Lab: MLFlow Data

The mlflow.data module helps you record your model training and evaluation datasets to runs with MLflow Tracking, as well as retrieve dataset information from runs. It provides the following important interfaces:

- `Dataset` : Represents a dataset used in model training or evaluation, including features, targets, predictions, and metadata such as the dataset's name, digest (hash) schema, profile, and source. You can log this metadata to a run in MLflow Tracking using the mlflow.log_input() API. mlflow.data provides APIs for constructing Datasets from a variety of Python data objects, including Pandas DataFrames (mlflow.data.from_pandas()), NumPy arrays (mlflow.data.from_numpy()), Spark DataFrames (mlflow.data.from_spark() / mlflow.data.load_delta()), and more.
- `DatasetSource` : Represents the source of a dataset. For example, this may be a directory of files stored in S3, a Delta Table, or a web URL. Each Dataset references the source from which it was derived. A Dataset's features and targets may differ from the source if transformations and filtering were applied. You can get the DatasetSource of a dataset logged to a run in MLflow Tracking using the mlflow.data.get_source() API.

Lab Solution

Complete solution for this lab is available in the `lab3_mlflow_data.ipynb` notebook.

The following example demonstrates how to use mlflow.data to log a training dataset to a run, retrieve information about the dataset from the run, and load the dataset's source.

```
import mlflow.data
import pandas as pd
from mlflow.data.pandas_dataset import PandasDataset

# Construct a Pandas DataFrame using iris flower data from a web URL
dataset_source_url = "http://archive.ics.uci.edu/ml/machine-learning-databases/wine-
quality/winequality-red.csv"
df = pd.read_csv(dataset_source_url)
# Construct an MLflow PandasDataset from the Pandas DataFrame, and specify the web URL
# as the source
dataset: PandasDataset = mlflow.data.from_pandas(df, source=dataset_source_url)

with mlflow.start_run():
    # Log the dataset to the MLflow Run. Specify the "training" context to indicate
that the
    # dataset is used for model training
    mlflow.log_input(dataset, context="training")

# Retrieve the run, including dataset information
run = mlflow.get_run(mlflow.last_active_run().info.run_id)
dataset_info = run.inputs.dataset_inputs[0].dataset
print(f"Dataset name: {dataset_info.name}")
print(f"Dataset digest: {dataset_info.digest}")
print(f"Dataset profile: {dataset_info.profile}")
print(f"Dataset schema: {dataset_info.schema}")

# Load the dataset's source, which downloads the content from the source URL to the
local
```

```
# filesystem
dataset_source = mlflow.data.get_source(dataset_info)
dataset_source.load()
```

pandas

Constructs a PandasDataset instance from a Pandas DataFrame, optional targets, optional predictions, and source.

```
import mlflow
import pandas as pd

x = pd.DataFrame(
    [["tom", 10, 1, 1], ["nick", 15, 0, 1], ["juli", 14, 1, 1]],
    columns=["Name", "Age", "Label", "ModelOutput"],
)
dataset = mlflow.data.from_pandas(x, targets="Label", predictions="ModelOutput")
```

NumPy

Constructs a NumpyDataset object from NumPy features, optional targets, and source. If the source is path like, then this will construct a DatasetSource object from the source path. Otherwise, the source is assumed to be a DatasetSource object.

Basic Example

```
import mlflow
import numpy as np

x = np.random.uniform(size=[2, 5, 4])
y = np.random.randint(2, size=[2])
dataset = mlflow.data.from_numpy(x, targets=y)
```

Dict Example

```
import mlflow
import numpy as np

x = {
    "feature_1": np.random.uniform(size=[2, 5, 4]),
    "feature_2": np.random.uniform(size=[2, 5, 4]),
}
y = np.random.randint(2, size=[2])
dataset = mlflow.data.from_numpy(x, targets=y)
```