

# MLFLOW

## Etape 1. Préparation de l'environnement

1. Mise en place d'un environnement ubuntu
2. Mise à jour des packages de ubuntu

```
Sudo apt-get update
```

```
root@abir-VirtualBox:/home/abir# sudo apt update
Atteint :1 http://tn.archive.ubuntu.com/ubuntu focal InRelease
Réception de :2 http://security.ubuntu.com/ubuntu focal-security InRelease [128
kB]
Réception de :3 http://tn.archive.ubuntu.com/ubuntu focal-updates InRelease [12
8 kB]
Réception de :4 https://apt.releases.hashicorp.com focal InRelease [12.9 kB]
Réception de :5 http://tn.archive.ubuntu.com/ubuntu focal-backports InRelease [
```

## Etape 2. Mise en place de l'environnement MLFlow

1. Installation de python3

```
sudo apt-get install python3-pip
```

```
root@abir-VirtualBox:/home/abir# sudo apt install python3-pip
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Les paquets suivants ont été installés automatiquement et ne sont plus nécessai
res :
  libaio1 libcgi-fast-perl libcgi-pm-perl libevent-core-2.1-7
```

2. Mise à jour de pip3

```
sudo pip3 install --upgrade pip
```

```
root@abir-VirtualBox:/home/abir# pip3 install --upgrade pip
Collecting pip
  Downloading pip-24.3.1-py3-none-any.whl (1.8 MB)
    | 1.8 MB 861 kB/s
Installing collected packages: pip
  Attempting uninstall: pip
```

3. Installation de mlflow et scikit-learn

```
sudo pip3 install mlflow scikit-learn
```

```
root@abir-VirtualBox:/home/abir# pip3 install mlflow scikit-learn
Collecting mlflow
  Downloading mlflow-2.17.2-py3-none-any.whl (26.7 MB)
    | 26.7 MB 1.7 MB/s
Collecting scikit-learn
  Downloading scikit_learn-1.3.2-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_
x86_64.whl (11.1 MB)
    | 11.1 MB 807 kB/s
Collecting matplotlib<4
  Downloading matplotlib-3.7.5-cp38-cp38-manylinux_2_12_x86_64.manylinux2010_x8
6_64.whl (9.2 MB)
    | 9.2 MB 1.4 MB/s
```

### Etape 3. Exemple de MLFlow

1. Créez un fichier Python `mlflow_classification_example.py`:

```
import mlflow
import mlflow.sklearn
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, accuracy_score
# Charger le jeu de données Iris
iris = load_iris()
X = iris.data
y = iris.target

# Diviser le jeu de données en ensembles d'entraînement et de test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Commencer une nouvelle expérience avec MLflow
with mlflow.start_run():

    # Définir les hyperparamètres du modèle
    n_estimators = 100 # Nombre d'arbres dans la forêt
    max_depth = 5 # Profondeur maximale des arbres

    # Initialiser le modèle RandomForestClassifier
    model = RandomForestClassifier(n_estimators=n_estimators, max_depth=max_depth,
random_state=42)

    # Entraîner le modèle
    model.fit(X_train, y_train)

    # Faire des prédictions
    y_pred = model.predict(X_test)

    # Calculer l'accuracy
    accuracy = accuracy_score(y_test, y_pred)

    # Log des paramètres, des métriques et du modèle
    mlflow.log_param("n_estimators", n_estimators)
    mlflow.log_param("max_depth", max_depth)
    mlflow.log_metric("accuracy", accuracy)

    # Log du modèle
    mlflow.sklearn.log_model(model, "random_forest_model")

    print(f"Accuracy: {accuracy}")
    print("Classification Report:")
    print(classification_report(y_test, y_pred))
```

Nous allons entraîner un modèle de classification de type "Forêt Aléatoire" (Random Forest) pour prédire l'espèce d'une fleur à partir des caractéristiques de l'iris. Nous utiliserons le jeu de données Iris, qui est un jeu de données bien connu pour les tâches de classification, et nous suivrons les performances du modèle avec MLflow.

2. Exécutez votre script Python pour entraîner le modèle et enregistrer les résultats dans MLflow :

```
python3 mlflow_classification_exemple.py
```

```
root@abir-VirtualBox:/home/abir# python3 mlflow_classification_exemple.py
2024/11/26 00:27:58 WARNING mlflow.models.model: Model logged without a signature and input example. Please set 'input_example' parameter when logging the model to auto infer the model signature.
Accuracy: 1.0
Classification Report:
              precision    recall  f1-score   support

     0           1.00        1.00        1.00        19
     1           1.00        1.00        1.00        13
     2           1.00        1.00        1.00        13

 accuracy          1.00
 macro avg          1.00
 weighted avg       1.00
```

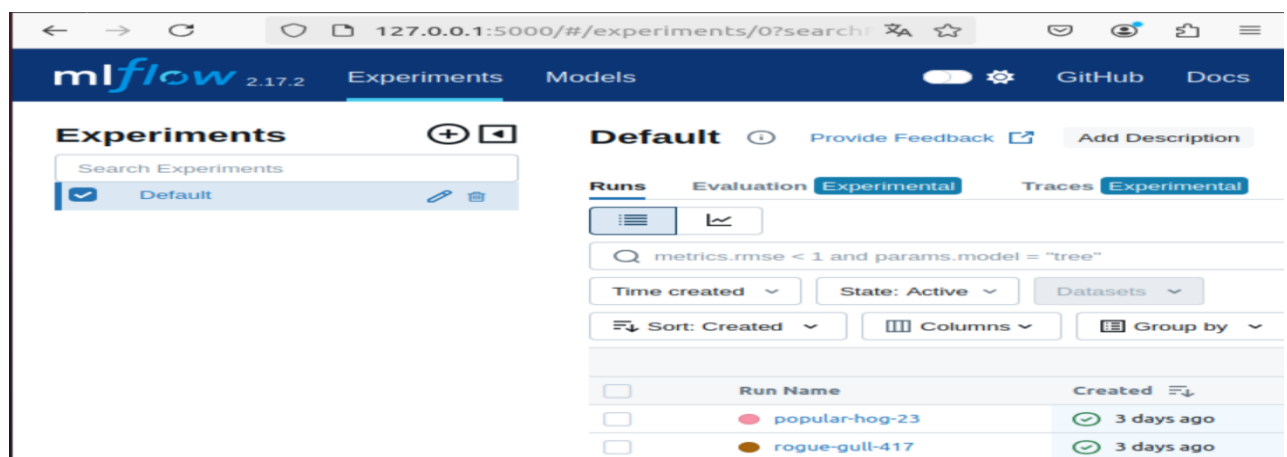
3. Visualiser les résultats avec l'interface utilisateur de MLflow

Après avoir exécuté le script, vous pouvez démarrer l'interface web de MLflow pour visualiser vos expériences :

```
mlflow ui
```

L'interface sera accessible à l'adresse suivante dans votre navigateur :

<http://127.0.0.1:5000>



L'interface utilisateur vous permettra de visualiser les différents runs, d'examiner les hyperparamètres, les métriques (telles que l'accuracy), ainsi que de comparer plusieurs exécutions.

#### 4. Charger et utiliser le modèle enregistré avec MLflow

Une fois que le modèle est enregistré, vous pouvez le charger et l'utiliser pour des prédictions ultérieures. Par exemple, pour charger le modèle sauvegardé et l'utiliser pour faire des prédictions sur de nouvelles données, voici un exemple de code :

```
import mlflow
import mlflow.sklearn

# Charger le modèle depuis le run enregistré
model_uri = "runs:/<run_id>/random_forest_model"
model = mlflow.sklearn.load_model(model_uri)

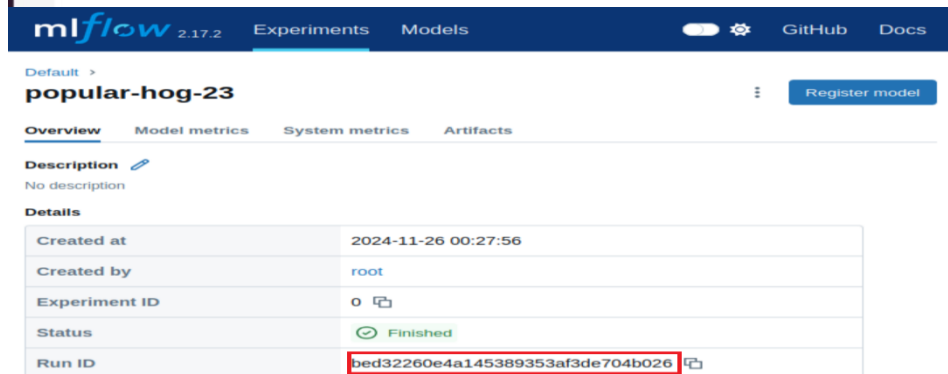
# Utiliser le modèle pour faire des prédictions sur de nouvelles données
new_data = [[5.1, 3.5, 1.4, 0.2]] # Exemple d'échantillon
prediction = model.predict(new_data)

print(f"Prediction: {prediction}")
```

**Remarque :** Remplacez <run\_id> par l'ID du run que vous pouvez trouver dans l'interface MLflow.

Dans notre cas :

```
1 import mlflow
2 import mlflow.sklearn
3
4 # Charger le modèle depuis le run enregistré
5 model_uri = "runs:/bed32260e4a145389353af3de704b026/random_forest_model"
6 model = mlflow.sklearn.load_model(model_uri)
7
8 # Utiliser le modèle pour faire des prédictions sur de nouvelles données
9 new_data = [[5.1, 3.5, 1.4, 0.2]] # Exemple d'échantillon
10 prediction = model.predict(new_data)
11
12 print(f"Prediction: {prediction}")
13
```



Details	
Created at	2024-11-26 00:27:56
Created by	root
Experiment ID	0
Status	Finished
Run ID	bed32260e4a145389353af3de704b026

Résultat de l'exécution du script de prédiction :

```
root@abir-VirtualBox: /home/abir# python3 mlflow_prediction.py
Prediction: [0]
```