MLFLOW

MLflow is an open source platform for managing the end-to-end machine learning lifecycle. It tackles four primary functions:

* Tracking experiments to record and compare parameters and results ([MLflow Tracking](https://mlflow.org/docs/latest/tracking.html" \l "tracking)).
* Packaging ML code in a reusable, reproducible form in order to share with other data scientists or transfer to production ([MLflow Projects](https://mlflow.org/docs/latest/projects.html" \l "projects)).
* Managing and deploying models from a variety of ML libraries to a variety of model serving and inference platforms ([MLflow Models](https://mlflow.org/docs/latest/models.html" \l "models)).
* Providing a central model store to collaboratively manage the full lifecycle of an MLflow Model, including model versioning, stage transitions, and annotations ([MLflow Model Registry](https://mlflow.org/docs/latest/model-registry.html" \l "registry)).
* INSTALLATION:

pip install mlflow

* REPOSITORY:

git clone https://github.com/mlflow/mlflow

* TRACKING API

Logs the the metrics and artifacts(files).

* TRACKING UI

mlflow ui --host IP

To view the performance metrics visually.

MLFLOW Part 1(Tracking Module):

Simple Steps to convert a ML model to MLFLow.

* 1. Import required MLFlow packages.
  2. Use MLFlow Automatic Logging to log every metric and parameters of the model.
  3. Initiate the training model with mlflow.start\_run to start logging.

Part 2(MLFlow Projects):

1. A MLFlow Project is a format for packaging ML models code in a reusable and reproductible way.

Part 3(MLFlow Models):

1. Deploy ML model in enviroenments such as Azure Databricks, databricks community edition.
2. Deploy locally

mlflow model serve –model-uri runs://model\_id/model --port 1234

Part 4(MLFlow Model registry):

1. Train a Model and register the best iteration.
2. Load and evaluate the current production model.
3. Transition the model to staging.
4. Compare both staging and production model.
5. Transition the model from staging to production.

**#Register the model for Staging & for production**

mlflow ui --backend-store-uri sqlite:///mlruns.db --host localhost

**#Set enviorment variable for the tracking URL where the Model Registry is**

export MLFLOW\_TRACKING\_URI=sqlite:///mlruns.db

**#Serve the production model from the model registry**

mlflow models serve --model-uri models:/101/production --host localhost--no-conda

**#To test query images**

python score\_images\_rest.py --host <http://localhost> flower\_photos/query/