**Best Practice Setup on Windows**

1. Install MLflow:

pip install mlflow

1. Start MLflow Tracking Server:

mlflow server --backend-store-uri sqlite:///mlflow.db --default-artifact-root ./mlruns --host 127.0.0.1 --port 5000

1. Open MLflow UI in browser:

http://127.0.0.1:5000

1. Run your training script — now registration works.

**Run thebelow code:**

 Loads the **diabetes dataset** and splits it into training and testing sets.

 Trains a **Random Forest Regressor** with n\_estimators=50.

 Configures **MLflow tracking** to use a local server (http://localhost:5000) and sets up an experiment named "Diabetes\_Model".

 Starts an MLflow run and **logs model parameters** (like number of estimators).

 **Saves the trained model** to MLflow’s artifact store.

 Retrieves the MLflow run\_id to reference the run.

 **Registers the trained model** in MLflow’s Model Registry as "Diabetes\_Predictor\_Model".

import mlflow                               # Import MLflow for tracking experiments and models

import mlflow.sklearn                      # MLflow module to log and load sklearn models

from sklearn.datasets import load\_diabetes  # Import diabetes dataset from sklearn

from sklearn.model\_selection import train\_test\_split  # For splitting dataset into training/testing sets

from sklearn.ensemble import RandomForestRegressor    # Random Forest model for regression tasks

# Load data

X, y = load\_diabetes(return\_X\_y=True)      # Load diabetes dataset; X = features, y = target values

X\_train, X\_test, y\_train, y\_test = train\_test\_split(

    X, y, test\_size=0.2, random\_state=42   # Split data: 80% train, 20% test, fixed random seed

)

# Train model

model = RandomForestRegressor(

    n\_estimators=50, random\_state=42       # Create Random Forest Regressor with 50 trees

)

model.fit(X\_train, y\_train)                # Train the model on training data

# Start MLflow tracking

mlflow.set\_tracking\_uri("http://localhost:5000")  # Set MLflow tracking server URI (local server)

mlflow.set\_experiment("Diabetes\_Model")           # Set experiment name in MLflow to group runs

# Start a new MLflow run to track this experiment

with mlflow.start\_run():

    mlflow.log\_param("n\_estimators", 50)           # Log model parameter (number of estimators)

    mlflow.sklearn.log\_model(model, "model")       # Save and log trained model in MLflow

    run\_id = mlflow.active\_run().info.run\_id       # Get unique Run ID for this MLflow run

# Register model in MLflow Model Registry

model\_uri = f"runs:/{run\_id}/model"                # Create model URI from run ID

mlflow.register\_model(model\_uri, "Diabetes\_Predictor\_Model")  # Register model with a given name



