

Experiment tracking

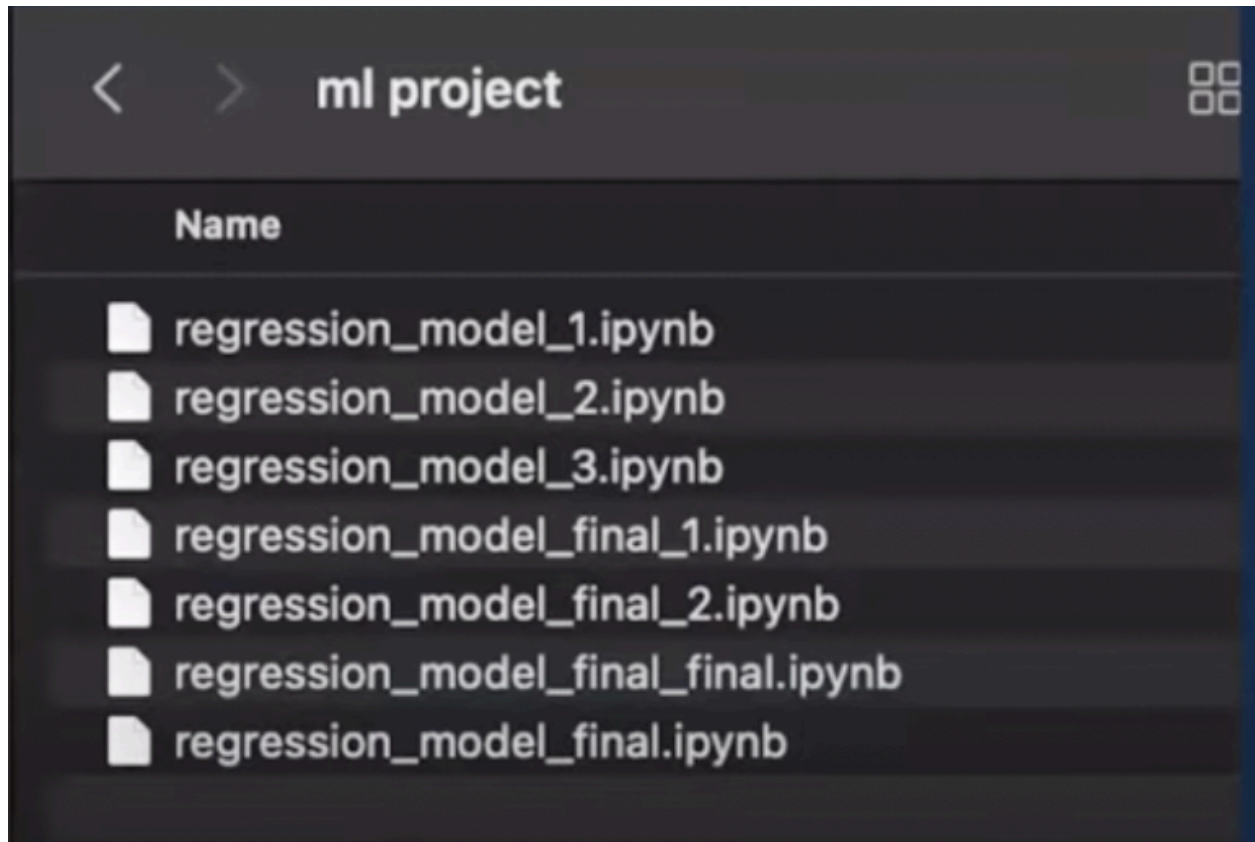
You can think of experiments as the process of building an ML model. When we say experiment run, we mean each trial in an ML experiment. So the ML experiment is actually the whole process that a data scientist may start playing with some data, models and hyperparameters. Each of these trials is an experiment run.

Experiment tracking is the process of keeping track of all the relevant information from ML experiments.

- **Organize** all the necessary components of a specific experiment. It's important to have everything in one place and know where it is so you can use them later.
- **Reproduce** past results (easily) using saved experiments.
- **Log** iterative improvements across time, data, ideas, teams, etc.

We have created a model successfully, but now we have new set of data how do you proceed on working with it

- we can change the data and run the code again
 - but we will lose the output and results from the old data
- we can create new cells below these to create a new model with the new data
 - but then when we have a lot of experiments in one file it will be really difficult finding the one we want to look at
- We can create new files for each experiment
 - but for actually comparing the results and outputs you'll still have to open each file and look into it closely



These are not the best ways of keeping track of the work and experiments that you perform, we need to create something that is easy to manage, clearly shows the results and metrics, logs the changes and hyperparameters for us.

ML Flow

<https://mlflow.org/>

MLflow is an open-source platform to manage Machine Learning Lifecycle. In layman's terms, it can track and store data, parameters, and metrics to be retrieved later or displayed nicely on a web interface. Furthermore, MLflow is a framework-agnostic tool, so any ML / DL framework can quickly adapt to the ecosystem that MLflow proposes.

MLflow emerges as a platform that offers tools for tracking metrics, artifacts, and metadata.

ML flow Tracking

MLflow Tracking is an API-based tool for logging metrics, parameters, model versions, code versions, and files. MLflow Tracking is integrated with a UI for visualizing and managing artifacts, models, files, etc.

Each MLflow Tracking session is organized and managed under the concept of runs.

- A run refers to the execution of code where the artifact log is performed explicitly.
- An MLflow experiment is the primary unit of organization and access control for MLflow runs; all MLflow runs belong to an experiment. Experiment:{run,run.....run}
- Experiments let you visualize, search for, and compare runs, as well as download run artifacts and metadata for analysis in other tools.
- An MLflow run corresponds to a single execution of model code. Each run records the some information about that particular trial

MLflow categorizes these into:

- **Parameters** (via `mlflow.log_param()`). Parameters are variables that you change or tweak when tuning your model.
- **Metrics** (using `mlflow.log_metric()`). Metrics are values that you want to measure as a result of tweaking your parameters. Typical metrics that are tracked can be items like F1 score, RMSE, MAE etc.
- **Artifacts** (using `mlflow.log_artifact()`). Artifacts are any other items that you wish to store. Typical artifacts to keep track of are PNGs of graphs, plots, confusion matrix, and also pickled model files
- **Params** are something you want to tune based on the metrics, whereas tags are some extra information that doesn't necessarily associate with the model's performance. there's no hard constraint on which to use to log which; they can be used interchangeably without error.