

Artículo

[Eduardo Anglada](#) · 22 ene, 2021 Lectura de 3 min

[Open Exchange](#)

Cómo utilizar IntegratedML para crear un adaptador de Machine Learning para la interoperabilidad de IRIS

Actualización: se añadió soporte para el modelo de regresión
¡Hola a todos!

En este breve artículo, os mostraré cómo escribir un adaptador para utilizar la interoperabilidad de IRIS con los modelos de Machine Learning (ML) administrados por IntegratedML de IRIS.

El adaptador

El adaptador solo utiliza las funciones SQL [PREDICT](#) y [PROBABILITY](#) de IntegratedML. Es solo una sencilla consulta 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 }
```

Tened en cuenta que el nombre del modelo se obtiene a partir de la propiedad denominada "Model". Dicha propiedad debe estar definida en la clase host que utiliza el adaptador, de lo contrario se producirá una excepción. Por ejemplo:

La lista de los modelos en la configuración del adaptador se realiza en dos pasos:

1. Creación de un método en una clase que sea hija de %ZEN.Portal.ContextSearch para tener todos los modelos de clasificación y que los devuelva (dc.Ens.Adapter.ClassificationMLContextSearch)

2. Configuración de dicha clase y método como un alimentador para la propiedad Model que se encuentra en el parámetro SETTINGS en la clase del adaptador (dc.Ens.Adapter.ClassificationMLAdapter)

Para los modelos de regresión, existe la clase dc.Ens.Adapter.RegressionMLContextSearch, que carga todos los modelos de regresión.

En este caso, el filtro MODETYPE se configuró como 'regression' en vez de 'classification':

Cómo utilizar el adaptador

Para la demostración, simulé un sencillo sistema de pago para las transacciones con tarjeta de crédito, con capacidad de detectar fraudes mediante un modelo de clasificación con ML. Cuando detecta una transacción sospechosa, emite una alerta.

Para utilizar el adaptador, cread una clase host (una clase Business Process o Business Operation) que utiliza como adaptador la clase dc.ENS.Adapter.ClassificationMLAdapter.

Ahora, podéis utilizar el método del adaptador Classify(), y proporcionar una muestra de las características esperadas por el modelo:

Para utilizarlo, cread una clase host (una clase Business Process o Business Operation) que utilice como adaptador la clase dc.ENS.Adapter.ClassificationMLAdapter

Podéis usarlos según vuestras necesidades. En el ejemplo, solo era necesario el resultado para la predicción de fraudes, por lo que la clase Business Operation solo utiliza el valor devuelto en la propiedad Predicted:

Para los modelos de regresión, los resultados se modelan mediante la clase dc.Ens.Adapter.RegressionResult. Esta clase tiene una propiedad llamada Estimated.

Para obtener un valor estimado a partir de una muestra, la clase adaptador para el modelo de regresión tiene el método Estimate.

El resultado final se muestra a continuación:

El código completo está disponible en [OpenExchange](#).

Espero que os resulte útil.

José

[#IntegratedML](#) [#Interoperabilidad](#) [#Ensemble](#) [#InterSystems IRIS](#)
[Ir a la aplicación en InterSystems Open Exchange](#)

fuelle: <https://es.community.intersystems.com/post/c%C3%B3mo-utilizar-integratedml-para-crear-un-adaptador-de-machine-learning-para-la>