Name: Osama Abdul Razzak(2303.KHI.DEG.029)
Peer Name: Rahima Siddiqui(2303.KHI.DEG.030)

Assignment 3.4(a) and 3.4(b)

Write a component that will log metadata of your Classification model that you trained on the day dedicated to Supervised Learning. Remember to include all metadata that are important to track for this problem.

Run your Classification model that you trained on the day dedicated to Supervised Learning in MLFlow.

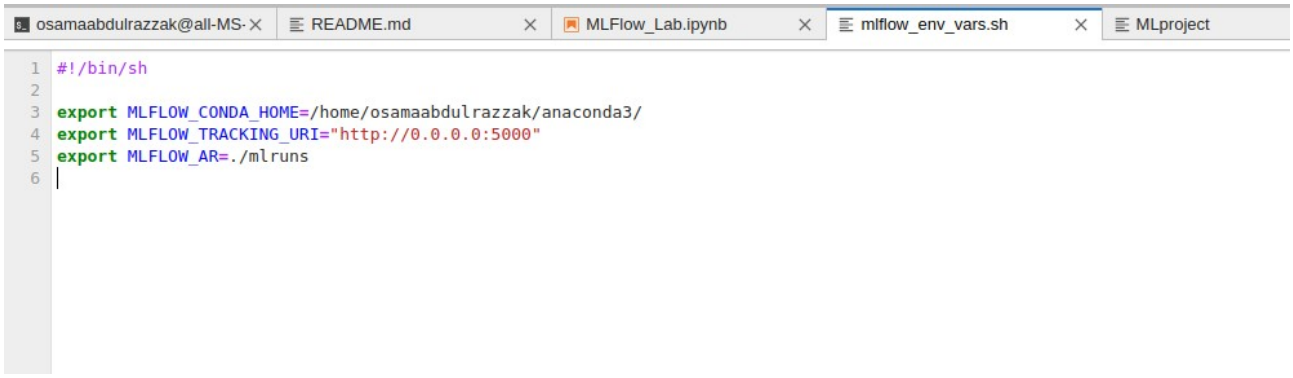
We work on Logistics Regression Classification model

First we created virtual environment and install required libraries init

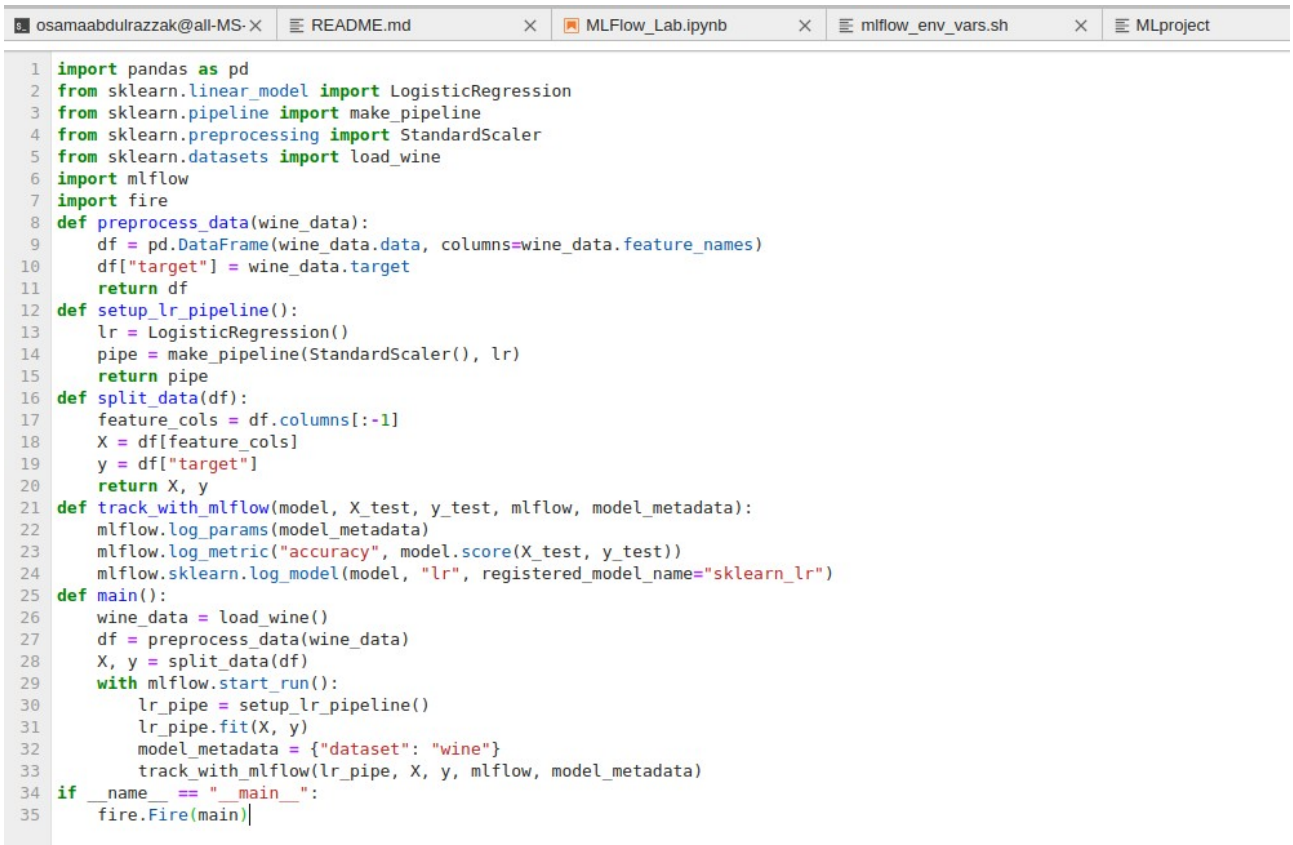
also install kernel in virtual environment and use that kernel for the task

and here are environment variable for the code

Name: Osama Abdul Razzak(2303.KHI.DEG.029)
Peer Name: Rahima Siddiqui(2303.KHI.DEG.030)



and python file for logistic regression

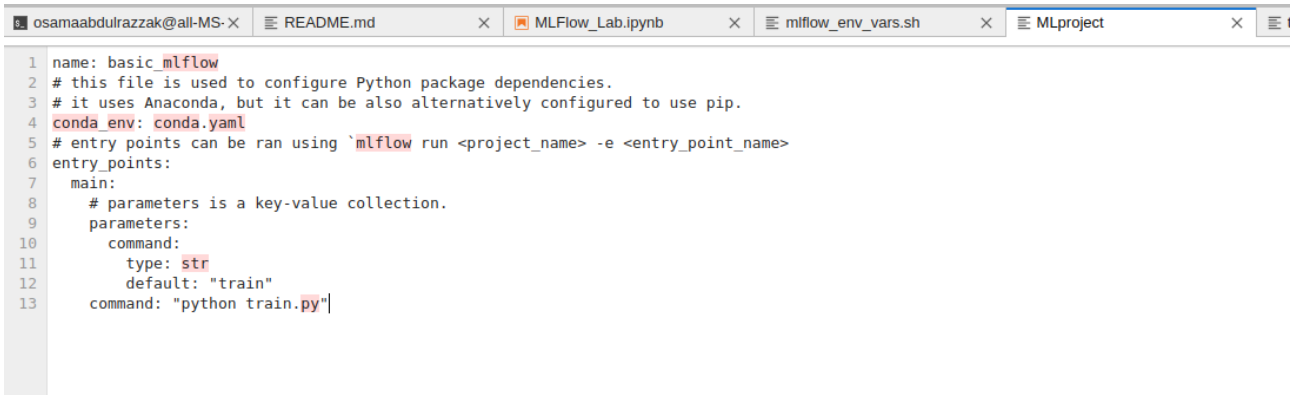


import pandas as pd
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.datasets import load_wine

Name: Osama Abdul Razzak(2303.KHI.DEG.029)
Peer Name: Rahima Siddiqui(2303.KHI.DEG.030)

```
import mlflow
import fire
def preprocess_data(wine_data):
    df = pd.DataFrame(wine_data.data, columns=wine_data.feature_names)
    df["target"] = wine_data.target
    return df
def setup_lr_pipeline():
    lr = LogisticRegression()
    pipe = make_pipeline(StandardScaler(), lr)
    return pipe
def split_data(df):
    feature_cols = df.columns[:-1]
    X = df[feature_cols]
    y = df["target"]
    return X, y
def track_with_mlflow(model, X_test, y_test, mlflow, model_metadata):
    mlflow.log_params(model_metadata)
    mlflow.log_metric("accuracy", model.score(X_test, y_test))
    mlflow.sklearn.log_model(model, "lr", registered_model_name="sklearn_lr")
def main():
    wine_data = load_wine()
    df = preprocess_data(wine_data)
    X, y = split_data(df)
    with mlflow.start_run():
        lr_pipe = setup_lr_pipeline()
        lr_pipe.fit(X, y)
        model_metadata = {"dataset": "wine"}
        track_with_mlflow(lr_pipe, X, y, mlflow, model_metadata)
if __name__ == "__main__":
    fire.Fire(main)
```

and we use ml_project file, which required to run for bash command in ipynb file

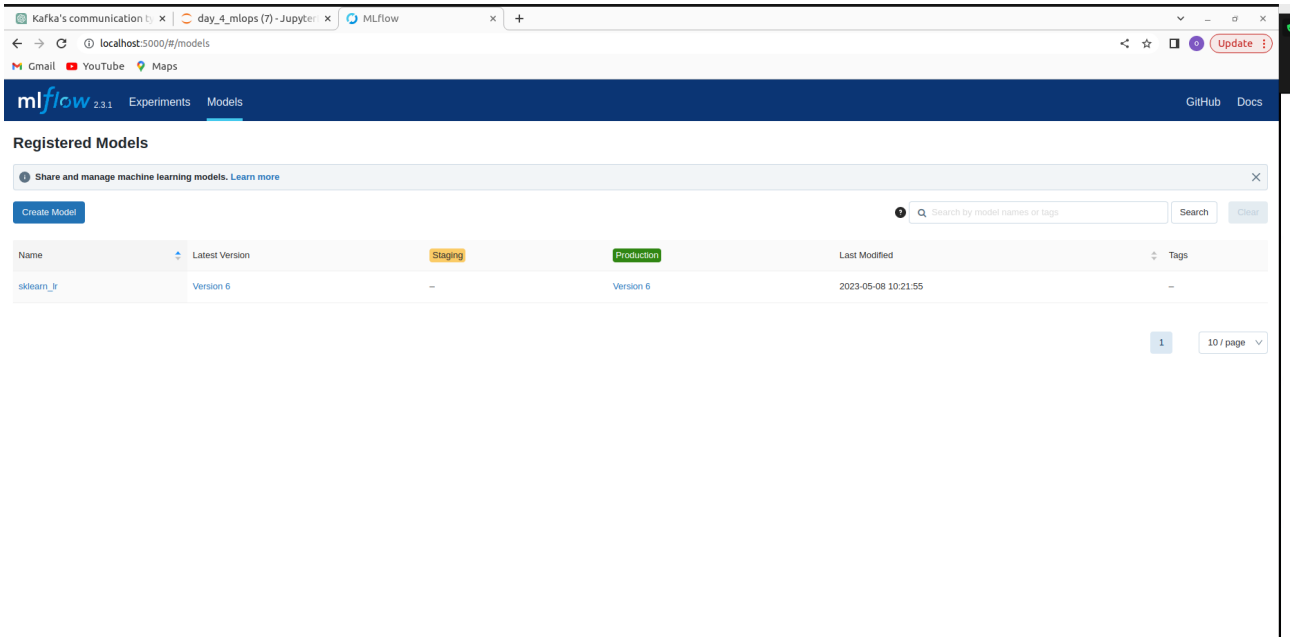


Name: Osama Abdul Razzak(2303.KHI.DEG.029)
Peer Name: Rahima Siddiqui(2303.KHI.DEG.030)

than we ml-flow server on 5000 port
by using following bash command

```
%%bash --bg

mlflow server --host 0.0.0.0 \
    --port 5000 \
    --backend-store-uri sqlite:///mlflow.db \
    --default-artifact-root ./mlruns
```



and check out ml_project file

```
%cat MLproject
```

```
name: basic_mlflow
# this file is used to configure Python package dependencies.
# it uses Anaconda, but it can be also alternatively configured to use pip.
conda_env: conda.yaml
# entry points can be ran using `mlflow run <project_name> -e <entry_point_name>
entry_points:
  main:
    # parameters is a key-value collection.
    parameters:
      command:
        type: str
        default: "train"
    command: "python train.py"
```

First we need to download data. We will use weather data from previous machine learning tutorial.

Name: Osama Abdul Razzak(2303.KHI.DEG.029)
Peer Name: Rahima Siddiqui(2303.KHI.DEG.030)

then we run our model

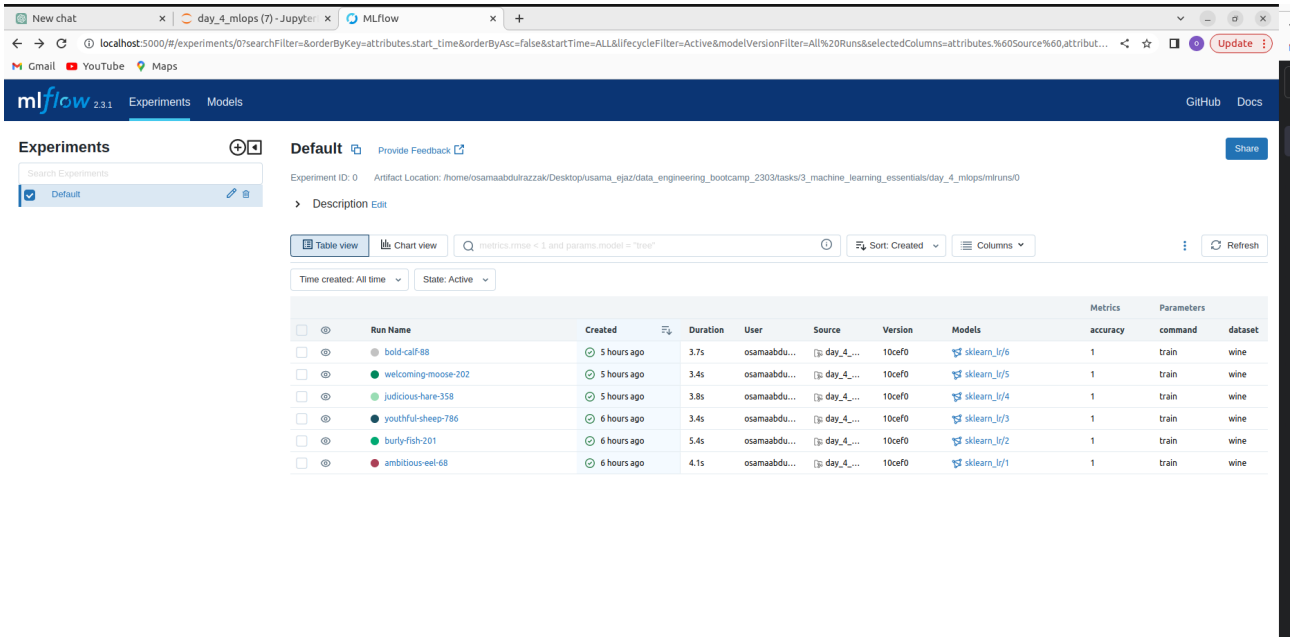
```
import sklearn
```

```
sklearn.__version__
```

```
'1.2.2'
```

```
%%bash
source mlflow_env_vars.sh
mlflow run .
```

```
2023/05/08 10:21:38 INFO mlflow.utils.conda: Conda environment mlflow-dd0fbdd40ba98798131458f29496394bd1a3fb33 already exists.
2023/05/08 10:21:38 INFO mlflow.projects.utils: === Created directory /tmp/tmprx1mwlr3 for downloading remote URIs passed to arguments of type 'path' ===
2023/05/08 10:21:38 INFO mlflow.projects.backend.local: === Running command 'source /home/osamaabdulrazzak/anaconda3/bin/../etc/profile.d/conda.sh && conda activate mlflow-dd0fbd
d40ba98798131458f29496394bd1a3fb33 1>&2 && python train.py' in run with ID 'ca65b85542f641e3971569bcf21cf286' ===
/home/osamaabdulrazzak/Desktop/usama_ejaz/data_engineering_bootcamp_2303/tasks/3_machine_learning_essentials/mlops-student/lib/python3.10/site-packages/_distutils_hack/__init__.p
y:33: UserWarning: Setuptools is replacing distutils.
  warnings.warn("Setuptools is replacing distutils.")
Registered model 'sklearn_lr' already exists. Creating a new version of this model...
2023/05/08 10:21:40 INFO mlflow.tracking._model_registry.client: Waiting up to 300 seconds for model version to finish creation. Model name: sklearn_lr, version 6
Created version '6' of model 'sklearn_lr'.
2023/05/08 10:21:40 INFO mlflow.projects: === Run (ID 'ca65b85542f641e3971569bcf21cf286') succeeded ===
```



and then give the path

```
%%bash
last_model_path=$(ls -tr mlruns/0/ | tail -1)
cat mlruns/0/$last_model_path/artifacts/lr/MLmodel
```

```
artifact_path: lr
flavors:
  python_function:
    env:
      conda: conda.yaml
      virtualenv: python_env.yaml
    loader_module: mlflow.sklearn
    model_path: model.pkl
    predict_fn: predict
    python_version: 3.10.6
  sklearn:
    code: null
    pickled_model: model.pkl
    serialization_format: cloudpickle
    sklearn_version: 1.2.2
mlflow_version: 2.3.1
model_uuid: 26013996b4b74346bd36fb829b2d7f93
run_id: ca65b85542f641e3971569bcf21cf286
utc_time_created: '2023-05-08 05:21:39.335761'
```

Name: Osama Abdul Razzak(2303.KHI.DEG.029)
Peer Name: Rahima Siddiqui(2303.KHI.DEG.030)

then run the another port for model prediction

```
%%bash --bg
source mlflow_env_vars.sh
mlflow --version
mlflow models serve -m models:/sklearn_lr/Production -p 5002 --env-manager=conda
```

here we did prediction on two row

[28]:
```
%%bash
data='[[13.27, 4.28, 2.26, 20.0, 120.0, 1.59, 0.69, 0.43, 1.35, 10.2, 0.59, 1.56, 835.0],[13.27, 4.28, 2.26, 20.0, 120.0, 1.59, 0.69, 0.43, 1.35, 10.2, 0.59, 1.56, 835.0]]'
echo $data
curl -d "{\"inputs\": $data}" -H 'Content-Type: application/json' 127.0.0.1:5002/invocations
```

```
[[13.27, 4.28, 2.26, 20.0, 120.0, 1.59, 0.69, 0.43, 1.35, 10.2, 0.59, 1.56, 835.0],[13.27, 4.28, 2.26, 20.0, 120.0, 1.59, 0.69, 0.43, 1.35, 10.2, 0.59, 1.56, 835.0]]
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   200  100    23  100   177   9725  74841 --:--:-- --:--:-- --:--:--   97k
{"predictions": [2, 2]}
```

[50]:
```
%%bash
data='[[14.23,1.71,2.43,15.6,127.0,2.80,3.06,0.28,2.29,5.64,1.04,3.92,1065.0],[14.23,1.71,2.43,15.6,127.0,2.80,3.06,0.28,2.29,5.64,1.04,3.92,1065.0]]'
echo $data

curl -d "{\"instances\": $data}" -H 'Content-Type: application/json' 127.0.0.1:5002/invocations
```

```
[[14.23,1.71,2.43,15.6,127.0,2.80,3.06,0.28,2.29,5.64,1.04,3.92,1065.0],[14.23,1.71,2.43,15.6,127.0,2.80,3.06,0.28,2.29,5.64,1.04,3.92,1065.0]]
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   181  100    23  100   158  12405  85221 --:--:-- --:--:-- --:--:--  176k
{"predictions": [0, 0]}
```

[52]:
```
%%bash
data='[[14.23,1.71,2.43,15.6,127.0,2.80,3.06,0.28,2.29,5.64,1.04,3.92,1065.0]]'
columns='["alcohol","malic_acid","ash","alcalinity_of_ash","magnesium","total_phenols","flavanoids","nonflavanoid_phenols","proanthocyanins","color_intensity","hue","od280/od315_
echo $data
curl -d "{\"dataframe_split\":{\"columns\":$columns, \"data\": $data}}" -H 'Content-Type: application/json' 127.0.0.1:5002/invocations
```

```
[[14.23,1.71,2.43,15.6,127.0,2.80,3.06,0.28,2.29,5.64,1.04,3.92,1065.0]]
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   331  100    20  100   311   5878  91416 --:--:-- --:--:-- --:--:--  107k
{"predictions": [0]}
```

Voila! We see that the model outputs correct predictions.