

Article

[José Pereira](#) · Nov 16, 2020 3m read

[Open Exchange](#)

Using IntegratedML to create a ML adapter for IRIS Interoperability

Update: added support for regression model
Hi everyone!

In this brief article, I'll show you how to write an adapter for IRIS Interoperability for use ML models managed by IRIS IntegratedML.

The adapter

The adapter just uses IntegratedML SQL functions [PREDICT](#) and [PROBABILITY](#), to get the predicted class from model and its probability. It's just a simple SQL:

```
33 Method Classify(pSampleRequest As %RegisteredObject) As dc.Ens.Adapter.ClassificationResult
34 {
35     Set ..Request = pSampleRequest
36     Set sqlSample = ..GetSampleFromRequest()
37     Set sql =
38     "SELECT " _
39     "PROBABILITY("_.Model_" FOR 1) AS Probability, " _
40     "PREDICT("_.Model_" ) AS Predicted " _
41     "FROM ("_sqlSample_")"
42     Set rset = ##class(%SQL.Statement).%ExecDirect(, sql)
43
44     Set result = ##class(dc.Ens.Adapter.ClassificationResult).%New()
45     If (rset.%Next()) {
46         Set result.Probability = rset.Probability
47         Set result.Predicted = rset.Predicted
48     }
49     Return result
50 }
```

Notice that the model name is referenced by Model property. Such property must be defined in host class that uses the adapter, otherwise an exception will be thrown. For example:

The model list in the adaptor settings is done by two steps:

1. Creating a method into a class which extends %ZEN.Portal.ContextSearch to load all classification models and returning them (dc.Ens.Adapter.ClassificationMLContextSearch)

2. Configuring such class and method as feeder for property Model into SETTINGS parameter in the adaptor class (dc.Ens.Adapter.ClassificationMLAdapter)

For regression models, there's the class dc.Ens.Adapter.RegressionMLContextSearch, which loads all regression models.

In this case, the MODETYPE filter is set to 'regression' instead of 'classification':

Using the adapter

For demonstration, I simulated a simple credit card transaction payment system with fraud detection capabilities, using a ML classification model. When a suspicious transaction is detected, an alert is issued.

In order to use the adapter, create a host class (a Business Process or Business Operation class) which uses as adapter the class dc.ENS.Adapter.ClassificationMLAdapter.

Now, you can use adapter's method Classify(), and provide a sample of features expected by the model:

In order to use it, create a host class (a Business Process or Business Operation class) which uses as adapter the class dc.ENS.Adapter.ClassificationMLAdapter

You can use them as your needs. In the example, just the result for fraud prediction was necessary, so the Business Operation class just use value returned into Predicted property:

For regression models, the results is modeled by dc.Ens.Adapter.RegressionResult class. This class has a property called Estimated.

For get a estimation value from a sample, the adapter class for regression model has the Estimate method.

The final result is displayed below:

The complete code is available in [OpenExchange](#).
Hope this could be useful.
José

[#IntegratedML](#) [#Interoperability](#) [#Ensemble](#) [#InterSystems](#) [IRIS](#)
[Check the related application on InterSystems Open Exchange](#)

Source URL: <https://community.intersystems.com/post/using-integratedml-create-ml-adapter-iris-interoperability>