databricksLogRunsToExperiment

Log runs to an experiment

Types of experiments

There are two types of experiments in MLflow: *notebook* and *workspace*.

- A notebook experiment is associated with a specific notebook. Databricks creates a notebook experiment by default when a run is started using `mlflow.start_run()` and there is no active experiment.
- Workspace experiments are not associated with any notebook, and any notebook can log a run to these experiments by using the experiment name or the experiment ID when initiating a run.

This notebook creates a Random Forest model on a simple dataset and uses the MLflow Tracking API to log the model and selected model parameters and metrics.

```
# Import the dataset from scikit-learn and create the training and test datasets.
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_diabetes

db = load_diabetes()
X = db.data
y = db.target
X_train, X_test, y_train, y_test = train_test_split(X, y)
```

By default, MLflow runs are logged to the notebook experiment, as illustrated in the following code block.

```
import mlflow
import mlflow.sklearn
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error

# In this run, neither the experiment_id nor the experiment_name parameter is
provided. MLflow automatically creates a notebook experiment and logs runs to it.
# Access these runs using the Experiment sidebar. Click Experiment at the upper
right of this screen.
with mlflow.start_run():
  n_estimators = 100
  max_depth = 6
  max_features = 3
  # Create and train model
  rf = RandomForestRegressor(n_estimators = n_estimators, max_depth = max_depth,
max_features = max_features)
  rf.fit(X_train, y_train)
  # Make predictions
  predictions = rf.predict(X_test)

  # Log parameters
  mlflow.log_param("num_trees", n_estimators)
  mlflow.log_param("maxdepth", max_depth)
  mlflow.log_param("max_feat", max_features)

  # Log model
  mlflow.sklearn.log_model(rf, "random-forest-model")

  # Create metrics
  mse = mean_squared_error(y_test, predictions)

  # Log metrics
  mlflow.log_metric("mse", mse)
```

To log MLflow runs to a workspace experiment, use `mlflow.set_experiment()` as illustrated in the following code block. An alternative is to set the experiment_id parameter in `mlflow.start_run()` ; for example, `mlflow.start_run(experiment_id=1234567)` .