Setup:

1. I started with installing MLflow framework on my local machine. MLflow is used to visualize various experiments.

2. After installing mlflow I Installed a MySQL plugin for the MLflow setup for storing the data in MySQL.

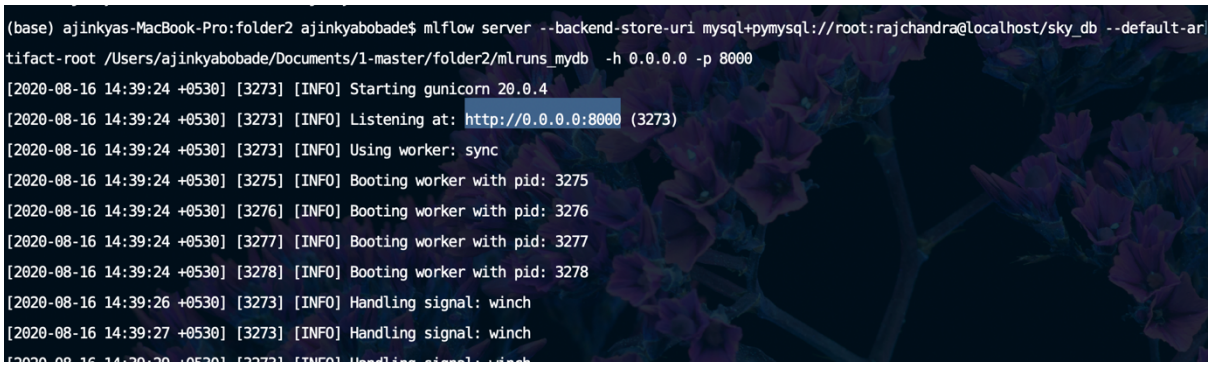
3. To start MLflow server run the following code in your conda (base) environment:

```
mlflow server --backend-store-uri
mysql+pymysql://<db_username>:<db_password>@localhost/<db_name> --default-artifact-root
<path_of_artifact_folder>-h 0.0.0.0 -p 8000
```

What this code will do is start a MLflow server and store all the metadata on a MySQL database named <db_name>. All the artifacts (models, PNG, csv, etc) files will be stored on <path_of_artifact_folder>. Note that artifacts can also be stored on cloud storage. In my case I use localhost to store artifacts.

MLflow backend + MySQL :

Start MLflow server and route metadata to MySql backend as follows:

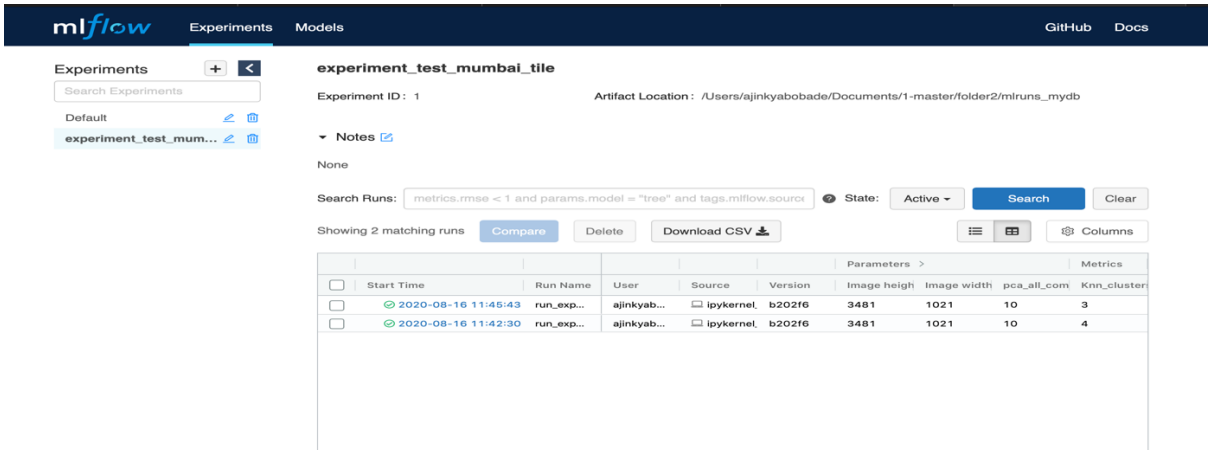


After starting backend I ran my python script, which has mlflow integration, Script location:

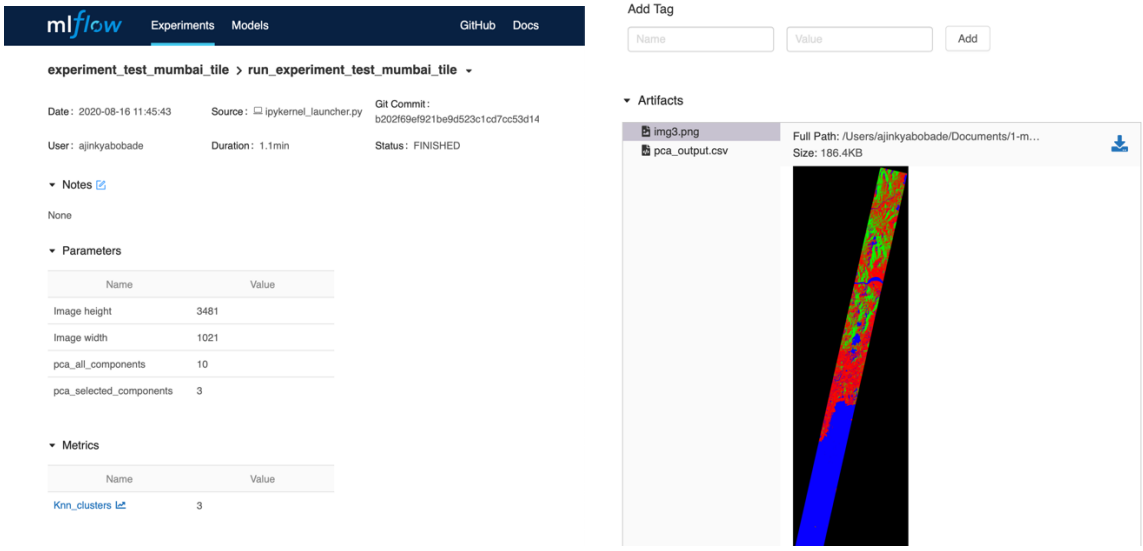
https://github.com/ajinkya933/Assignment_solution/blob/master/mlflow_integration/mlflow_assignment.ipynb

MLflow frontend:

Go to : http://0.0.0.0:8000 on your localhost after you start the MLflow server. Then This we see this output:



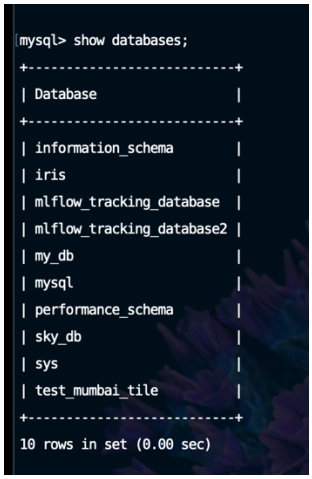
The above UI tracks multiple experiments. In each experiment there are multiple runs tracked. Now let's go to experiment_test_mumbai_tile > run_ experiment_test_mumbai_tile



In this experiment I have checked output of Mumbai tile where (Knn clusters = 3) Similarly there are other experiments where (Knn clusters = 2,4, ..etc)

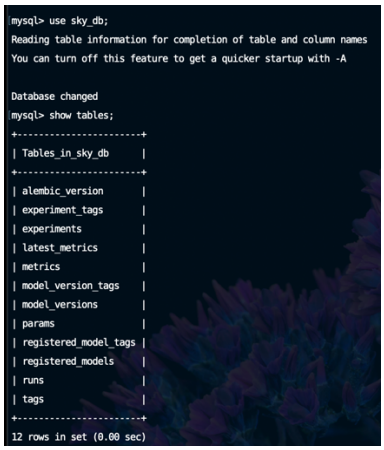
MySql backend:

All the metadata is stored on MySql backend. To login and get the data we use:



```
mysql> show databases;
+--------------------------+
| Database                 |
+--------------------------+
| information_schema       |
| iris                     |
| mlflow_tracking_database |
| mlflow_tracking_database2 |
| my_db                    |
| mysql                    |
| performance_schema       |
| sky_db                   |
| sys                      |
| test_mumbai_tile         |
+--------------------------+
10 rows in set (0.00 sec)
```

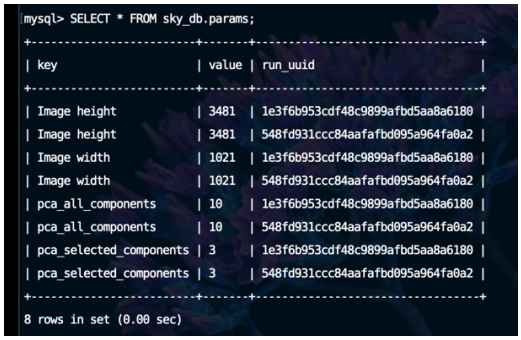
Our database name is sky_db. Now, list tables In sky databse:



```
mysql> use sky_db;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+---------------------+
| Tables_in_sky_db    |
+---------------------+
| alembic_version     |
| experiment_tags     |
| experiments         |
| latest_metrics      |
| metrics             |
| model_version_tags  |
| model_versions      |
| params              |
| registered_model_tags |
| registered_models   |
| runs                |
| tags                |
+---------------------+
12 rows in set (0.00 sec)
```

All the parameters on front end are saved in out params table of sky_db:



```
mysql> SELECT * FROM sky_db.params;
+------------------------+-------+----------------------------------+
| key                    | value | run_uuid                         |
+------------------------+-------+----------------------------------+
| Image height           | 3481  | 1e3f6b953cdf48c9899afbd5aa8a6180 |
| Image height           | 3481  | 548fd931ccc84aafafbd095a964fa0a2 |
| Image width            | 1021  | 1e3f6b953cdf48c9899afbd5aa8a6180 |
| Image width            | 1021  | 548fd931ccc84aafafbd095a964fa0a2 |
| pca_all_components     | 10    | 1e3f6b953cdf48c9899afbd5aa8a6180 |
| pca_all_components     | 10    | 548fd931ccc84aafafbd095a964fa0a2 |
| pca_selected_components | 3     | 1e3f6b953cdf48c9899afbd5aa8a6180 |
| pca_selected_components | 3     | 548fd931ccc84aafafbd095a964fa0a2 |
+------------------------+-------+----------------------------------+
8 rows in set (0.00 sec)
```

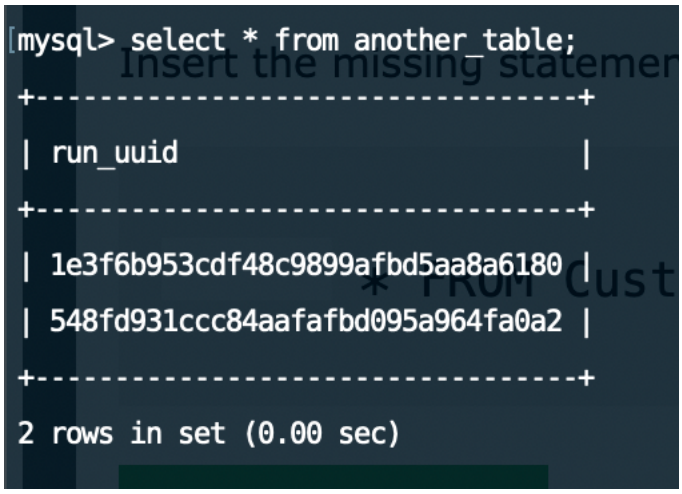
Querying tables:

1. Query using experiment name, only those tables which have ran successfully on MLflow and get their UUID

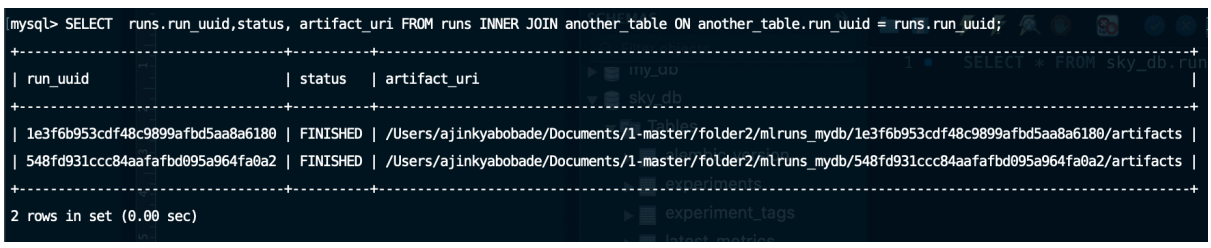
2. Create another table with above entries using:

```
CREATE TABLE another_table SELECT run_uuid FROM tags WHERE value =
'run_experiment_test_mumbai_tile' UNION SELECT run_uuid FROM runs
WHERE status = 'FINISHED'
```

Once done you can view this output using:

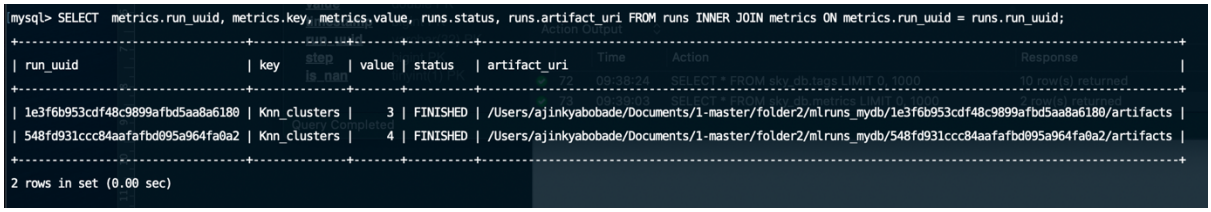


Using this UUID, look this UUID in another table and start constructing tables



Now join metrics to above table using:

```
SELECT  metrics.run_uuid, metrics.key, metrics.value, runs.status,
runs.artifact_uri FROM runs INNER JOIN metrics ON metrics.run_uuid =
runs.run_uuid;
```



```
mysql> SELECT  metrics.run_uuid, metrics.key, metrics.value, runs.status, runs.artifact_uri FROM runs INNER JOIN metrics ON metrics.run_uuid = runs.run_uuid;
+----------------------------------+-------------+-------+----------+----------------------------------------------------------------------------------------------------------------------------+
| run_uuid                         | key         | value | status   | artifact_uri                                                                                                               |
+----------------------------------+-------------+-------+----------+----------------------------------------------------------------------------------------------------------------------------+
| 1e3f6b953cdf48c9899afbd5aa8a6180 | Knn_clusters |     3 | FINISHED | /Users/ajinkyabobade/Documents/1-master/folder2/mlruns_mydb/1e3f6b953cdf48c9899afbd5aa8a6180/artifacts |
| 548fd931ccc84aafafbd095a964fa0a2 | Knn_clusters |     4 | FINISHED | /Users/ajinkyabobade/Documents/1-master/folder2/mlruns_mydb/548fd931ccc84aafafbd095a964fa0a2/artifacts |
+----------------------------------+-------------+-------+----------+----------------------------------------------------------------------------------------------------------------------------+
2 rows in set (0.00 sec)
```

Now depending on value you can take run_uuid and search for its artfact uri or param values.