Python azureml SDK functions and classes

Workspace
- create(), from_config()

Experiment
- Experiment(ws = workspace, name = ' ')

Run
- run = experiment.submit( )

- Model.register(ws, model_path = ".pkl", model_name = " ")

- For compute target:
  - runconfig
    - AmlCompute
    - CondaDependencies()
  - Submit all this to ScriptRunConfig()
  - ScriptRunConfig(source_dir = ' ', script = 'train.py', run_config = ' ')
  - run = Experiment.submit(config = ' ')
  - run.wait_for_completion( )

- Pipeline
  - PythonScriptStep()
  - Pipeline(ws, steps = [, ,]) ← steps can be linked
  - pip_run = experiment.submit(pipeline)

- Data
  - Dataset() - existing Azure datastore
  - PipelineDataset() - typed tabular data
  - PipelineData() - intermediary file or data between two steps in a pipeline.

- AutoMLConfig()
  - best_model = run.get_output()

- Model Deploy
  - InferenceConfig()
  - register the model to deploy.
  - AciWebservice.deploy_configuration() ← deployment config.
  - Model.deploy(ws, name = ' ', models = [ ], inference_config, deployment_config)

- Dataset(), TabularDataset(), FileDataset()

① Workspace

ws = Workspace.from_config() ws-name

② Experiment

run = experiment.start_logging()

run.log()

run.complete(), get_details(), get_metrics().

RunDetails(run).show()

③ Experiment Script:

run = Run.get_context()

RunConfiguration(), ScriptRunConfig(), experiment.submit().

Estimator()

Create a ArgumentParser for parameterization.

Framework specific Estimator → Sklearn

④ Datastore

= get_default_datastore()

- upwd_files()
- as_download() → compute_context, ok for small data.
- Tabular_from_delimited_files()
- .register() for registration.
-
- Sklearn(inputs = [—.as_named_input('')]

Estimator

pass data object into the estimator