Highlight Note

Running an Experiment Inline

You can use the Python SDK to run an Azure Machine Learning experiment inline, for example in a notebook. The key steps you need to perform in your code are:

1. Create or retrieve a named experiment in your workspace.
2. Start a run of the experiment – retrieving an active **Run** object.
3. Use the **Run** object to log metrics you want to review later.
4. Save or upload files to the run's **outputs** folder so that they are stored in the run history.
5. Complete the run

After completing the run, you can view its logs, metrics and outputs in Azure Machine Learning studio or use the SDK to retrieve them in code. There's also a widget that you can use to display run details in a notebook.

Logging Metrics

Every experiment generates log files that include the messages that would be written to the terminal during interactive execution. This enables you to use simple `print` statements to write messages to the log. However, if you want to record named metrics for comparison across runs, you can do so by using the **Run** object; which provides a range of logging functions specifically for this purpose. These include:

- **log**: Record a single named value.
- **log_list**: Record a named list of values.
- **log_row**: Record a row with multiple columns.
- **log_table**: Record a dictionary as a table.
- **log_image**: Record an image file or a plot.

More Information: For more information about logging metrics during experiment runs, see **Monitor Azure ML experiment runs and metrics** in the Azure Machine Learning documentation.

For example, following code records the number of observations (records) in a CSV file:

```
from azureml.core import Experiment
import pandas as pd

# Create an Azure ML experiment in your workspace
experiment = Experiment(workspace=ws, name='my-experiment')

# Start logging data from the experiment
run = experiment.start_logging()

# load the data and count the rows
data = pd.read_csv('data.csv')
row_count = (len(data))
```

```
# Log the row count
run.log('observations', row_count)

# Complete the experiment
run.complete()
```

Retrieving and Viewing Logged Metrics

You can view the metrics logged by an experiment run in Azure Machine Learning studio or by using the **RunDetails** widget in a notebook, as shown here:

```
from azureml.widgets import RunDetails

RunDetails(run).show()
```

You can also retrieve the metrics using the **Run** object's **get_metrics** method, which returns a JSON representation of the metrics, as shown here:

```
import json

# Get logged metrics
metrics = run.get_metrics()
print(json.dumps(metrics, indent=2))
```

The previous code produces output similar to this:

```
{
  "observations": 15000
}
```

Experiment Output Files

In addition to logging metrics, an experiment can generate output files. Often these are trained machine learning models, but you can save any sort of file and make it available as an output of your experiment run. The output files of an experiment are saved in its **outputs** folder.

The technique you use to add files to the outputs of an experiment depend on how your running the experiment. The examples shown so far control the experiment lifecycle inline in your code, and when taking this approach you can upload local files to the run's **outputs** folder by using the **Run** object's **upload_file** method in your experiment code as shown here:

```
run.upload_file(name='outputs/sample.csv', path_or_stream='./sample.csv')
```

When running an experiment in a remote compute context (which we'll discuss later in this course), any files written to the **outputs** folder in the compute context are automatically uploaded to the run's **outputs** folder when the run completes.

Whichever approach you use to run your experiment, you can retrieve a list of output files from the **Run** object like this:

```
import json
```

```
files = run.get_file_names()
print(json.dumps(files, indent=2))
```

The previous code produces output similar to this:

```
[
  "outputs/sample.csv"
]
```