MLFlow မှာ

1. MLflow Tracking
2. MLflow Projects
3. MLflow Models
4. MLflow Registry ဆိုပြီးရှိ။ တပိုင်းစီဟာ လွတ်လပ်စွာ အသုံးပြုနိုင်တယ်။ ပေါင်းစပ်အသုံးပြုပါက ပိုမိုကောင်းမွန်။

**MLflow Tracking**

Tracking ရဲ့ အဓိက ရည်ရွယ်ချက်ကတော့ code တခါ run တိုင်း အောက်ပါ အချက်အလက်တွေကို မှတ်တမ်းတင်ထားဖို့ပါ။

1. Code version,
2. Start and End Time
3. Source (the run)
4. Parameters
5. Metrics
6. Artifacts

log\_metric, log\_para, log\_artifacts သုံးပြီး tracking လုပ်ထားနိုင်ပါတယ်။ Tracking အတွက် tracking server တခုခုကို configure မလုပ်သေးရင် "mlruns" ဆိုတဲ့ local folder မှာ သိမ်းထားပါတယ်။

Tracking လုပ်ထားသမျှ ပြန်ကြည့်ချင်ရင် pip install mlflow နဲ့သွင်းပါ။ ပြီးရင် mlflow ui ခေါ်ပြီး ပေါ်လာတဲ့ localhost link မှာကြည့်နိုင်ပါတယ်။

MLflow က Code တွေထည့်ပြီးပါက python [python file path] နဲ့ run လိုက်ရင် (Train တဲ့သဘော) mlruns folder အောက်မှာ experiment ID နဲ့ folder တစ်ခု ပေါ်လာမယ်။ အထဲမှာ Artifacts folder ရှိမယ်။ Artifact folder ထဲမှာ MLproject, model.pkl, conda.yaml, requirements.txt ဆိုတဲ့ 4 ဖိုင်တွေ့မယ်။

MLproject ဆိုတာကတော့ MLmodel ရဲ့ Description file ပါ

model.pkl ဆိုတာကတော့ ကိုယ့် model ရဲ့ serialized model ပါ

conda.yaml ဆိုတာကတော့ conda dependencies တွေကို ပြထားပါတယ်။

requirements.txt ကတော့ libraries requirement တွေပေါ့။

**Remote Tracking Server**

ကိုယ့်စက်မှာ track မလုပ်ခိုင်းဘဲ team တစ်ခုအတွက် ဗဟိုပြုထားချင်Remote Tracking Server ကို logging လုပ်မယ်ဆိုရင် LFLOW\_TRACKING\_URI environment variable to your server’s URI ဒါမှမဟုတ် ကိုယ့် program ရဲ့အစမှာ အောက်ပါ code ကိုထည့်ပါ

import mlflow

mlflow.set\_tracking\_uri("http://YOUR-SERVER:4040")

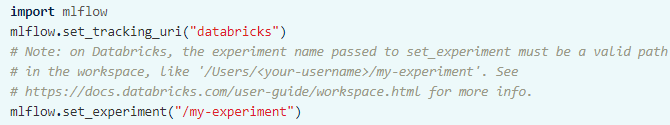
mlflow.set\_experiment("my-experiment")

**နောက်တနည်းကတော့**

Databrick community ကို signup လုပ်ပြီးတော့လည်း remote server ကိုသုံးနိုင်ပါတယ်။

ဒါပေမယ့် သူက experimental ပါ။ လုပ်မယ်ဆိုရင်

*set the MLFLOW\_TRACKING\_URI environment variable to “databricks”, or add the following to the start of your program:*



**mlflow ui**

mlflow ui နှင့်ပက်သက်ပြီး ရှိနေတဲ့ directory မှာရှိတဲ့ mlruns ထဲက အရာတွေကိုပဲ ပြသမှာပါ။

**MLflow Projects**

**MLflow Project** ကို runတဲ့အခါ mlflow run နဲ့ run ရပါတယ်။ ဒီအတွက် dependencies တွေ environment တွေ ဂရုစိုက်စရာမလိုပါဘူး။ MLproject ဆိုတဲ့ ဖိုင်မှာ လိုတဲ့ dependencies တွေဖော်ပြထားပါတယ်။ MLProject file ကတော့ mandatory ပါ။ conda.yaml ကြောင့် conda env မှာ run ပါလိမ့်မယ်။

MLflow Project ကို run တဲ့အခါ local directory က project ကိုrun နိုင်လို Github URI မှာတင်ထားတဲ့ project ကိုလည်းနဲ့တခါတည်း run လို့ရပါတယ်။

Default အနေနဲ့ mlflow ကို run တဲ့အခါ Conda ကိုသုံးပြီး dependencies တွေကို install လုပ်ပါတယ်။ Conda မသုံးဘဲ run ချင်ရင် --no-conda လို့ flag တခု ထည့်ထောင်ပေးရမှာဖြစ်ပါတယ်။ အဲလိုဆိုရင် ကိုယ့်ရဲ့ python environment ထဲမှာ လိုတဲ့ dependencies တွေထည့်ထားဖို့လိုပါတယ်။

mlflow run mlflow/examples/sklearn\_elasticnet\_wine -P alpha=0.43

ဒီလို command ကို run လိုက်မယ်ဆိုရင် new Conda env မှာ run လိမ့်မယ်။ Repository မှာသာ MLProject ရှိမယ်ဆိုရင် တခါတည်း run လို့ရပါတယ်။ဥပမာ repo မှာ MLProject File ရှိရင် github ကိုလှမ်းrun နိုင်ပါတယ်။

**ဒီလို run တာက library တွေ သူ့ဘာသာ ရှာသွင်းလိမ့်မယ်။ dependencies တွေကိုပေါ့။ --no-conda ဆိုတဲ့ flag ထောင်ထားမယ်ဆိုရင်တော့ library dependencies တွေဟာ မိမိဘာသာသွင်းထားရမှာဖြစ်ပါတယ်။**

mlflow run https://github.com/mlflow/mlflow-example.git -P alpha=5.0 **--no-conda**

MLflow.\*.log\_model လုပ်တဲ့အခါ Pip Requirement တွေကို သတ်မှတ်နိုင်ပါတယ်။

[Ref: https://www.mlflow.org/docs/latest/tutorials-and-examples/tutorial.html#packaging-training-code-in-a-conda-environment]

အခု MLProject Convention ကိုသုံးပြီး pack လုပ်ပြီးပါပြီ။ The best model လည်း ရွေးချယ်ပြီးပါပြီ။

အခု MLflow models ကို သုံးဖို့ အချိန်ရောက်ပါပြီ။ MLflow models ဆိုတာ standard format တခုပါ။ အဲဒီ format က real-time API (like REST API) တွေမှာပါ သုံးလို့ရမှာပါ။

တခု သတိထားမိပါလိမ့်မယ်။ Training ပြီးတဲ့အခါ MLflow ထဲက function တစ်ခုဟာ Model ကို Artifact တစ်ခုအနေနဲ့ သိမ်းခဲ့ပါတယ်။

mlflow.sklearn.log\_model(lr, "model")

အဲဒီ Artifact ကိုကြည့်ချင်ရင် mlflow ui မှာ တခုချင်းကို click နှိပ်ပြီး model မှာ expand လုပ်ကြည့်နိုင်ပါတယ်။

အဲဒီမှာ file နှစ်ခုပါပါတယ်။ MLmodel နဲ့ model.pkl ပါ။ MLmodel က metadata file ပါ။ model.pkl က model ကို serialized လုပ်ထားတာပါ။

**Serving the model**

MLmodel format က mlflow ရဲ့ စံformat တခုဖြစ်သလို အမျိုးမျိုးသော tools တွေက မတူကွဲပြားတဲ့ အရသာအမျိုးမျိုးနဲ့ model တွေကို save ပေးတဲ့ format ဖြစ်ပါတယ်။

ဥပမာ..model တော်တော်များများက python function အနေနဲ့ အလုပ်လုပ်နိုင်တဲ့ အတွက် ဘယ်model ကို ဘယ် python function အဖြစ်အသုံးချဖို့ကို MLmodel file က ကြေငြာပေးပါတယ်။

နမူနာအနေနဲ့...

python sklearn\_logistic\_regression/train.py

ဒီ scikit-learn နဲ့ ရေးထားတဲ့ model ကို run တဲ့ အခါ experiment ID တခု ထုတ်ပေးမယ်ဆိုတာ သိပြီးဖြစ်တယ်။

အဲဒီ ID နဲ့ folder အောက်မှာ MLmodel description file နဲ့ pickled scikit-learn model တစ်ခု ရှိလိမ့်မယ်ဆိုတာ မှန်းဆပြီးဖြစ်လိမ့်မယ်။

အဲဒီ ID နဲ့ path ကို various tools ဆီကို ပို့ပေးပြီး အသုံးချနိုင်ပါတယ်။

ဥပမာ... MLflow တွေမှာ python-based model တွေအတွက် simple REST server တစ်ခု default အနေနဲ့ ပါပြီးဖြစ်ပါတယ်။

ဒီတော့ ဒီလိုလေး pass လုပ်ပေးနိုင်ပါတယ်...။

mlflow models serve -m runs:/<RUN\_ID>/model

ဒါဆိုရင် model က ဆာဗာဆီရောက်သွားပါပြီ။ dependencies အတွက် ပူစရာမလိုတော့ပါဘူး။

Default အနေနဲ့ port က 5000 မှာပွင့်ပါတယ်။ တခြားport မှာဖွင့်ချင်ရင်

mlflow models serve -m runs:/<RUN\_ID>/model --port 1234

or

mlflow **models serve -m** [your\_model\_path]/artifacts/model -p 1234

ဟုရေးပါ။ အဲဒီလို ဆာဗာ စလိုက်ပြီးဆိုတာနဲ့ sample data တွေပို့ပြီး prediction တွေခန့်မှန်းနိုင်ပါပြီ။

ဒီနေရာမှာ 1234 က Port ကို ပြောတာပါ။

**သတိ**

**"The version of Python used to create the model must be the same as the one running mlflow models serve."**

Command Line မှာ အောက်ပါ ကုဒ်ကို ထည့်ပြီး Predict နိုင်ပါတယ်။ ထပ်မံလေ့လာရန် [**MLflow deployment tools documentation.**](https://www.mlflow.org/docs/latest/models.html#local-model-deployment)

curl -X POST -H "Content-Type:application/json; format=pandas-split" --data "{\"columns\":[\"alcohol\", \"chlorides\", \"citric acid\", \"density\", \"fixed acidity\", \"free sulfur dioxide\", \"pH\", \"residual sugar\", \"sulphates\", \"total sulfur dioxide\", \"volatile acidity\"],\"data\":[[12.8, 0.029, 0.48, 0.98, 6.2, 29, 3.33, 1.2, 0.39, 75, 0.66]]}" http://127.0.0.1:1234/invocations

အဖြေကို...

[5.120775719594933]

လို ခပ်ဆင်ဆပ်ထုတ်ပေးလိမ့်မယ်။

mlflow run အနေနဲ့ run ရင် mlrun က တတန်းတည်းမှာရှိတယ်

python file ကို ဝင်run ရင် python file နဲ့ တတန်းတည်းမှာ mlrun ရှိတယ် (Pycharm တွင်)

VS Code တွင်မှု အပြင် MLFlow folder **အကြီး**နဲ့တတန်းတည်း mlrun တွင် ရှိတယ်။

(သိပ်တော့ အံ့ဩစရာမဟုတ်။ directory ရှိတဲ့နေရာပေါ်မူတည်တာပါ။ vscode တွင် python file ရှိသည့်နေရာတွေင် mlrun ပြလိုပါက cd ဖြင့် python file ရှိသည့်နေရာသို့ အရင်သွားပါ)

mlfow ui နဲ့ကြည့်တဲ့အခါ အရင် run ပြီးသားရှိပါက ပိတ်ရန်လိုပါသည်။ နို့မို့ အသစ်ကိုမပြနိုင်ပါဘူး။

အထူးသဖြင့် mlflow ui ရှိနေတဲ့ mlrun folder directory မတူတဲ့အခါ အရင်ဖွင့်ထားတဲ့ mlflow ui ကိုပဲပြနေမှာပါ။