e888ad3fc8fd916f4d188e/fastapi-0.124.4-py3-none-any.whl.metadata
    Using cached fastapi-0.124.4-py3-none-any.whl.metadata (30 kB)
Collecting uvicorn
  Obtaining dependency information for uvicorn from https://files.pythonhosted.org/packages/ee/d9/d88e73ca598f4f6ff671fb5fde8a32925c2e08a637303a1d12883c7305fa/uvicorn-0.38.0-py3-none-any.whl.metadata
    Using cached uvicorn-0.38.0-py3-none-any.whl.metadata (6.8 kB)
Collecting joblib
  Obtaining dependency information for joblib from https://files.pythonhosted.org/packages/1e/e8/685f47e0d754320684db4425a0967f7d3fa70126bffd76110b7009a0090f/joblib-1.5.2-py3-none-any.whl.metadata
    Using cached joblib-1.5.2-py3-none-any.whl.metadata (5.6 kB)
Collecting python-dateutil>=2.8.2 (from pandas)
  Obtaining dependency information for python-dateutil>=2.8.2 from https://files.pythonhosted.org/packages/ec/57/56b9bcc3c9c6a792fcba139543cee77261f3651ca9da0c93f5c1221264b/python_dateutil-2.9.0.post0-py2.py3-none-any.whl.metadata
    Using cached python_dateutil-2.9.0.post0-py2.py3-none-any.whl.metadata (8.4 kB)
```

Étape 3 : Générer le dataset

```
(venv_mllops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01> New-Item src/generate_data.py
```

Répertoire : C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01\src

Mode	LastWriteTime	Length	Name
-a----	14/12/2025	16:56	0 generate_data.py

```
(venv_mllops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01> python src/generate_data.py
[OK] Dataset généré : C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01\data\raw.csv (rows=1200, seed=42)
(venv_mllops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01>
```

```

EXPLORER
MLOPS-LAB-01
  data
    raw.csv
  logs
  models
  registry
    current_model.txt
  src
    generate_data.py <-- selected
    venv_mllops
      Include
      Lib
      Scripts
    pyvenv.cfg

generate_data.py X raw.csv
src > generate_data.py > ...
1  from pathlib import Path
2  from typing import Final
3
4
5  import numpy as np
6  import pandas as pd
7
8  """
9  Module de génération d'un dataset synthétique de churn client.
10
11
12 Ce script génère un fichier CSV `data/raw.csv` contenant des données
13 synthétiques d'abonnement (tenure, plaintes, usage, etc.) et une variable
14 binaire `churn` indiquant si le client quitte le service.
15
16
17 Il est pensé comme point de départ pour un lab MLOps :
18 - jeu de données contrôlé et reproduit ;
19 - logique métier simple mais réaliste ;
20 - génération déterministe grâce à une graine pseudo-aléatoire.
21 """
22
23
24
25 # -----
26 # Constantes de chemin
27 # -----
28
29
30 ROOT: Final[Path] = Path(__file__).resolve().parents[1]
31 DATA_DIR: Final[Path] = ROOT / "data"
32 RAW_PATH: Final[Path] = DATA_DIR / "raw.csv"
33
```

```

EXPLORER ... generate_data.py raw.csv X
MLOPS-LAB-01 ...
  data
    raw.csv
  logs
  models
  registry
    current_model.txt
  src
    generate_data.py
  venv_mlops
    Include
    Lib
    Scripts
  pyvenv.cfg

data > raw.csv > data
1 tenure_months,num_complaints,avg_session_minutes,plan_type,region,churn
2 6,3,21.5,basic,AF,1
3 46,3,50.19,basic,EU,0
4 39,2,47.79,basic,AF,0
5 26,3,41.45,premium,AF,1
6 26,2,58.67,premium,EU,0
7 51,0,32.73,basic,AF,0
8 6,0,71.38,basic,NA,0
9 42,3,32.27,basic,AF,1
10 12,3,38.06,basic,NA,1
11 6,4,30.23,basic,EU,1
12 32,1,39.47,basic,NA,0
13 58,0,45.03,premium,AS,1
14 44,4,33.63,premium,EU,1
15 45,1,56.32,basic,AF,1
16 43,0,37.87,basic,EU,0
17 47,2,38.25,basic,EU,1
18 31,2,24.09,basic,NA,1
19 8,2,40.02,basic,AS,1
20 50,0,33.39,basic,AF,0
21 27,1,27.7,basic,NA,1
22 30,1,25.68,premium,AF,1
23 22,1,67.06,basic,EU,0
24 11,0,31.84,basic,NA,0
25 55,1,17.52,basic,EU,0
26 47,0,26.62,premium,NA,0
27 38,1,34.38,basic,NA,1
28 24,2,41.56,basic,AF,1
29 49,1,41.66,basic,EU,0
30 33,3,43.12,basic,AS,1
31 27,2,27.26,premium,AF,1
32 27,0,39.09,basic,AS,0
33 14,1,24.49,premium,AF,0
34 6,0,27.56,basic,AS,1
35 33,0,23.99,basic,EU,0
36 53,1,20.74,basic,EU,0
37 4,1,47.77,basic,EU,1

```

Étape 4 : Préparer les données + quality checks

```
(venv_mlops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mlops-lab-01> New-Item src/prepare_data.py
```

```
Répertoire : C:\Users\hp\OneDrive\Bureau\TP MLops\mlops-lab-01\src
```

Mode	LastWriteTime	Length	Name
-a----	14/12/2025	17:39	0 prepare_data.py

```
(venv_mlops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mlops-lab-01> |
```

```
(venv_mlops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mlops-lab-01> python src/prepare_data.py
[OK] Fichier prétraité généré : C:\Users\hp\OneDrive\Bureau\TP MLops\mlops-lab-01\data\processed.csv
[OK] Statistiques d'entraînement générées : C:\Users\hp\OneDrive\Bureau\TP MLops\mlops-lab-01\registry\train_stats.json
(venv_mlops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mlops-lab-01> |
```

EXPLORER

... MIOPS-LAB-01

- data
 - processed.csv
 - raw.csv
- logs
- models
- registry
 - current_model.txt
 - train_stats.json
- src
 - generate_data.py
 - prepare_data.py
- venv_mllops
 - Include
 - Lib
 - Scripts
- pyenv.cfg

OUTLINE

generate_data.py prepare_data.py

```

2
3 """
4 Module de prétraitement des données de churn.
5
6 Ce script :
7 1. Charge le fichier brut `data/raw.csv` ;
8 2. Applique des règles de nettoyage simples :
9   - clip des valeurs négatives sur `avg_session_minutes` ;
10  - normalisation des champs catégoriels (`plan_type`, `region`) ;
11 3. Exécute des contrôles qualité (schéma, taux de valeurs manquantes,
12  type des colonnes numériques) ;
13 4. Sauvegarde :
14   - un fichier nettoyé `data/processed.csv` ;
15   - un fichier `registry/train_stats.json` contenant moyenne et
16     écart-type des variables numériques (pour normalisation ultérieure).
17
18 Ce module est typiquement utilisé comme étape "prétraitement" d'un pipeline
19 MLOps avant l'entraînement du modèle.
20 """
21
22 from pathlib import Path
23 from typing import Final
24
25 import json
26 import pandas as pd
27
28 # -----
29 # Constantes de chemin
30 # -----
31
32 ROOT: Final[Path] = Path(__file__).resolve().parents[1]
33 DATA_DIR: Final[Path] = ROOT / "data"
34 RAW_PATH: Final[Path] = DATA_DIR / "raw.csv"
35 PROCESSED_PATH: Final[Path] = DATA_DIR / "processed.csv"
36 TRAIN_STATS_PATH: Final[Path] = ROOT / "registry" / "train_stats.json"
37 # -----

```

EXPLORER

... MIOPS-LAB-01

- data
 - processed.csv
 - raw.csv
- logs
- models
- registry
 - current_model.txt
 - train_stats.json
- src
 - generate_data.py
 - prepare_data.py
- venv_mllops
 - Include
 - Lib
 - Scripts
- pyenv.cfg

OUTLINE

generate_data.py prepare_data.py processed.csv

```

data > processed.csv > data
1 tenure_months,num_complaints,avg_session_minutes,plan_type,region,churn
2 6,3,21.5,basic,AF,1
3 46,3,50.19,basic,EU,0
4 39,2,47.79,basic,AF,0
5 26,3,41.45,premium,AF,1
6 26,2,58.67,premium,EU,0
7 51,0,32.73,basic,AF,0
8 6,0,71.38,basic,NAN,0
9 42,3,32.27,basic,AF,1
10 12,3,38.06,basic,NAN,1
11 6,4,30.23,basic,EU,1
12 32,1,39.47,basic,NAN,0
13 58,0,45.03,premium,AS,1
14 44,4,33.63,premium,EU,1
15 45,1,56.32,basic,AF,1
16 43,0,37.87,basic,EU,0
17 47,2,38.25,basic,EU,1
18 31,2,24.09,basic,NAN,1
19 8,2,40.02,basic,AS,1
20 50,0,33.39,basic,AF,0
21 27,1,27.7,basic,NAN,1
22 30,1,25.68,premium,AF,1
23 22,1,67.06,basic,EU,0
24 11,0,31.84,basic,NAN,0
25 55,1,17.52,basic,EU,0
26 47,0,26.62,premium,NAN,0
27 38,1,34.38,basic,NAN,1
28 24,2,41.56,basic,AF,1
29 49,1,41.66,basic,EU,0
30 33,3,43.12,basic,AS,1
31 27,2,27.26,premium,AF,1
32 27,0,39.09,basic,AS,0
33 14,1,24.49,premium,AF,0
34 6,0,27.56,basic,AS,1
35 33,0,23.99,basic,EU,0
36 53,1,20.74,basic,EU,0

```

```

EXPLORER ... generate_data.py prepare_data.py train_stats.json
MLOPS-LAB-01 ...
  data
    processed.csv
    raw.csv
  logs
  models
  registry
    current_model.txt
  src
    generate_data.py
    prepare_data.py
  venv_mllops
    Include
    Lib
    Scripts
  pyvenv.cfg
  train_stats.json

```

```

registry > train_stats.json > ...
1  {
2    "tenure_months": {
3      "mean": 30.24583333333334,
4      "std": 16.97263851086721
5    },
6    "num_complaints": {
7      "mean": 1.174166666666666,
8      "std": 1.111993591263636
9    },
10   "avg_session_minutes": {
11     "mean": 35.123675000000006,
12     "std": 11.837695289590059
13   }
14 }

```

Étape 5 : Entraîner, versionner et valider le modèle

```

[venv_mllops] PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01> New-Item src/train.py

Répertoire : C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01\src

Mode          LastWriteTime      Length Name
----          -----          ---- -
-a---  14/12/2025      17:43          0 train.py

(venv_mllops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01> |

```

```

EXPLORER ... generate_data.py prepare_data.py train.py
MLOPS-LAB-01 ...
  data
    processed.csv
    raw.csv
  logs
  models
  registry
    current_model.txt
  src
    generate_data.py
    prepare_data.py
  train.py
  venv_mllops
    Include
    Lib
    Scripts
  pyvenv.cfg
  train_stats.json

```

```

src > train.py > ...
1  from __future__ import annotations
2
3  """
4  Module d'entraînement et d'enregistrement d'un modèle de churn.
5
6  Ce script :
7  1. Charge le jeu de données prétraité `data/processed.csv` ;
8  2. Sépare les variables explicatives (features) de la cible `churn` ;
9  3. Définit un pipeline scikit-learn :
10   - prétraitement (StandardScaler pour les variables numériques,
11     | OneHotEncoder pour les catégorielles) ;
12   - modèle de régression logistique ;
13 4. Coupe les données en train / test ;
14 5. Entraîne le modèle, évalue les métriques (accuracy, precision, recall, F1) ;
15 6. Compare la F1 à une baseline "bête" (prédire toujours 0) ;
16 7. Sauvegarde :
17   - le modèle entraîné dans le dossier `models/` ;
18   - les métadonnées d'entraînement (métriques, seed, version, etc.)
19     | dans `registry/metadata.json` ;
20   - le fichier `registry/current_model.txt` si le modèle passe le gate.
21
22 Ce module illustre une étape typique "Train + Register" d'un pipeline MLOps
23 minimalistique, avec un premier niveau de gouvernance via une F1 minimale.
24 """
25
26 from datetime import datetime
27 from pathlib import Path
28 from typing import Any, Final
29
30 import json
31
32 import joblib
33 import pandas as pd
34 from sklearn.compose import ColumnTransformer
35 from sklearn.linear_model import LogisticRegression
36 from sklearn.metrics import (

```

```
(venv_mllops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01> python src/train.py
[METRICS] {
    "accuracy": 0.6433333333333333,
    "precision": 0.6687898089171974,
    "recall": 0.65625,
    "f1": 0.6624605678233438,
    "baseline_f1": 0.0
}
[OK] Modèle sauvegardé : C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01\models\churn_model_v1_20251214_164431.joblib
[DEPLOY] Refusé par le gate : F1 insuffisante ou baseline non battue.
(venv_mllops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01> |
```

```
(venv_mllops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01> dir models
```

Répertoire : C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01\models

Mode	LastWriteTime	Length	Name
-a----	14/12/2025 17:44	3690	churn_model_v1_20251214_164431.joblib
-a----	14/12/2025 17:47	3690	churn_model_v1_20251214_164716.joblib

```
(venv_mllops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01> |
```

```
(venv_mllops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01> dir registry
```

Répertoire : C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01\registry

Mode	LastWriteTime	Length	Name
-a----	14/12/2025 17:47	37	current_model.txt
-a----	14/12/2025 17:47	927	metadata.json
-a----	14/12/2025 17:40	284	train_stats.json

```
(venv_mllops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01> |
```

```
(venv_mllops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01> type registry\metadata.json
```

```
[{"model_file": "churn_model_v1_20251214_164431.joblib",
"version": "v1",
"trained_at_utc": "20251214_164431",
"data_file": "processed.csv",
"seed": 42,
"metrics": {
    "accuracy": 0.6433333333333333,
    "precision": 0.6687898089171974,
    "recall": 0.65625,
    "f1": 0.6624605678233438,
    "baseline_f1": 0.0
},
"gate_f1": 0.7,
"passed_gate": false
},
{
"model_file": "churn_model_v1_20251214_164716.joblib",
"version": "v1",
"trained_at_utc": "20251214_164716",
"data_file": "processed.csv",
"seed": 42,
"metrics": {
    "accuracy": 0.6433333333333333,
    "precision": 0.6687898089171974,
    "recall": 0.65625,
```

```

    "metrics": {
        "accuracy": 0.6433333333333333,
        "precision": 0.6687898089171974,
        "recall": 0.65625,
        "f1": 0.6624605678233438,
        "baseline_f1": 0.0
    },
    "gate_f1": 0.7,
    "passed_gate": false
},
{
    "model_file": "churn_model_v1_20251214_164716.joblib",
    "version": "v1",
    "trained_at_utc": "20251214_164716",
    "data_file": "processed.csv",
    "seed": 42,
    "metrics": {
        "accuracy": 0.6433333333333333,
        "precision": 0.6687898089171974,
        "recall": 0.65625,
        "f1_threshold_05": 0.6624605678233438,
        "f1": 0.7164179104477612,
        "best_threshold": 0.36,
        "baseline_f1": 0.0
    },
    "gate_f1": 0.7,
    "passed_gate": true
}
]
(venv_mllops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01> |

```

Étape 6 : Inspecter la registry et le modèle courant

```

(venv_mllops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01> New-Item src/evaluate.py

Répertoire : C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01\src

Mode          LastWriteTime      Length Name
----          -----          ---- -
-a---  14/12/2025      17:46           0 evaluate.py

(venv_mllops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01> |

```

```

EXPLORER ... generate_data.py prepare_data.py train.py evaluate.py X
MLOPS-LAB-01
  data
    processed.csv
    raw.csv
  logs
  models
  registry
    current_model.txt
  metadata.json
  train_stats.json
  src
    evaluate.py
    generate_data.py
    prepare_data.py
    train.py
  venv_mllops
    Include
    Lib
    Scripts
    pyvenv.cfg

```

```

src > evaluate.py ...
1  from __future__ import annotations
2
3
4 """
5   Module d'entraînement d'un modèle de churn avec tuning du seuil optimal (F1).
6
7   Ce script :
8     1. Charge le dataset prétraité `data/processed.csv` ;
9     2. Crée un pipeline scikit-learn :
10       - Standardisation des variables numériques,
11       - OneHotEncoding des variables catégorielles,
12       - Régression logistique ;
13     3. Découpe train/test avec stratification ;
14     4. Entraîne le modèle ;
15     5. Calcule :
16       - Les métriques standard avec seuil = 0.5,
17       - Le seuil optimal maximisant la F1,
18       - Une baseline triviale (prédir toujours 0) ;
19     6. Sauvegarde :
20       - Le modèle dans `models/` ,
21       - Les métadonnées dans `registry/metadata.json` ,
22       - Le modèle courant dans `registry/current_model.txt` si le gate F1 est validé.
23
24
25
26 Ce script illustre une étape "Train + Eval + Register" comme dans un vrai pipeline MLOps,
27 avec un composant de tuning simple mais efficace : optimisation du seuil.
28 """
29
30
31 import json
32 from datetime import datetime
33 from pathlib import Path
34 from typing import Any, Final
35
36
37 import joblib

```

> OUTLINE
> TIMELINE

```
(venv_mllops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01> python src/evaluate.py
[METRICS] {
  "accuracy": 0.6433333333333333,
  "precision": 0.6687898089171974,
  "recall": 0.65625,
  "f1_threshold_05": 0.6624605678233438,
  "f1": 0.7164179104477612,
  "best_threshold": 0.36,
  "baseline_f1": 0.0
}
[OK] Modèle sauvgardé : C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01\models\churn_model_v1_20251214_164716.joblib
[DEPLOY] Modèle activé : churn_model_v1_20251214_164716.joblib
(venv_mllops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01>
```

```
(venv_mllops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01> type registry\metadata.json
[
  {
    "model_file": "churn_model_v1_20251214_164431.joblib",
    "version": "v1",
    "trained_at_utc": "20251214_164431",
    "data_file": "processed.csv",
    "seed": 42,
    "metrics": {
      "accuracy": 0.6433333333333333,
      "precision": 0.6687898089171974,
      "recall": 0.65625,
      "f1": 0.6624605678233438,
      "baseline_f1": 0.0
    },
    "gate_f1": 0.7,
    "passed_gate": false
  },
  {
    "model_file": "churn_model_v1_20251214_164716.joblib",
    "version": "v1",
    "trained_at_utc": "20251214_164716",
    "data_file": "processed.csv",
    "seed": 42,
    "metrics": {
      "accuracy": 0.6433333333333333,
      "precision": 0.6687898089171974,
      "recall": 0.65625,
```

```

    "metrics": {
        "accuracy": 0.6433333333333333,
        "precision": 0.6687898089171974,
        "recall": 0.65625,
        "f1": 0.6624605678233438,
        "baseline_f1": 0.0
    },
    "gate_f1": 0.7,
    "passed_gate": false
},
{
    "model_file": "churn_model_v1_20251214_164716.joblib",
    "version": "v1",
    "trained_at_utc": "20251214_164716",
    "data_file": "processed.csv",
    "seed": 42,
    "metrics": {
        "accuracy": 0.6433333333333333,
        "precision": 0.6687898089171974,
        "recall": 0.65625,
        "f1_threshold_05": 0.6624605678233438,
        "f1": 0.7164179104477612,
        "best_threshold": 0.36,
        "baseline_f1": 0.0
    },
    "gate_f1": 0.7,
    "passed_gate": true
}
]

```

```
(venv_mllops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01> type registry\current_model.txt
churn_model_v1_20251214_164716.joblib
(venv_mllops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01> |
```

Étape 7 : Créer une API /predict qui utilise le modèle courant

```
(venv_mllops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01> New-Item src/api.py

Répertoire : C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01\src

Mode          LastWriteTime      Length Name
----          -----          ---- 
-a---  14/12/2025     17:52           0 api.py

(venv_mllops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01> |
```

```

EXPLORER          ...
src > api.py > predict
1   from __future__ import annotations
2
3
4 """
5 API FastAPI de prédiction de churn pour le lab MLOps.
6
7
8 Ce service :
9 - charge dynamiquement le modèle courant indiqué dans `registry/current_model.txt` ;
10 - expose un endpoint `/health` pour vérifier l'état de l'API et du modèle ;
11 - expose un endpoint `/predict` pour faire une prédiction de churn à partir
12 de features simples (tenure, plaintes, durée de session, type d'abonnement,
région) ;
13 - journalise chaque requête de prédiction dans `logs/predictions.log` au
14 format JSON (une ligne par prédiction).
15
16
17 Cette API illustre une étape "Serve" dans un pipeline MLOps minimal :
18 un modèle versionné est promu côté registry, puis utilisé par un service
d'inférence léger.
19 """
20
21
22
23
24 import json
25 import time
26 from pathlib import Path
27 from typing import Any, Optional
28
29
30 import joblib
31 import pandas as pd
32 from fastapi import FastAPI, HTTPException
33 from pydantic import BaseModel, Field
34
35
36 # -----
37 # Constantes de chemin

```

```
(venv_mlops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01> uvicorn src.api:app --reload
INFO: Will watch for changes in these directories: ['C:\\\\Users\\\\hp\\\\OneDrive\\\\Bureau\\\\TP MLops\\\\mllops-lab-01']
INFO: Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO: Started reloader process [22868] using StatReload
INFO: Started server process [10376]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: 127.0.0.1:64322 - "GET /health HTTP/1.1" 200 OK
INFO: 127.0.0.1:64322 - "GET /favicon.ico HTTP/1.1" 404 Not Found
INFO: 127.0.0.1:56041 - "GET /predict HTTP/1.1" 405 Method Not Allowed
INFO: 127.0.0.1:51897 - "POST /predict HTTP/1.1" 200 OK
INFO: 127.0.0.1:51901 - "POST /predict HTTP/1.1" 200 OK
INFO: 127.0.0.1:52473 - "GET /health HTTP/1.1" 200 OK
INFO: 127.0.0.1:52473 - "GET /predict HTTP/1.1" 405 Method Not Allowed
INFO: 127.0.0.1:52473 - "GET /predict HTTP/1.1" 405 Method Not Allowed
```

```
(venv_mlops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01> Invoke-RestMethod -Method Post -Uri http://127.0.0.1:8000/predict `
>> -ContentType "application/json" `
>> -Body '{
>>   "tenure_months": 6,
>>   "num_complaints": 3,
>>   "avg_session_minutes": 12.5,
>>   "plan_type": "basic",
>>   "region": "AF",
>>   "request_id": "req-001"
>> }'

request_id : req-001
model_version : churn_model_v1_20251214_165918.joblib
prediction : 1
probability : 0,907065
latency_ms : 5,834
features : @{tenure_months=6; num_complaints=3; avg_session_minutes=12,5; plan_type=basic; region=AF}
ts : 1765731776
```

```
(venv_mllops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01> Invoke-RestMethod -Method Post -Uri http://127.0.0.1:8000/predict
>> -ContentType "application/json" `
>> -Body `{
>>   "tenure_months": 48,
>>   "num_complaints": 0,
>>   "avg_session_minutes": 60,
>>   "plan_type": "premium",
>>   "region": "EU",
>>   "request_id": "req-safe"
>> }` 

request_id      : req-safe
model_version  : churn_model_v1_20251214_165918.joblib
prediction      : 0
probability    : 0,139973
latency_ms     : 6,982
features       : @{tenure_months=48; num_complaints=0; avg_session_minutes=60; plan_type=premium; region=EU}
ts              : 1765731856
```

```
(venv_mllops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01> type logs\predictions.log
>{"request_id": "req-001", "model_version": "churn_model_v1_20251214_164716.joblib", "prediction": 1, "probability": 0.907065, "latency_ms": 5.964, "features": {"tenure_months": 6, "num_complaints": 3, "avg_session_minutes": 12.5, "plan_type": "basic", "region": "AF"}, "ts": 1765731362}
>{"request_id": "req-safe", "model_version": "churn_model_v1_20251214_164716.joblib", "prediction": 0, "probability": 0.139973, "latency_ms": 5.818, "features": {"tenure_months": 48, "num_complaints": 0, "avg_session_minutes": 60.0, "plan_type": "premium", "region": "EU"}, "ts": 1765731385}
>{"request_id": "req-001", "model_version": "churn_model_v1_20251214_165918.joblib", "prediction": 1, "probability": 0.907065, "latency_ms": 5.834, "features": {"tenure_months": 6, "num_complaints": 3, "avg_session_minutes": 12.5, "plan_type": "basic", "region": "AF"}, "ts": 1765731776}
>{"request_id": "req-safe", "model_version": "churn_model_v1_20251214_165918.joblib", "prediction": 0, "probability": 0.139973, "latency_ms": 6.982, "features": {"tenure_months": 48, "num_complaints": 0, "avg_session_minutes": 60.0, "plan_type": "premium", "region": "EU"}, "ts": 1765731856}
(venv_mllops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01> |
```

```
(venv_mllops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01> python src/evaluate.py
[METRICS] {
  "accuracy": 0.6433333333333333,
  "precision": 0.6687898089171974,
  "recall": 0.65625,
  "f1_threshold_05": 0.6624605678233438,
  "f1": 0.7164179104477612,
  "best_threshold": 0.36,
  "baseline_f1": 0.0
}
[OK] Modèle sauvegardé : C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01\models\churn_model_v1_20251214_170512.joblib
[DEPLOY] Modèle activé : churn_model_v1_20251214_170512.joblib
(venv_mllops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01> |
```

Étape 8 : Déetecter une dérive des données via les logs

```
PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01> New-Item src/monitor_drift.py

Répertoire : C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01\src

Mode          LastWriteTime          Length Name
----          -----          ---- 
-a---  14/12/2025      18:00            0 monitor_drift.py

PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01> |
```

```

EXPLORER
...
MLOPS-LAB-01
  data
    processed.csv
    raw.csv
  logs
    predictions.log
  models
  registry
    current_model.txt
    metadata.json
    train_stats.json
  src
    __pycache__
    api.py
    evaluate.py
    generate_data.py
    monitor_drift.py
    prepare_data.py
    train.py
  venv_mllops
    Include
    Lib
    Scripts
    pyvenv.cfg

  OUTLINE
  TIMELINE
  0:0 0:0

monitor_drift.py
...
src > monitor_drift.py > ...
1   from __future__ import annotations
2
3
4 """
5   Script de détection simple de data drift sur les features d'entrée.
6
7
8 Ce script :
9 1. Charge les statistiques d'entraînement (moyenne / écart-type) depuis
10 | `registry/train_stats.json` (généré par `prepare_data.py`) ;
11 2. Charge les requêtes de prédition récentes depuis `logs/predictions.log`
12 | (une prédition par ligne au format JSON) ;
13 3. Compare la moyenne des features observées en production aux moyennes
14 | d'entraînement à l'aide d'un score Z :
15 |   z = |mean_prod - mean_train| / std_train
16 4. Déclenche une alerte si z >= z_threshold pour au moins une feature ;
17 5. Logge un message indiquant la possibilité d'envoyer l'alerte vers
18 | un outil de monitoring externe (si `MONITORING_TOKEN` est défini).
19
20
21 Ce script illustre un mécanisme de monitoring très simple mais pédagogique
22 pour un lab MLOps : détecter un drift sur les distributions des features
23 d'entrée, en se basant sur les moyennes.
24 """
25
26
27 import json
28 import os
29 from pathlib import Path
30 from typing import Final
31
32
33 import pandas as pd
34
35
36 # -----
37 # Configuration & chemins

```

```

PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01> python src/monitor_drift.py
== Drift check sur 2 requêtes récentes ==
- tenure_months: mean_prod=27.000 | mean_train=30.246 | z=0.191
- num_complaints: mean_prod=1.500 | mean_train=1.174 | z=0.293
- avg_session_minutes: mean_prod=36.250 | mean_train=35.124 | z=0.095
Résultat : aucun drift détecté.
PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01>

```

Étape 9 : Gérer les versions du modèle et revenir en arrière

```

(vENV_mllops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01> python -c "from src.train import main; main(version='v2', gate_f1=0.70)"
[METRICS] {
  "accuracy": 0.6433333333333333,
  "precision": 0.6687898089171974,
  "recall": 0.65625,
  "f1": 0.6624605678233438,
  "baseline_f1": 0.0
}
[OK] Modèle sauvegardé : C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01\models\churn_model_v2_20251214_170845.joblib
[DEPLOY] Refusé par le gate : F1 insuffisante ou baseline non battue.
(vENV_mllops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01>

```

```
(venv_mllops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01> type registry\metadata.json
```

```
[
```

```
{
```

```
    "model_file": "churn_model_v1_20251214_164431.joblib",  
    "version": "v1",  
    "trained_at_utc": "20251214_164431",  
    "data_file": "processed.csv",  
    "seed": 42,  
    "metrics": {  
        "accuracy": 0.6433333333333333,  
        "precision": 0.6687898089171974,  
        "recall": 0.65625,  
        "f1": 0.6624605678233438,  
        "baseline_f1": 0.0  
    },  
    "gate_f1": 0.7,  
    "passed_gate": false  
},  
{
```

```
    "model_file": "churn_model_v1_20251214_164716.joblib",  
    "version": "v1",  
    "trained_at_utc": "20251214_164716",  
    "data_file": "processed.csv",  
    "seed": 42,  
    "metrics": {  
        "accuracy": 0.6433333333333333,  
        "precision": 0.6687898089171974,  
        "recall": 0.65625,  
        "f1_threshold_05": 0.6624605678233438,
```

```
        "f1": 0.7164179104477612,  
        "best_threshold": 0.36,  
        "baseline_f1": 0.0  
    },  
    "gate_f1": 0.7,  
    "passed_gate": true  
},  
{
```

```
    "model_file": "churn_model_v1_20251214_165918.joblib",  
    "version": "v1",  
    "trained_at_utc": "20251214_165918",  
    "data_file": "processed.csv",  
    "seed": 42,  
    "metrics": {  
        "accuracy": 0.6433333333333333,  
        "precision": 0.6687898089171974,  
        "recall": 0.65625,  
        "f1_threshold_05": 0.6624605678233438,  
        "f1": 0.7164179104477612,  
        "best_threshold": 0.36,  
        "baseline_f1": 0.0  
    },  
    "gate_f1": 0.7,  
    "passed_gate": true  
},  
{
```

```
    "model_file": "churn_model_v1_20251214_170512.joblib",  
    "version": "v1",  
    "trained_at_utc": "20251214_170512",  
    "data_file": "processed.csv",
```

```
"seed": 42,
"metrics": {
    "accuracy": 0.6433333333333333,
    "precision": 0.6687898089171974,
    "recall": 0.65625,
    "f1_threshold_05": 0.6624605678233438,
    "f1": 0.7164179104477612,
    "best_threshold": 0.36,
    "baseline_f1": 0.0
},
"gate_f1": 0.7,
"passed_gate": true
},
{
    "model_file": "churn_model_v2_20251214_170845.joblib",
    "version": "v2",
    "trained_at_utc": "20251214_170845",
    "data_file": "processed.csv",
    "seed": 42,
    "metrics": {
        "accuracy": 0.6433333333333333,
        "precision": 0.6687898089171974,
        "recall": 0.65625,
        "f1": 0.6624605678233438,
        "baseline_f1": 0.0
    },
    "gate_f1": 0.7,
    "passed_gate": false
}
```

```
"metrics": {
    "accuracy": 0.6433333333333333,
    "precision": 0.6687898089171974,
    "recall": 0.65625,
    "f1_threshold_05": 0.6624605678233438,
    "f1": 0.7164179104477612,
    "best_threshold": 0.36,
    "baseline_f1": 0.0
},
"gate_f1": 0.7,
"passed_gate": true
},
{
"model_file": "churn_model_v2_20251214_170845.joblib",
"version": "v2",
"trained_at_utc": "20251214_170845",
"data_file": "processed.csv",
"seed": 42,
"metrics": {
    "accuracy": 0.6433333333333333,
    "precision": 0.6687898089171974,
    "recall": 0.65625,
    "f1": 0.6624605678233438,
    "baseline_f1": 0.0
},
"gate_f1": 0.7,
"passed_gate": false
}
]
(venv_mllops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01> |
```

```
(venv_mllops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01> New-Item src/rollback.py

Répertoire : C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01\src

Mode          LastWriteTime        Length Name
----          -----          ----  --
-a---  14/12/2025      18:09           0  rollback.py

(venv_mllops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mllops-lab-01> |
```

EXPLORER

- ✓ MLOPS-LAB-01
 - ✓ data
 - processed.csv
 - raw.csv
 - ✓ logs
 - predictions.log
 - > models
 - ✓ registry
 - current_model.txt
 - metadata.json
 - train_stats.json
 - ✓ src
 - > __pycache__
 - api.py
 - evaluate.py
 - generate_data.py
 - monitor_drift.py
 - prepare_data.py
 - rollback.py
 - train.py
 - ✓ venv_mlops
 - Include
 - Lib
 - Scripts
 - pyvenv.cfg

... prepare_data.py train.py evaluate.py api.py monitor_drift.py rollback.py ▶ ×

```

src > rollback.py > ...
1   from __future__ import annotations
2
3
4 """
5 Script utilitaire de gestion du registry de modèles.
6
7
8 Objectif principal :
9 - Lister les modèles connus dans `registry/metadata.json` ;
10 - Permettre de changer le modèle courant (`current_model.txt`) vers :
11   - un modèle spécifique (passé via `target`),
12   - ou, par défaut, le modèle précédent (rollback d'une version).
13
14
15 Typiquement utilisé comme :
16 - un outil de rollback simple après un déploiement raté ;
17 - un mécanisme pédagogique pour illustrer le "model registry" en MLOps.
18 """
19
20
21 import json
22 from pathlib import Path
23 from typing import Final, Optional
24
25
26 # -----
27 # Chemins et constantes
28 # -----
29
30
31 ROOT: Final[Path] = Path(__file__).resolve().parents[1]
32 REGISTRY_DIR: Final[Path] = ROOT / "registry"
33 CURRENT_MODEL_PATH: Final[Path] = REGISTRY_DIR / "current_model.txt"
34 METADATA_PATH: Final[Path] = REGISTRY_DIR / "metadata.json"
35
36
37

```

In 157 Col 11 - Sources 4

```
(venv_mlops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mlops-lab-01> python src/rollback.py
[OK] rollback / activation => current_model = churn_model_v1_20251214_170512.joblib
(venv_mlops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mlops-lab-01> |
```

```
(venv_mlops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mlops-lab-01> python -c "from src.rollback import main; main('churn_model_v1_20251214_164431.joblib')"
[OK] rollback / activation => current_model = churn_model_v1_20251214_164431.joblib
(venv_mlops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mlops-lab-01> |
```

```
(venv_mlops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mlops-lab-01> type registry\current_model.txt
churn_model_v1_20251214_164431.joblib
(venv_mlops) PS C:\Users\hp\OneDrive\Bureau\TP MLops\mlops-lab-01> |
```