Highlight Note

Adding Explanations to Training Experiments

When you use an estimator or a script to train a model in an Azure Machine Learning experiment, you can create an explainer and upload the explanation it generates to the run output for later analysis.

Creating an Explanation in the Experiment Script

To create an explanation in the experiment script, you'll need to ensure that the **azureml-interpret** package is installed in the run environment. Then you can use these to create an explanation from your trained model and upload it to the run outputs.

```
# Import Azure ML run library
from azureml.core.run import Run
from azureml.interpret.explanation.explanation_client import ExplanationClient
from interpret.ext.blackbox import TabularExplainer
# other imports as required

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

# code to train model goes here

# Get explanation
explainer = TabularExplainer(model, X_train, features=features, classes=labels)
explanation = explainer.explain_global(X_test)

# Get an Explanation Client and upload the explanation
explain_client = ExplanationClient.from_run(run)
explain_client.upload_model_explanation(explanation, comment='Tabular Explanation')

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

Viewing the Explanation

You can view the explanation you created for your model in the **Explanations** tab for the run in Azure Machine learning studio.

You can also use the **ExplanationClient** object to download the explanation in Python.

```
from azureml.interpret.explanation.explanation_client import ExplanationClient

client = ExplanationClient.from_run_id(workspace=ws,
                                       experiment_name=experiment.experiment_name,
                                       run_id=run.id)
explanation = client.download_model_explanation()
feature_importances = explanation.get_feature_importance_dict()
```