100 XP

Run a training script

5 minutes

You can use a **ScriptRunConfig** to run a script-based experiment that trains a machine learning model.

Writing a script to train a model

When using an experiment to train a model, your script should save the trained model in the **outputs** folder. For example, the following script trains a model using Scikit-Learn, and saves it in the **outputs** folder using the **joblib** package:

> **Note**
>
> This sample code is an incomplete extract that shows the concept of training a model using Scikit-Learn.

Python Copy

```
from azureml.core import Run
import pandas as pd
import numpy as np
import joblib
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

# Get the experiment run context
run = Run.get_context()

# Prepare the dataset
diabetes = pd.read_csv('data.csv')
X, y = diabetes[['Feature1','Feature2','Feature3']].values,
diabetes['Label'].values
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30)

# Train a logistic regression model
reg = 0.1
model = LogisticRegression(C=1/reg, solver="liblinear").fit(X_train, y_train)

# calculate accuracy
y_hat = model.predict(X_test)
acc = np.average(y_hat == y_test)
run.log('Accuracy', np.float(acc))
```

```
# Save the trained model
os.makedirs('outputs', exist_ok=True)
joblib.dump(value=model, filename='outputs/model.pkl')

run.complete()
```

To prepare for an experiment that trains a model, a script like this is created and saved in a folder. For example, you could save this script as **training_script.py** in a folder named **training_folder**. Since the script includes code to load training data from **data.csv**, this file should also be saved in the folder.

Running the script as an experiment

To run the script, create a **ScriptRunConfig** that references the folder and script file. You generally also need to define a Python (Conda) environment that includes any packages required by the script. In this example, the script uses Scikit-Learn so you must create an environment that includes that. The script also uses Azure Machine Learning to log metrics, so you need to remember to include the **azureml-defaults** package in the environment.

Python Copy

```
from azureml.core import Experiment, ScriptRunConfig, Environment
from azureml.core.conda_dependencies import CondaDependencies

# Create a Python environment for the experiment
sklearn_env = Environment("sklearn-env")

# Ensure the required packages are installed
packages = CondaDependencies.create(conda_packages=['scikit-learn','pip'],
                                    pip_packages=['azureml-defaults'])
sklearn_env.python.conda_dependencies = packages

# Create a script config
script_config = ScriptRunConfig(source_directory='training_folder',
                                script='training.py',
                                environment=sklearn_env)

# Submit the experiment
experiment = Experiment(workspace=ws, name='training-experiment')
run = experiment.submit(config=script_config)
run.wait_for_completion()
```

Next unit: Using script parameters

Continue >