Highlight Note

Explainers

You can use the Azure Machine Learning SDK to create explainers for models, even if they were not trained using an Azure Machine Learning experiment.

Creating an Explainer

To interpret a model, you must install the **azureml-interpret** package and use it to create an explainer. There are multiple types of explainer, including:

- **MimicExplainer** - An explainer that creates a *global surrogate model* that approximates your trained model and can be used to generate explanations. This explainable model must have the same kind of architecture as your trained model (for example, linear or tree-based).
- **TabularExplainer** - An explainer that acts as a wrapper around various SHAP explainer algorithms, automatically choosing the one that is most appropriate for your model architecture.
- **PFIExplainer** - a *Permutation Feature Importance* explainer that analyzes feature importance by shuffling feature values and measuring the impact on prediction performance.

The following code sample shows how to create an instance of each of these explainer types:

```
# MimicExplainer
from interpret.ext.blackbox import MimicExplainer
from interpret.ext.glassbox import DecisionTreeExplainableModel

mim_explainer = MimicExplainer(model=loan_model,
                               initialization_examples=X_test,
                               explainable_model = DecisionTreeExplainableModel,
                               features=['loan_amount','income','age','marital_status'],
                               classes=['reject', 'approve'])


# TabularExplainer
from interpret.ext.blackbox import TabularExplainer

tab_explainer = TabularExplainer(model=loan_model,
                               initialization_examples=X_test,
                               features=['loan_amount','income','age','marital_status'],
                               classes=['reject', 'approve'])


# PFIExplainer
from interpret.ext.blackbox import PFIExplainer

pfi_explainer = PFIExplainer(model = loan_model,
                               features=['loan_amount','income','age','marital_status'],
                               classes=['reject', 'approve'])
```

Explaining Global Feature Importance

To retrieve global importance values for the features in your model, you call the **explain_global()** method of your explainer to get a global explanation, and then use the **get_feature_importance_dict()** method to get a dictionary of the feature importance values:

```
# MimicExplainer
global_mim_explanation = mim_explainer.explain_global(X_train)
global_mim_feature_importance = global_mim_explanation.get_feature_importance_dict()


# TabularExplainer
global_tab_explanation = tab_explainer.explain_global(X_train)
global_tab_feature_importance = global_tab_explanation.get_feature_importance_dict()


# PFIExplainer
global_pfi_explanation = pfi_explainer.explain_global(X_train, y_train)
global_pfi_feature_importance = global_pfi_explanation.get_feature_importance_dict()
```

Note: The code is the same for **MimicExplainer** and **TabularExplainer**. The **PFIExplainer** requires the actual labels that correspond to the test features.

Explaining Local Feature Importance

To retrieve local feature importance from a **MimicExplainer** or a **TabularExplainer**, you must call the **explain_local()** method of your explainer, specifying the subset of cases you want to explain. Then you can use the **get_ranked_local_names()** and **get_ranked_local_values()** methods to retrieve dictionaries of the feature names and importance values, ranked by importance.

```
# MimicExplainer
local_mim_explanation = mim_explainer.explain_local(X_test[0:5])
local_mim_features = local_mim_explanation.get_ranked_local_names()
local_mim_importance = local_mim_explanation.get_ranked_local_values()


# TabularExplainer
local_tab_explanation = tab_explainer.explain_local(X_test[0:5])
local_tab_features = local_tab_explanation.get_ranked_local_names()
local_tab_importance = local_tab_explanation.get_ranked_local_values()
```

Note: The code is the same for **MimicExplainer** and **TabularExplainer**. The **PFIExplainer** doesn't support local feature importance explanations.