



インターシステムズ製品での PMML モデルの使用

Version 2024.1
2024-06-03

インターシステムズ製品での PMML モデルの使用

InterSystems IRIS Data Platform Version 2024.1 2024-06-03

Copyright © 2024 InterSystems Corporation

All rights reserved.

InterSystems®, HealthShare Care Community®, HealthShare Unified Care Record®, IntegratedML®, InterSystems Caché®, InterSystems Ensemble®, InterSystems HealthShare®, InterSystems IRIS®, および TrakCare は、InterSystems Corporation の登録商標です。HealthShare® CMS Solution Pack™ HealthShare® Health Connect Cloud™, InterSystems IRIS for Health™, InterSystems Supply Chain Orchestrator™, および InterSystems TotalView™ For Asset Management は、InterSystems Corporation の商標です。TrakCare は、オーストラリアおよび EU における登録商標です。

ここで使われている他の全てのブランドまたは製品名は、各社および各組織の商標または登録商標です。

このドキュメントは、インターシステムズ社(住所: One Memorial Drive, Cambridge, MA 02142)あるいはその子会社が所有する企業秘密および秘密情報を含んでおり、インターシステムズ社の製品を稼動および維持するためにのみ提供される。この発行物のいかなる部分も他の目的のために使用してはならない。また、インターシステムズ社の書面による事前の同意がない限り、本発行物を、いかなる形式、いかなる手段で、その全てまたは一部を、再発行、複製、開示、送付、検索可能なシステムへの保存、あるいは人またはコンピュータ言語への翻訳はしてはならない。

かかるプログラムと関連ドキュメントについて書かれているインターシステムズ社の標準ライセンス契約に記載されている範囲を除き、ここに記載された本ドキュメントとソフトウェアプログラムの複製、使用、廃棄は禁じられている。インターシステムズ社は、ソフトウェアライセンス契約に記載されている事項以外にかかるソフトウェアプログラムに関する説明と保証をするものではない。さらに、かかるソフトウェアに関する、あるいはかかるソフトウェアの使用から起こるいかなる損失、損害に対するインターシステムズ社の責任は、ソフトウェアライセンス契約にある事項に制限される。

前述は、そのコンピュータソフトウェアの使用およびそれによって起こるインターシステムズ社の責任の範囲、制限に関する一般的な概略である。完全な参照情報は、インターシステムズ社の標準ライセンス契約に記載され、そのコピーは要望によって入手することができる。

インターシステムズ社は、本ドキュメントにある誤りに対する責任を放棄する。また、インターシステムズ社は、独自の裁量にて事前通知なしに、本ドキュメントに記載された製品および実行に対する代替と変更を行う権利を有する。

インターシステムズ社の製品に関するサポートやご質問は、以下にお問い合わせください:

InterSystems Worldwide Response Center (WRC)

Tel: +1-617-621-0700

Tel: +44 (0) 844 854 2917

Email: support@InterSystems.com

目次

インターシステムズ製品での PMML モデルの使用.....	1
1 背景	1
2 InterSystems IRIS での PMML のサポート	1
2.1 サポートされているモデル	1
2.2 あやめサンプル	2
3 PMML モデルを含むクラスを作成	2
3.1 生成されたクラス	3
4 PMML モデルを実行するためのテスト・ページ	3
4.1 サンプル・モデル・テスト・ページ	3
4.2 単一入力レコードのテスト・ページ	5
5 プログラムによる PMML モデルの実行	7
5.1 単一入力レコードを持つモデルの実行	7
5.2 入力レコードのバッチを持つモデルの実行	7
6 InterSystems IRIS ビジネス・インテリジェンスで PMML を使用するためのオプション	7
6.1 ピボット・テーブルからのモデルの呼び出し	8
6.2 Analytics リストに PMML 予測を含める	9
6.3 Analytics キューブへのバッチ結果のエクスポート	9
7 関連項目	9

インターシステムズ製品での PMML モデルの使用

このページでは、InterSystems IRIS® データ・プラットフォームによる PMML (Predictive Modelling Markup Language) の実行時サポートの使用方法について説明します。

1 背景

PMML (Predictive Modelling Markup Language) は、分析モデルを表すための XML ベースの標準です。アプリケーションが統計およびデータ・マイニング・モデルを簡単に再使用および共有できるように定義するための方法を提供します。モデルを生成するために使用される分析ツール (PMML:R、KNIME、SAS、および SPSS などのツール) は InterSystems IRIS または プロダクション環境で使用されるツールとはアーキテクチャが大きく異なるため、この標準が特に役立ちます。

一般的なシナリオでは、データ・サイエンティストは分析ツールを使用して大量の履歴データに基づいてデータ・マイニング・モデルを生成します。このモデルはその後 PMML にエクスポートされます。このモデルは、実行時環境に配置して、受信状態で実行し、モデルのターゲット・メトリックの値を予測することができます。

詳細は、<http://www.dmg.org/> を参照してください。

2 InterSystems IRIS での PMML のサポート

InterSystems IRIS は以下のように PMML 4.1、4.2、4.3、4.4、および 4.4.1 の実行時サポートを提供します。

- PMML モデルが含まれる `%DeepSee.PMML.Definition` のサブクラスである PMML モデル・クラスを定義できます。そのクラスがコンパイルされるときに、システムはその中に記述されているモデルを実行するために必要なコードを生成します。
- InterSystems IRIS は、指定したデータ入力に基づいて、モデルを実行するための API を提供します。
- InterSystems IRIS は、API を使用するサンプルのテスト・ページを提供します。

2.1 サポートされているモデル

InterSystems IRIS は PMML 4.1、4.2、4.3、4.4、4.4.1、および以下の PMML モデルをサポートします。

- Naive Bayes モデル (<https://dmg.org/pmml/v4-4-1/NaiveBayes.html>)
- 回帰モデル (<https://dmg.org/pmml/v4-4-1/Regression.html>)
- プロビット・リンク関数およびコントラスト行列を除く一般回帰モデル (<https://dmg.org/pmml/v4-4-1/GeneralRegression.html>)
- 入れ子モデルを除くツリー・モデル (<https://dmg.org/pmml/v4-4-1/TreeModel.html>)
- 基本的なサポートであるテキスト・モデル (<https://dmg.org/pmml/v4-4-1/Text.html>)

- ・ サポート・ベクター・マシン (SVM) モデル (<https://dmg.org/pmml/v4-4-1/SupportVectorMachine.html>)
- ・ クラスタリング・モデル (<https://dmg.org/pmml/v4-4-1/ClusteringModel.html>)
- ・ RuleSet モデル (<https://dmg.org/pmml/v4-4-1/RuleSet.html>)
- ・ ニューラル・ネットワーク・モデル (<https://dmg.org/pmml/v4-4-1/NeuralNetwork.html>)

InterSystems IRIS は <MiningModel> 要素もサポートします。これは、“モデルのセグメンテーション” (予測のバランスを向上させるために複数のモデルの出力を結合するプロセス) を提供します。<https://dmg.org/pmml/v4-4-1/MultipleModels.html> を参照してください。InterSystems IRIS は“モデル構造”の手法をサポートしません。この手法は非推奨となっています。

2.2 あやめサンプル

このページでは Samples-Data-Mining サンプル (<https://github.com/interystems/Samples-Data-Mining>) を使用します。サンプルは、専用のネームスペース (SAMPLES など) を作成して、そのネームスペースにロードすることをお勧めします。一般的な手順は、“[InterSystems IRIS で使用するサンプルのダウンロード](#)”を参照してください。

このサンプルには、予測分析に使用される既知のサンプルである Iris データ・セットのコピーが含まれます。Iris データ・セットは、異なる 3 種のあやめ約 50 本における花卉およびがく片測定の測定値を提供します。これらの測定値は、あやめの種を強く予測します。

サンプルの設定後、DataMining.PMML.Iris クラス内で PMML モデルを使用できます。このクラスは、以下のモデルを含む PMML 定義を含みます。

- ・ 花卉およびがく片測定値に基づいて、あやめの種を予測するツリー・モデル
- ・ 花卉幅、花卉測定値および種に基づいて、がく片の長さを予測する一般回帰モデル

3 PMML モデルを含むクラスの作成

PMML モデルを含むクラスの作成する手順は以下のとおりです。

1. [PMML モデル・テスト・ページ](#)にアクセスします。
2. [新規作成] をクリックします。
3. [クラス名] で、完全修飾クラス名を入力します。
4. [PMML ファイル] で、[参照] をクリックして、PMML ファイルを選択します。
5. [インポート] をクリックします。

または、IDE 使用して、%DeepSee.PMML.Definition のサブクラスを作成します。このクラスでは、PMML という名前の XData ブロックを作成し、その XData ブロックに PMML 定義を貼り付けます。この XData ブロックに対して、以下のよう XMLNamespace キーワードを設定します。

```
XMLNamespace = "http://www.intersystems.com/deepsee/pmml"
```

使用例は、[サンプル・クラス DataMining.PMML.Iris](#) を参照してください。以下は、部分的な抽出です。

```
Class DataMining.PMML.Iris Extends %DeepSee.PMML.Definition
{
XData PMML [ XMLNamespace = "http://www.intersystems.com/deepsee/pmml" ]
{
<PMML version="4.1">
<Header>
```

```

<Timestamp>03/07/2013 11:54:41</Timestamp>
</Header>
<DataDictionary numberOfFields="5">
  <DataField name="PetalLength" displayName="PetalLength" optype="continuous" dataType="double" />
  <DataField name="PetalWidth" displayName="PetalWidth" optype="continuous" dataType="double" />
  <DataField name="SepalLength" displayName="SepalLength" optype="continuous" dataType="double" />
  <DataField name="SepalWidth" displayName="SepalWidth" optype="continuous" dataType="double" />
  <DataField name="Species" displayName="Species" optype="categorical" dataType="string">
    <Value value="IRIS-SETOSA" property="valid" />
    <Value value="IRIS-VERSICOLOR" property="valid" />
    <Value value="IRIS-VIRGINICA" property="valid" />
  </DataField>
</DataDictionary>
...

```

このサンプルの設定および使用についての詳細は、“[あやめサンプル](#)”を参照してください。

可能な場合は、入力フィールド名が 50 文字未満になるようにしてください。これより長いフィールド名もサポートされますが、長いフィールド名に値を指定するとすぐに混乱が生じます。また、生成された SQL ストアド・プロシージャなどの特定のエンドポイントでは、制限された長さの引数のみがサポートされ、長い引数は切り捨てられて一意の整数が追加されます。

3.1 生成されたクラス

PMML クラス (PackageName.ClassName) をコンパイルすると、システムは以下のクラスを生成します。

- PackageName.ClassName.Data は、モデルを実行するために必要なデータを表します。このクラスは、PMML 定義の <DataDictionary> 要素に対応しています。
- PackageName.ClassName.ModelName は、モデルを表します。
このクラスはモデルを実行するためのメソッドを提供します。
- PackageName.ClassName.ModelName.Input は、指定されたモデルの入力を表します。
- PackageName.ClassName.ModelName.Output は、指定されたモデルの出力を表します。

InterSystems IRIS は、これらのクラスを使用してモデルを実行します。

4 PMML モデルを実行するためのテスト・ページ

InterSystems IRIS は、[レコードのバッチ](#)または[単一入力レコード](#)に対する PMML モデルを実行するために使用できるテスト・ページを提供します。これらのページにアクセスする手順は以下のとおりです。

1. 管理ポータルを開きます。
2. 適切なネームスペースに切り替えます。
3. [Analytics]→[ツール]→[PMML モデル・テスター] をクリックします。

4.1 サンプル・モデル・テスト・ページ

次のようなページが表示されます (部分的に表示)。

**PMML Model**Model Data source

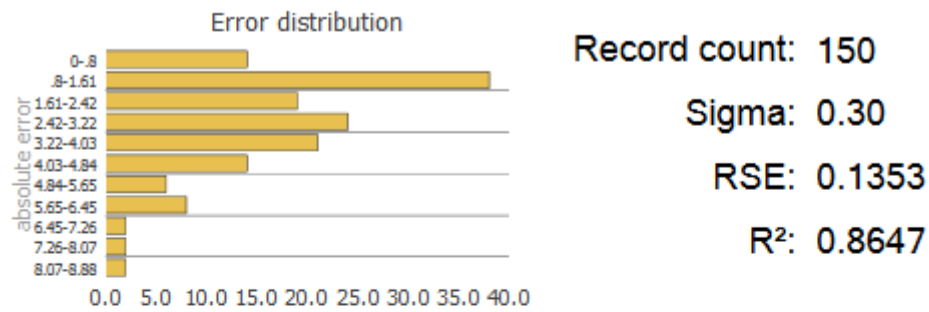
Custom data source

このページを使用する手順は以下のとおりです。

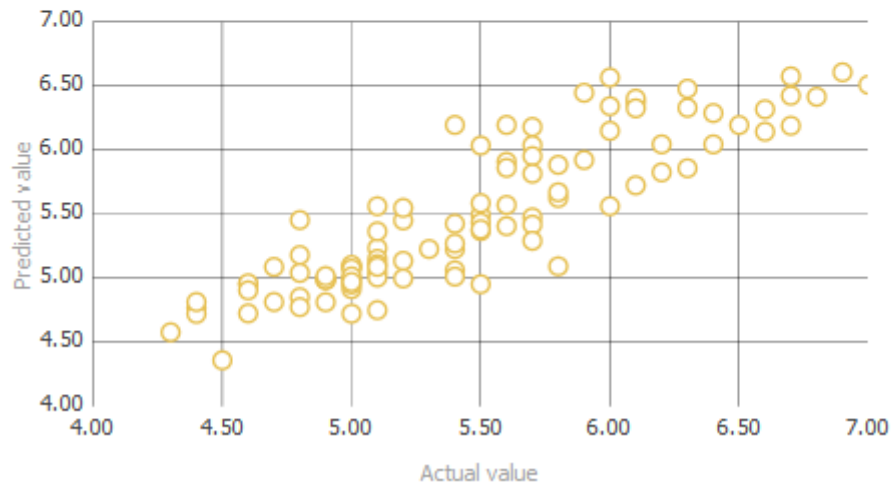
1. **【開く】** をクリックし、モデル・クラスを選択して、**【OK】** をクリックします。
2. **【モデル】** ドロップダウン・リストからモデルをクリックします。
3. **【データソース】** ドロップダウン・リストからオプションをクリックします。オプションには以下のものがあります。
 - ・ **【分析データセット】** – モデルが開発されたソース・データを選択します。
 - ・ **【カスタム・データ・ソース (SQL)】** – レコードのセットを提供する SQL クエリを入力できるようにします。
4. **【カスタム・データ・ソース (SQL)】** を選択した場合、SQL SELECT クエリを **【カスタム・データ・ソース】** に入力します。
5. **【実行】** をクリックします。

InterSystems IRIS はレコードの繰り返し処理を行い、結果の概要を表示します。詳細はモデルによって異なります。以下に例を示します。

Test results



Value distribution



4.2 単一入力レコードのテスト・ページ

単一入力レコードでモデルをテストすることも可能です。そのためには、[テスト]を押します。以下のようなダイアログ・ボックスが表示されます。

PMML Model test page

Data object

Species

Petal Length

Petal Width

Sepal Length

Sepal Width

Model

Predicted value:

Trace info:

[データ・オブジェクト] にリストされているフィールドは、PMML 定義のデータ・フィールドに対応します。

このページを使用するには、[モデル] ドロップダウン・リストからモデルを選択します。このモデルは、どのフィールドが入力フィールドであるか出力フィールドであるかを決定します。入力フィールドに値を入力します。すべての入力値を入力すると、ページに、指定されたモデルの出力フィールドの予測値が表示されます。以下はその例です。

PMML Model test page

Data object

Species

Petal Length

Petal Width

Sepal Length

Sepal Width

Model

Predicted value:

IRIS-VERSICOLOR

predicted value: IRIS-VERSICOLOR

probability: 1

Trace info:

Rank	Info
1	Tree node 1: (SepalWidth greaterThan "3.1450000000000004618")
2	Tree node 3: (PetalWidth greaterThan "1.3349816849816849018")
3	Tree node 7: (SepalLength lessThan "6.5753846153846158273")
4	Tree node 13: (PetalLength lessThan "5.0525000000000002131")

5 プログラムによる PMML モデルの実行

InterSystems IRIS では、PMML モデルを実行するために使用できる API も提供します。

5.1 単一入力レコードを持つモデルの実行

単一レコードの予測モデルを実行するには、以下の手順に従います。

1. 生成されたクラス `PackageName.ClassName.ModelName` のインスタンスを作成します。このクラスは、モデルを実行するために使用できるメソッドを定義します。
2. モデルの入力を作成します。これをより簡単に行うには、以下の例に示すように、入力値を含む多次元配列を定義します。

```
set data("PetalLength") = 123
set data("PetalWidth") = 456 // etc
```

または、生成されたクラス `PackageName.ClassName.Data` のインスタンスを作成し、そのプロパティを設定します。このインスタンスの目的は、入力値を含めることです。

3. モデル・インスタンスの `%ExecuteModel()` メソッドを呼び出します。

```
Method %ExecuteModel(ByRef pData,
                    Output pOutput As %DeepSee.PMML.ModelOutput) as %Status
```

`pData` では、ステップ 2 で作成した多次元配列またはデータ・オブジェクトを使用します。

以下に例を示します。

```
do model.%ExecuteModel(.data, .output)
```

このメソッドは、出力引数として、モデルの出力を含む `%DeepSee.PMML.ModelOutput` のインスタンスを返します。具体的には、これは指定されたモデルの生成されたクラス `PackageName.ClassName.ModelName.Output` のインスタンスです。

4. 出力の詳細を表示するには、`ZWRITE` を使用します。`pOutput` オブジェクトは、モデル定義の `Output` 要素内の各 `OutputField` に対して 1 つのプロパティを含みます。〈`Output`〉要素がない場合は、`pOutput` は、予測された `MiningField` 要素の後に指定された単一フィールドを含みます。

5.2 入力レコードのバッチを持つモデルの実行

入力レコードのバッチを持つ予測モデルを実行するには、`%DeepSee.PMML.Utils` の `%RunModel()`、`%RunModelFromResultSet`、または `%RunModelFromSQL` を使用します。これらのメソッドは、予測結果を `%DeepSee_PMML_Utils.TempResult` テーブルに保存します。

6 InterSystems IRIS ビジネス・インテリジェンスで PMML を使用するためのオプション

このセクションでは、InterSystems IRIS ビジネス・インテリジェンスで PMML を使用するためのオプションについて説明します。

- ・ [Analytics ピボット・テーブルからのモデルの呼び出し](#)
- ・ [詳細リストに予測を含める](#)
- ・ [キューブへの結果のエクスポート](#)

背景情報は、“InterSystems Business Intelligence のモデルの定義”を参照してください。

6.1 ピボット・テーブルからのモデルの呼び出し

PMML モデルをピボット・テーブル内から呼び出すことができます。そのためには、%KPI 関数を使用する計算メンバを定義して %DeepSee.Plugin.PMML プラグインを呼び出します。以下の構文を使用します。

```
%KPI("%DeepSee.PMML",fieldName,series,"PMML",modelName,parmName1,parmValue1,parmName2,parmValue2,"%CONTEXT")
```

以下はその説明です。

- ・ fieldName は、InterSystems IRIS PMML モデル・クラスの実出力フィールドの引用符付き名です。
- ・ series は、プラグイン内の系列 (行) のオプションの番号です。1 を指定するか、この引数を省略します。
- ・ modelName は、InterSystems IRIS PMML モデル・クラスの引用符付き名です。
- ・ parmName1、parmName2 (以降同様) は、%DeepSee.Plugin.PMML プラグインのパラメータのオプションの引用符付きの名前です。パラメータ名では大文字と小文字が区別されることに注意してください。%DeepSee.Plugin.PMML プラグインには、以下のパラメータを指定します。
 - － "%cube" — この KPI が実行されるキューブを指定します。
 - － "modelName" — 実行するモデルの名前を指定します。指定する場合、これは指定されたモデル・クラスのモデルである必要があります。空白のままにすると、このクラスの最初のモデルが実行されます。
 - － "aggregate" — さまざまなファクトを越えて予測される値を集計する方法を指定します。使用可能な値は、"average" (既定値)、"sum"、"max"、"min"、"maxFreqValue"、"maxProbValue"、"minFreqValue" および "minProbValue" です。

各出力フィールドの集計がすべて意味を持つとは限りません。

- － "NullValues" — 結果を集計するときに NULL 予測値を含めるかどうかを指定します。使用可能な値は、"ignore" (既定) または "count" です。

パラメータをリストする順序は、結果には影響しません。

16 個までのパラメータとその値を指定できます。

- ・ parmValue1、parmValue2 (以降同様) は、名前付きフィルタの対応値です。

特殊な %CONTEXT パラメータを使用すると、使用しない場合は無視されるクエリのコンテキストがプラグインで考慮されます。詳細は "%KPI" を参照してください。

例えば、以下の構文を使用して、単一モデルのみが含まれている Test.MyModel という名前の PMML モデル・クラスの実出力フィールド MyField の平均値を取得します。

```
%KPI("%DeepSee.PMML", "MyField", , "PMML", "Test.MyModel", "aggregate", "average", "%CONTEXT")
```

6.2 Analytics リストに PMML 予測を含める

レコード・レベル予測を詳細リストに含めるには、リスト・クエリに \$\$\$PMML トークンを使用します。このトークンは、PMML 定義クラス名およびモデル名をプライマリ・パラメータとして使用します。オプションの 3 番目の引数として、クエリに含める予測機能の名前を渡すことができます (この引数は既定で "predictedValue" に設定されています)。

以下は、このトークンを使用するリスト・クエリの定義を示しています。

```
UserID, TotalWagered, PercentLost "Lost %" , $$$PMML[MyPMML.Poker,PercentLost] "Predicted Loss %"
```

6.3 Analytics キューブへのバッチ結果のエクスポート

入力レコードのバッチを持つ予測モデルを実行した後に、結果をキューブにエクスポートすることができます。このオプションにより、結果を別の方法で視覚化することができます。キューブには、ActualValue と PredictedValue の 2 つのレベルがあります。

結果をキューブにエクスポートするには、[PMML テスト・ページ](#)を使用して [エクスポート] をクリックします。InterSystems IRIS により、以下の情報が求められます。

- ・ [エクスポート] – [データのみ] または [クラスとデータ]を選択します。
- ・ [結果クラス名] – 結果が書き込まれる永続クラスの名前を指定します。これは、キューブのソース・クラスとして使用されます。
- ・ [ソース・クラスへのリンク] – ソース・レコードを含むクラスを指定します。結果クラスには、このクラスを指す **Record** という名前のプロパティが含まれています。
- ・ [データを削除] – エクスポートを実行する前に結果クラス ([結果クラス名]) を空にする場合にこれを選択します。新しくエクスポートするデータを結果クラス・テーブルの末尾に追加する場合にはこれをクリアします。
- ・ [キューブ名] – キューブの論理名を指定します。
- ・ [キューブ・クラス名] – キューブのクラス名を指定します。
- ・ [クラスを上書き] – このエクスポートを以前に実行済みで、新しいデータと定義でそれらのクラスを上書きする場合にこれを選択します。

指定されたキューブを作成できる [キューブの作成] ダイアログ・ボックスが表示されます。[ビルド] または [キャンセル] をクリックします。後からアーキテクトを介してこのキューブにアクセスし、そこから作成することもできます。

キューブを作成した後に、アナライザを使用してキューブを調べます。以下に例を示します。ActualValue レベルが行として、PredictedValue レベルが列として使用されます。

Actual Value	IRIS-SETOSA	IRIS-VERSICOLOR	IRIS-VIRGINICA
IRIS-SETOSA	50		
IRIS-VERSICOLOR	3	45	2
IRIS-VIRGINICA		6	44

7 関連項目

- ・ <http://www.dmg.org/>
- ・ <https://github.com/interSystems/Samples-Data-Mining>

- ・ [InterSystems IRIS で使用するサンプルのダウンロード](#)
- ・ [InterSystems Business Intelligence のモデルの定義](#)