

# Dagshub: Platform with GitHub experience for Data scientists

Machine Learning Architects Basel

Bassem Ben Hamed

December 2022





# Plan

- Introduction à Dagshub
- Motivation et avantages
- Intégration de MLflow avec Dagshub



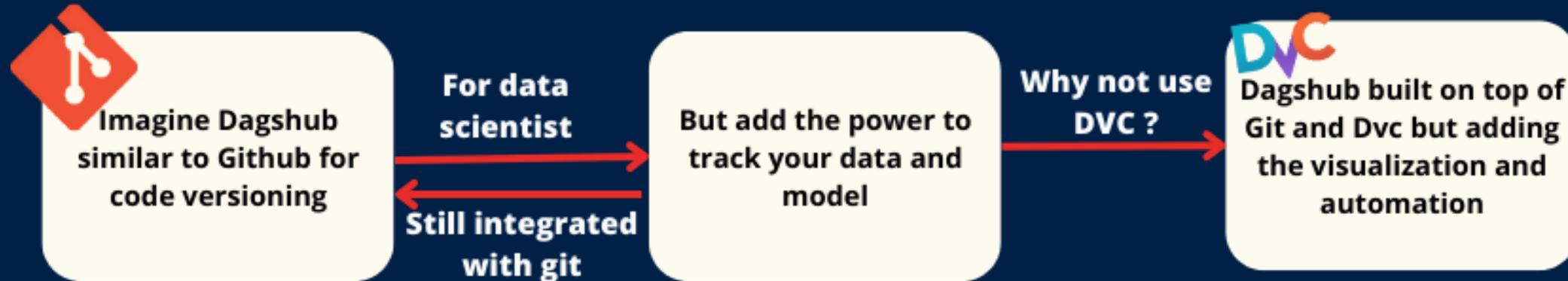
# Introduction à Dagshub

## Plateforme Open source

- Couvre l'ensemble du cycle de vie ML
- Versionne facilement vos ensembles de données et vos modèles (git et DVC).
- Suivre les expériences, étiquetage les données et visualisation, comparaison et partage des résultats.
- Vu global pour votre projet

# Introduction to Dagshub

## What is DagsHub ?

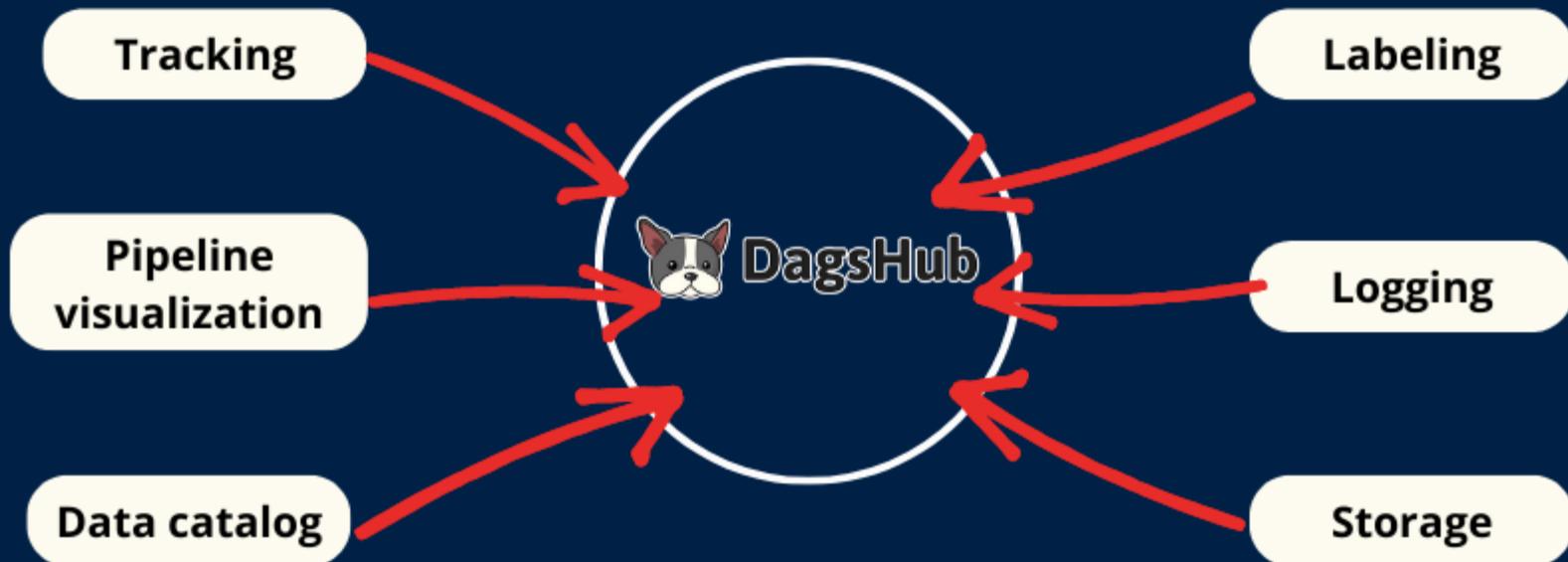


# Motivation et avantages

- La construction d'un modèle d'apprentissage automatique est un processus long, qui nécessite souvent des centaines d'itérations (expériences).
- Dagshub conçoit de meilleures expériences, grâce aux enseignements tirés des expériences précédentes.
- Dagshub améliore la reproductibilité en gardant la trace de la configuration qui avait des performances idéales.

# Motivation et avantages

## Advantages



# Motivation et avantages

## Integration simple

The screenshot shows a software interface with a top navigation bar featuring 'New file', 'Upload file', and a 'Remote' dropdown. Below this, there are three tabs: 'Git', 'DVC', and 'MLflow'. The 'Git' tab is selected, showing a 'Git remote:' section with a URL: [https://dagshub.com/dip.sdn/DIP\\_Dagshub.git](https://dagshub.com/dip.sdn/DIP_Dagshub.git). Below this, another 'Remote' dropdown is visible. The 'MLflow' tab is also present.

The screenshot shows a software interface with a top navigation bar featuring 'New file', 'Upload file', and a 'Remote' dropdown. Below this, there are three tabs: 'Git', 'DVC', and 'MLflow'. The 'DVC' tab is selected, showing a 'DVC remote:' section with a URL: [https://dagshub.com/dip.sdn/DIP\\_Dagshub.dvc](https://dagshub.com/dip.sdn/DIP_Dagshub.dvc). Below this, there is a section titled 'Add a DagsHub DVC remote:' containing the command: `dvc remote add origin https://dagshub.com/dip.sdn/DIP_Dagshub.dvc`. At the bottom, there is a 'Setup credentials:' section with commands: `dvc remote modify origin --local auth basic`, `dvc remote modify origin --local user dip.sdn`, and `dvc remote modify origin --local password your_token`.

# Integration MLflow avec Dagshub

```
import os  
os.environ['MLFLOW_TRACKING_USERNAME'] = "dip.sdn"  
os.environ["MLFLOW_TRACKING_PASSWORD"] = "████████████████████████████████████████"  
#setup mlflow  
mlflow.set_tracking_uri('https://dagshub.com/dip.sdn/DIP_Dagshub.mlflow') #your mlflow tracking uri
```

Setup dagshub  
environement

```
def train_model(alpha,l1_ratio):  
    with mlflow.start_run():  
        lr = ElasticNet(alpha=alpha, l1_ratio=l1_ratio, random_state=42)  
        lr.fit(train_x, train_y)  
        predicted_qualities = lr.predict(test_x)  
        (rmse, mae, r2) = eval_metrics(test_y, predicted_qualities)  
        print("Elasticnet model (alpha=%f, l1_ratio=%f):" % (alpha, l1_ratio))  
        print(" RMSE: %s" % rmse)  
        print(" MAE: %s" % mae)  
        print(" R2: %s" % r2)  
        mlflow.log_param("alpha", alpha)  
        mlflow.log_param("l1_ratio", l1_ratio)  
        mlflow.log_metric("rmse", rmse)  
        mlflow.log_metric("r2", r2)  
        mlflow.log_metric("mae", mae)  
        mlflow.sklearn.log_model(lr, "model")
```



how to log params  
metrics and model

# Integration MLflow avec Dagshub

The screenshot shows the Dagshub interface with the following components:

- Run name:** A blue box at the top left containing the text "Run name". A red arrow points from this box to the "Name" column in the experiment table.
- Source:** A blue box at the top center containing the text "Source". A blue arrow points from this box to the "Source" column in the experiment table.
- Params:** A blue box at the top right containing the text "Params". A blue arrow points from this box to the "l1\_ratio" and "alpha" columns in the experiment table.
- Metrics:** A blue box at the far right containing the text "Metrics". Three pink arrows point from this box to the "mae", "r2", and "rmse" columns in the experiment table.

The interface includes a navigation bar with links for Files, Experiments (5), Issues (0), Pull Requests (0), Reports (new), Discussions (0), Annotations, and Settings. Below the navigation bar is a toolbar with buttons for Compare, Reset filters, Archive, Labels (highlighted with a pink box), and Columns.

	Code	Name	Cre...	Labels	Sou...	Group	l1_ratio	alpha	mae	r2	rmse
<input type="checkbox"/>		cloud anteater	26 minutes ago		mlflow	Default	0.4	0.1	0.5570633397...	0.2112212129...	0.7020423742...
<input type="checkbox"/>		tree eagle	26 minutes ago		mlflow	Default	0.1	0.4	0.5649807351...	0.2071461309...	0.7038535250...
<input type="checkbox"/>		cherry starfish	26 minutes ago		mlflow	Default	0.1	0.1	0.5418462809...	0.2354407678...	0.6911801919...
<input type="checkbox"/>		meadow canidae	27 minutes ago		mlflow	Default	0.2	0.2	0.5597774636...	0.2117652969...	0.7018002049...
<input type="checkbox"/>		sun jellyfish	27 minutes ago		mlflow	Default	0.5	0.5	0.6040022590...	0.1415407078...	0.7323953326...

# Integration MLflow avec Dagshub

## MLflow Ui

The screenshot shows the MLflow UI interface. On the left, there is a sidebar with tabs for Git, DVC, and MLflow. The MLflow tab is selected. It displays the 'MLflow Tracking remote:' section, which contains the URL [https://dagshub.com/dip.sdn/DIP\\_Dagshub.mlflow](https://dagshub.com/dip.sdn/DIP_Dagshub.mlflow). A red circle highlights the 'Go to mlflow UI' button next to the URL. A red arrow points from this button to the main MLflow UI window on the right. The main window shows the 'Experiments' tab with the 'Default' experiment. It displays 5 matching runs, each with details like start time, duration, run name, user, source, version, models, mae, and r2 metrics. The 'Metrics' table is as follows:

	↓ Start Time	Duration	Run Name	User	Source	Version	Models	mae	r2
<input type="checkbox"/>	49 minutes ago	26.1s	-	dip.sdn	C:\Users\...	-	sklearn	0.557	0.211
<input type="checkbox"/>	50 minutes ago	21.3s	-	dip.sdn	C:\Users\...	-	sklearn	0.565	0.207
<input type="checkbox"/>	50 minutes ago	19.6s	-	dip.sdn	C:\Users\...	-	sklearn	0.542	0.235
<input type="checkbox"/>	50 minutes ago	18.1s	-	dip.sdn	C:\Users\...	-	sklearn	0.56	0.212
<input type="checkbox"/>	51 minutes ago	23.7s	-	dip.sdn	C:\Users\...	-	sklearn	0.604	0.142

# Integration MLflow avec Dagshub

## MLflow Ui(Artifats)

The screenshot illustrates the integration of MLflow with Dagshub. On the left, the MLflow UI displays a list of experiments. A red circle highlights the first run, which has a duration of 26.1s and was started 49 minutes ago. A red arrow points from this run to the detailed view on the right. The right panel shows the details for the run 'Run 2b5bb5b1123a428d8e70289c9782e22c'. It includes the run's date (2022-09-20 10:49:17), source (C:\Users\User\anaconda3\lib\site-packages\ipykernel\_launcher.py), user (dip.sdn), status (FINISHED), and lifecycle stage (active). The 'Artifacts' section is expanded, showing a tree structure for the 'model' directory. The 'MLmodel' file is selected, revealing its contents:

```
Full Path: mlflow-artifacts:/9a54350726e24fff93dd9022bfdcd3a/2b5bb5b1123a428d8e70289c9782e22c/artifacts/MLmodel
Size: 448B
artifact_path: model
flavors:
python_function:
env: conda.yaml
loader_module: mlflow.sklearn
model_path: model.pkl
python_version: 3.9.7
sklearn:
code: null
pickled_model: model.pkl
serialization_format: cloudpickle
sklearn_version: 0.24.2
mlflow_version: 1.27.0
model_uuid: 6838a4f0915944918668d9c9c82e46cb
run_id: 2b5bb5b1123a428d8e70289c9782e22c
utc_time_created: '2022-09-20 10:49:18.639499'
```