

Run a training script with the Python SDK

You can use the Python SDK for Azure Machine Learning to submit scripts as jobs. By using jobs, you can easily keep track of the input parameters and outputs when training a machine learning model.

Before you start

You'll need the latest version of the `azureml-ai-ml` package to run the code in this notebook. Run the cell below to verify that it is installed.

Note: If the `azure-ai-ml` package is not installed, run `pip install azure-ai-ml` to install it.

```
pip show azure-ai-ml
```

Connect to your workspace

With the required SDK packages installed, now you're ready to connect to your workspace.

To connect to a workspace, we need identifier parameters - a subscription ID, resource group name, and workspace name. Since you're working with a compute instance, managed by Azure Machine Learning, you can use the default values to connect to the workspace.

```
from azure.identity import DefaultAzureCredential,  
InteractiveBrowserCredential  
from azure.ai.ml import MLClient  
  
try:  
    credential = DefaultAzureCredential()  
    # Check if given credential can get token successfully.  
    credential.get_token("https://management.azure.com/.default")  
except Exception as ex:  
    # Fall back to InteractiveBrowserCredential in case  
DefaultAzureCredential not work  
    credential = InteractiveBrowserCredential()  
  
# Get a handle to workspace  
ml_client = MLClient.from_config(credential=credential)
```

Use the Python SDK to train a model

To train a model, you'll first create the `diabetes_training.py` script in the `src` folder. The script uses the `diabetes.csv` file in the same folder as the training data.

```
%%writefile src/diabetes-training.py  
# import libraries  
import pandas as pd  
import numpy as np
```

```

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve

# load the diabetes dataset
print("Loading Data...")
diabetes = pd.read_csv('diabetes.csv')

# separate features and labels
X, y =
diabetes[['Pregnancies','PlasmaGlucose','DiastolicBloodPressure','Tric-
epsThickness','SerumInsulin','BMI','DiabetesPedigree','Age']].values,
diabetes['Diabetic'].values

# split data into training set and test set
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.30, random_state=0)

# set regularization hyperparameter
reg = 0.01

# train a logistic regression model
print('Training a logistic regression model with regularization rate
of', reg)
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)
print('Accuracy:', acc)

# calculate AUC
y_scores = model.predict_proba(X_test)
auc = roc_auc_score(y_test,y_scores[:,1])
print('AUC: ' + str(auc))

```

Run the cell below to submit the job that trains a classification model to predict diabetes.

```

from azure.ai.ml import command

# configure job
job = command(
    code="./src",
    command="python diabetes-training.py",
    environment="AzureML-sklearn-0.24-ubuntu18.04-py37-cpu@latest",
    compute="aml-cluster",
    display_name="diabetes-pythonv2-train",
    experiment_name="diabetes-training"

```

```
)  
  
# submit job  
returned_job = ml_client.create_or_update(job)  
aml_url = returned_job.studio_url  
print("Monitor your job at", aml_url)
```