**1. Project Setup with Git**

Start by creating a new project directory:

*mkdir mlflow-data-versioning-demo*

*cd mlflow-data-versioning-demo*

*git init*

Create the following directory structure:

mlflow-data-versioning-demo

├── data/

├── models/

├── notebooks/

├── src/

├── requirements.txt

└── README.md

For README.md you can run below code-

*echo "# My Project" > README.md*

**Step 1: Create a Virtual Environment**

python -m venv venv

**Step 2: Activate the Virtual Environment**

For **Windows Command Prompt**:

venv\Scripts\activate

**Install Required Packages**

Now that your virtual environment is activated, install the necessary packages:

pip install pandas scikit-learn mlflow gitpython

**Add dependencies to requirements.txt**

pandas

scikit-learn

mlflow

gitpython

**Commit the initial setup:**

git add .

git commit -m "Initial project setup"

**Data Preparation and Versioning**

Download a sample dataset (e.g., the **Iris dataset**) and save it in the data/ directory.

*import pandas as pd*

*def load\_data():*

*url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/iris.csv"*

*columns = ["sepal\_length", "sepal\_width", "petal\_length", "petal\_width", "class"]*

*try:*

*# Load the dataset*

*df = pd.read\_csv(url, names=columns)*

*print("Data loaded successfully.")*

*# Save to CSV*

*df.to\_csv("data/iris.csv", index=False)*

*print("Data saved to data/iris.csv")*

*except Exception as e:*

*print(f"Error occurred: {e}")*

*if \_\_name\_\_ == "\_\_main\_\_":*

*load\_data()*

**Run the script:**

*python src/data\_loader.py*

**Version control the data file:**

*git add data/iris.csv*

*git commit -m "Added Iris dataset"*

**Setting Up MLflow for Experiment Tracking**

Create an MLflow tracking script:

*# src/train.py*

*import pandas as pd*

*import mlflow*

*import mlflow.sklearn*

*from sklearn.model\_selection import train\_test\_split*

*from sklearn.ensemble import RandomForestClassifier*

*from sklearn.metrics import accuracy\_score*

*# Load data*

*df = pd.read\_csv("data/iris.csv")*

*X = df.drop(columns=["class"])*

*y = df["class"]*

*# Split the data*

*X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)*

*# Set up MLflow experiment*

*mlflow.set\_experiment("Iris Classification")*

*with mlflow.start\_run():*

*# Train the model*

*model = RandomForestClassifier(n\_estimators=100, random\_state=42)*

*model.fit(X\_train, y\_train)*

*# Make predictions*

*y\_pred = model.predict(X\_test)*

*accuracy = accuracy\_score(y\_test, y\_pred)*

*# Log metrics, parameters, and model*

*mlflow.log\_param("n\_estimators", 100)*

*mlflow.log\_metric("accuracy", accuracy)*

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

*print(f"Model accuracy: {accuracy}")*

**Run the training script:**

*python src/train.py*

**Viewing MLflow Dashboard**

Start the MLflow tracking server to view the experiment results:

*mlflow ui*

**Model Training and Tracking with MLflow**

Now that you've logged a model, let's update the data and re-run the experiment to see how MLflow handles versioning.

* Modify the dataset (e.g., add noise to a feature) and save it as iris\_v2.csv.
* Adjust the data\_loader.py script to load the new version.

*git add data/iris\_v2.csv*

*git commit -m "Updated dataset with noise"*

Re-run the train.py script and observe the new run in MLflow. You'll see separate logs for different versions.

**Pushing Code, Data, and MLflow Artifacts to Git**

Finally, push all your changes to a remote Git repository:

*git remote add origin <your-repo-url>*

*git branch -M main*

*git push -u origin main*