

Artículo

[Alberto Fuentes](#) · 3 abr, 2020 Lectura de 4 min

Cómo cargar un modelo de ML a InterSystems IRIS (PMML)

¡Hola a todos!

Hoy hablaremos de cómo cargar un modelo de *Machine Learning* en IRIS y cómo probarlo, a través de un artículo genial de [Niyaz Khafizov](#).

Nota: lo siguiente se realizó utilizando Ubuntu 18.04, Apache Zeppelin 0.8.0 y Python 3.6.5.

Introducción

Actualmente, muchas de las diferentes herramientas que están disponibles para *data mining* permiten desarrollar modelos predictivos y analizar los datos con gran facilidad. La plataforma de datos InterSystems IRIS proporciona un soporte sólido para analizar grandes cantidades de datos e incrementar la velocidad de las aplicaciones, al tiempo que ofrece interoperabilidad junto con modernas herramientas de *data mining*.

En esta serie de artículos estamos explorando las capacidades de InterSystems IRIS para la minería de datos . En el primer [artículo](#) configuramos nuestra infraestructura y nos preparamos para empezar. En el segundo [artículo](#) construimos nuestro primer modelo predictivo para predecir las especies de flores utilizando instrumentos de Apache Spark y Apache Zeppelin. En este artículo, construiremos un modelo KMeans PMML y lo probaremos en InterSystems IRIS.

InterSystems IRIS proporciona excelentes funcionalidades para ejecutar PMML. Puedes subir tu modelo y probarlo con cualquier tipo de datos, IRIS mostrará su exactitud, precisión, F-score y muchas cosas más.

Comprobación de los requisitos

Primero, descarga [jpmml](#) (echa un vistazo a la tabla y selecciona la versión adecuada) y colócalo en cualquier directorio. Si utilizas Scala, esto será suficiente.

PySpark2PMML must be paired with JPMML-SparkML based on the following compatibility matrix:

Apache Spark version	JPMML-SparkML development branch	JPMML-SparkML uber-JAR file
2.0.X	1.1.X	1.1.20
2.1.X	1.2.X	1.2.12
2.2.X	1.3.X	1.3.8
2.3.X	master	1.4.5

Si utilizas Python, ejecuta la siguiente línea de comando en el terminal

```
pip3 install --user --upgrade git+https://github.com/jpmml/pyspark2pmml.git
```

Después del mensaje de que la instalación se realizó con éxito, ve a **Spark Dependencies** y añade la dependencia al jpmml descargado:

Dependencies

artifact	exclude
/home/guardian/Desktop/TMP/dev/java/lib/JDK18/intersystems-spark-1.0.0.jar	
/home/guardian/Desktop/TMP/dev/java/lib/JDK18/intersystems-jdbc-3.0.0.jar	
/home/guardian/Desktop/TMP/dev/java/lib/JDK18/jpmml-sparkml-executable-1.3.8.jar	

Crear modelo KMeans

El builder de PMML utiliza *pipelines*, por eso modificamos un poco el código escrito en el [artículo](#) anterior. Ejecuta el siguiente código en Zeppelin:

```
%pyspark
from pyspark.ml.linalg import Vectors
from pyspark.ml.feature import VectorAssembler
from pyspark.ml.clustering import KMeans
from pyspark.ml import Pipeline
from pyspark.ml.feature import RFormula
from pyspark2pmml import PMMLBuilder

dataFrame=spark.read.format("com.intersystems.spark").\
option("url", "IRIS://localhost:51773/NEWSAMPLE").option("user", "dev").\
option("password", "123").\
option("dbtable", "DataMining.IrisDataset").load() # load iris dataset

(trainingData, testData) = dataFrame.randomSplit([0.7, 0.3]) # split the data into two sets
assembler = VectorAssembler(inputCols = ["PetalLength", "PetalWidth", "SepalLength", "SepalWidth"],
outputCol="features") # add a new column with features

kmeans = KMeans().setK(3).setSeed(2000) # clustering algorithm that we use

pipeline = Pipeline(stages=[assembler, kmeans]) # First, passed data will run against assembler and after
will run against kmeans.
modelKMeans = pipeline.fit(trainingData) # pass training data

pmmlBuilder = PMMLBuilder(sc, dataFrame, modelKMeans)
pmmlBuilder.buildFile("KMeans.pmml") # create pmml model
```

El código creará un modelo para predecir las especies utilizando PetalLength, PetalWidth, SepalLength, SepalLength y SepalWidth como características. Utilizaremos el formato PMML.

PMML es un formato para intercambiar modelos predictivos basado en XML. Proporciona una manera para que las aplicaciones analíticas describan e intercambien modelos predictivos generados por la minería de datos y los algoritmos de *Machine Learning*. Esto nos permite separar el desarrollo de los modelos, de su ejecución.

En el resultado verás una ruta hacia el modelo PMML.

Cómo cargar un modelo de ML a InterSystems IRIS (PMML)

Published on InterSystems Developer Community (<https://community.intersystems.com>)

```
pmmlBuilder = PMMLBuilder(sc, dataframe, modelKMeans)
pmmlBuilder.buildFile("KMeans.pmml") # create pmml model
```

```
'/home/guardian/Desktop/zeppelin-0.8.0-bin-all/bin/KMeans.pmml'
```

Subir y probar el modelo PMML

Abre **IRIS Portal Gestión** -> **Menu** -> **Manage Web Applications** -> haz clic en tu **namespace** -> habilita **Analytics** -> **Save**.

Menu

Home | About | Help | Logout

System > Security Management > Web Applications

Server: guardian Namespace: %SYS
User: UnknownUser Licensed to: Sales Engineers Instance: SAMPLE

Refresh: ☒ off ☐ on 10 sec

Web applications that are currently defined:

Max rows: 1000 Results: 20 Page: < << 1 >> > of 1

Namespace	Namespace Default	Enabled	Type	Resource	Authentication Methods	
%SYS	No	Yes	CSP	%Development	Unauthenticated	Delete
%SYS	No	Yes	CSP		Unauthenticated	Delete
%SYS	No	Yes	CSP		Unauthenticated	Delete
%SYS	No	Yes	CSP		Unauthenticated	Delete
%SYS	No	Yes	CSP	%Development	Unauthenticated	Delete
%SYS	No	Yes	CSP		Unauthenticated	Delete
%SAMPLE	Yes	Yes	CSP	%Ens_Portal	Unauthenticated	Delete
%SAMPLE	Yes	Yes	CSP	%Ens_Portal	Unauthenticated	Delete
%USER	Yes	Yes	CSP		Unauthenticated	Delete
%SYS	No	Yes	CSP		Unauthenticated	Delete
%SYS	No	Yes	System,CSP		Unauthenticated	-
%SYS	No	Yes	System,CSP	%Development	Unauthenticated	-
/csp/sys	%SYS	Yes	System,CSP		Unauthenticated	-
/csp/sys/exp	%SYS	No	System,CSP	%Development	Unauthenticated	-
/csp/sys/mgr	%SYS	No	System,CSP	%Admin_Manage	Unauthenticated	-
/csp/sys/op	%SYS	No	System,CSP	%Admin_Operate	Unauthenticated	-
/csp/sys/sec	%SYS	No	System,CSP	%Admin_Secure	Unauthenticated	-
/isc/pki	%SYS	No	System,CSP		Unauthenticated	-
/isc/studio/rules	%SYS	No	System,CSP		Unauthenticated	-
/isc/studio/templates	%SYS	No	System,CSP	%Development	Unauthenticated	-

Save

Cancel

Edit definition for web application /csp/newsample:

GeneralApplication RolesMatching Roles

Name

/csp/newsample

Required. (e.g. /csp/appname)

Description

Interoperability Management Portal

Namespace

NEWSAMPLE

Default Application for NEWSAMPLE: /csp/newsample

Enable Application

☒

Enable

☐ REST

Dispatch Class

Required.

☒ CSP/ZEN

☒ Analytics

☒ Inbound Web Services

☐ Prevent login CSRF attack

Ahora, ve a **Analytics -> Tools -> PMML Model Tester**

Menu

Home | About | Help | Logout

Welcome, UnknownUser

Server: guardian
User: UnknownUser

Namespace: NEWSAMPLE Switch
Licensed to: Sales Engineers
Instance: SAMPLE

View:

Home

Analytics

Interoperability

System Operation

System Explorer

System Administration

Architect

Analyzer

User Portal

Text Analytics »

Tools »

Admin »

MDX Query Tool

Term List Manager

Listing Group Manager

Quality Measures

Worksheet Builder

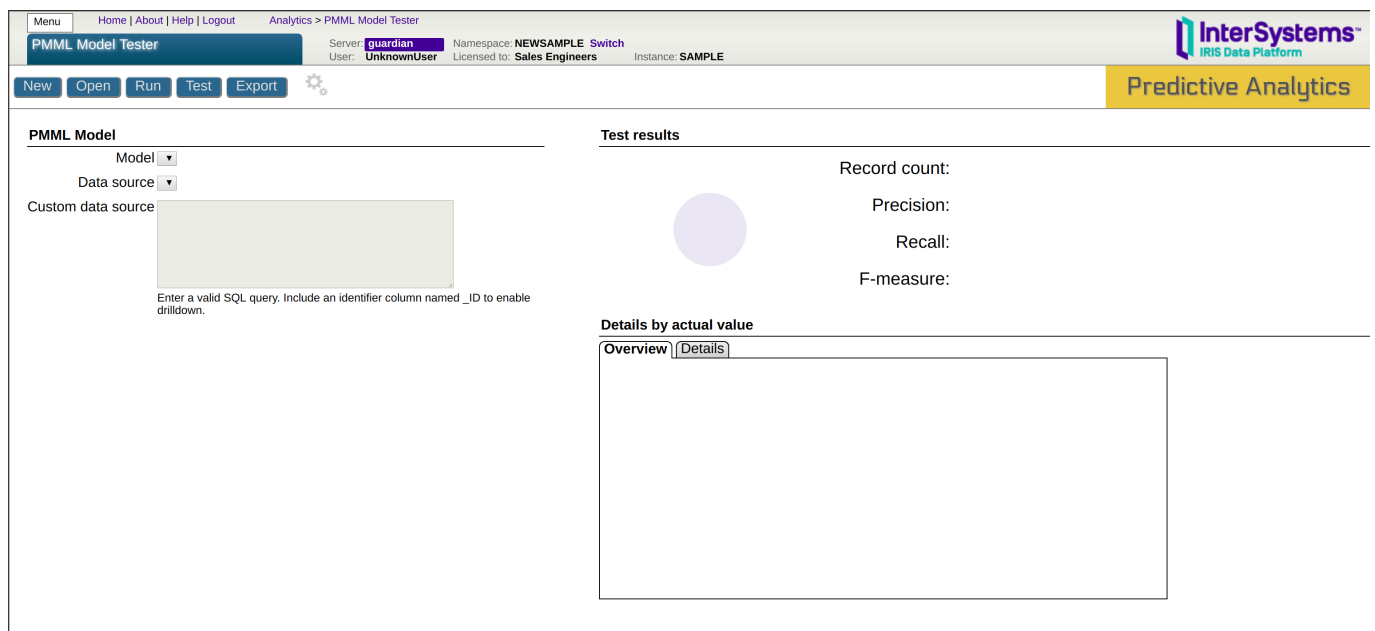
Model Browser

PMML Model Tester

Cómo cargar un modelo de ML a InterSystems IRIS (PMML)

Published on InterSystems Developer Community (<https://community.intersystems.com>)

Deberías ver algo como la imagen que se muestra a continuación:

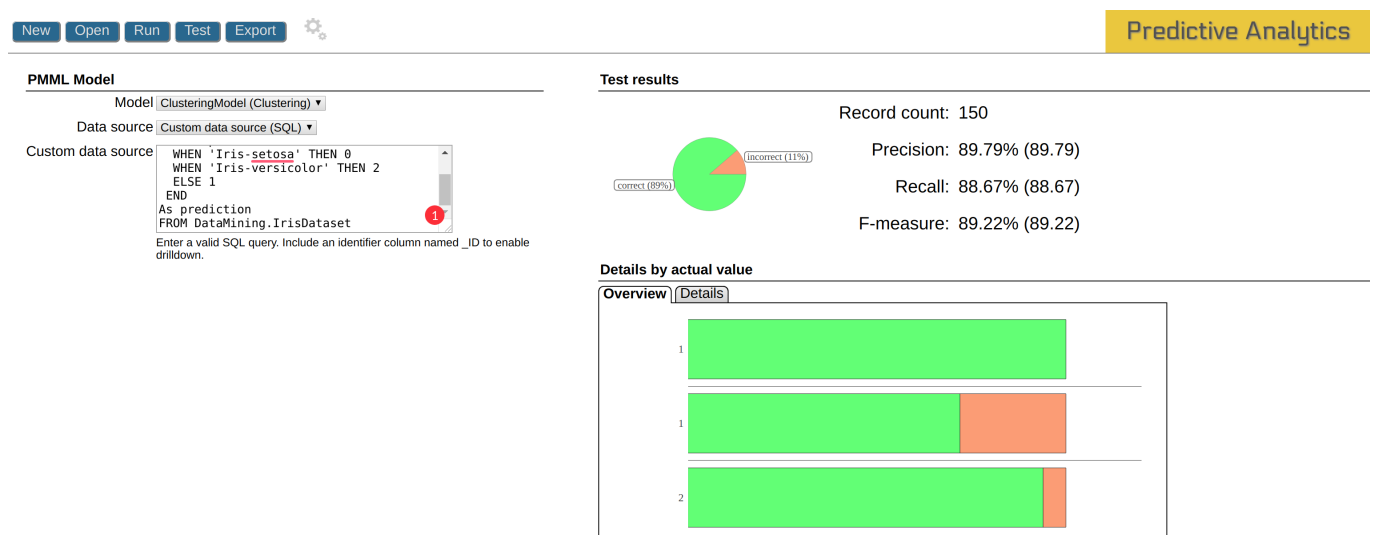


Haz clic en **New**, escribe un nombre para la clase, sube el archivo PMML (la ruta estaba en el resultado), y haz clic en **Import**. Pega la siguiente consulta SQL en **Custom data source**:

```
SELECT PetalLength, PetalWidth, SepalLength, SepalWidth, Species,  
CASE Species  
  WHEN 'Iris-setosa' THEN 0  
  WHEN 'Iris-versicolor' THEN 2  
  ELSE 1  
END  
As prediction  
FROM DataMining.IrisDataset
```

Usamos CASE aquí porque el agrupamiento KMedias devuelve conjuntos de números (0, 1, 2) y si no reemplazamos las especies por números, las contará de manera incorrecta.

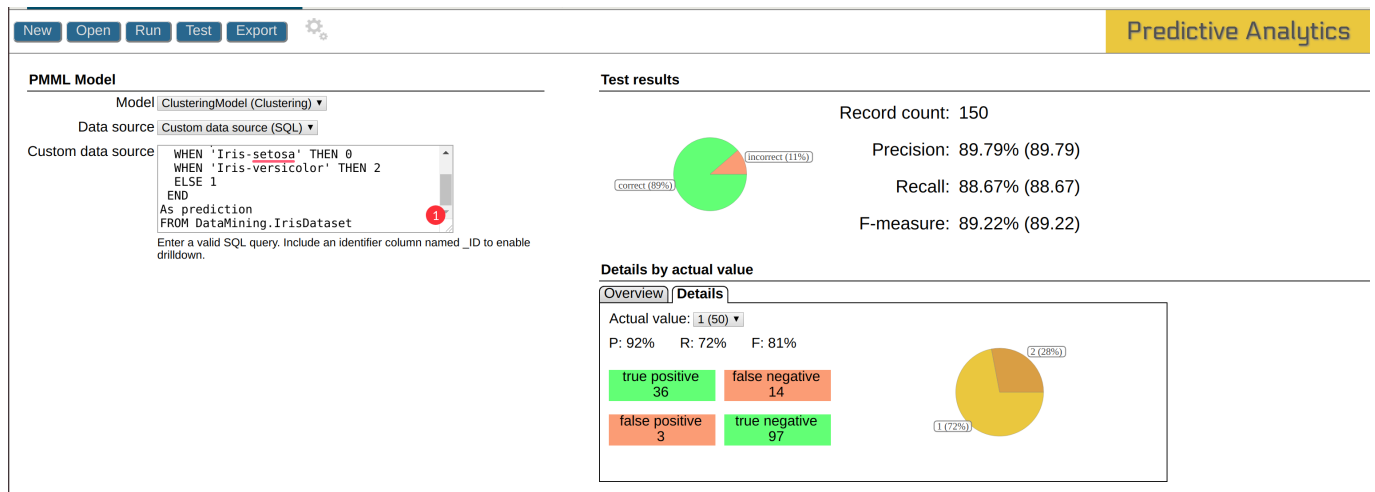
Mi resultado es el siguiente:



Cómo cargar un modelo de ML a InterSystems IRIS (PMML)

Published on InterSystems Developer Community (<https://community.intersystems.com>)

Aquí puedes ver el análisis detallado:



Si deseas obtener más información sobre qué son los **falsos positivos**, los **falsos negativos**..., consulta [Precision and recall](#).

Conclusión

Hemos descubierto que el PMML Model Tester es una herramienta muy útil para probar modelos con cualquier tipo de datos. Proporciona análisis detallados, gráficos y la posibilidad de enviar consultas SQL. Esto te permitirá probar tu modelo sin que necesites de alguna otra herramienta adicional.

Enlaces

[Artículo anterior](#)

[PySpark2PMML](#)

[JPMML](#)

[ML Pipelines](#)

[Documentación sobre Apache Spark documentation](#)

[#Análítica](#) [#API](#) [#Big Data](#) [#Artificial Intelligence \(AI\)](#) [#Machine Learning \(ML\)](#) [#Principiante](#) [#Python](#) [#InterSystems IRIS](#)

URL de
fuentes: <https://es.community.intersystems.com/post/c%C3%B3mo-cargar-un-modelo-de-ml-intersystems-iris-pmml>