1 Topic

Deployment of predictive models using the PMML standard with PDI-CE (Pentaho Data Intergration Community Edition – Kettle).

Model deployment is a crucial task of the data mining process. In the supervised learning, it can be the applying of the predictive model on new unlabeled cases. We have already described this task for various tools (e.g. <u>Tanagra</u>, <u>Sipina</u>, <u>Spad</u>, <u>R</u>). They have as common feature the use of the same tool for the model construction and the model deployment.

In this tutorial, we describe a process where we do not use the same tool for the model construction and the model deployment. This is only possible if (1) the model is described in a standard format, (2) the tool which used for the deployment can handle both the database with unlabeled instances and the model. Here, we use the PMML standard description for the sharing of the model, and the PDI-CE for the applying of the model on the unseen cases.

The **Predictive Model Markup Language** (PMML) is an XML-based markup language to provide a way for applications to define models related to predictive analytics and data mining and to share those models between PMML-compliant applications. Proprietary issues and incompatibilities are no longer a barrier to the exchange of models between applications¹. PMML is promoted by the Data Mining Group (DMG) which is an independent vendor led consortium that develops data mining standards². **Pentaho Data Integration**, codenamed Kettle, consists of a core data integration (ETL) engine, and GUI applications that allow the user to define data integration jobs and transformations³. We have already described the Community version of this tool (PDI-CE)⁴. We have observed that it is a very convenient tool for the handling of dataset.

In this tutorial, we create a decision tree with various tools such as SIPINA, KNIME or RAPIDMINER; we export the model in the PMML format; then, we use PDI-CE for applying the model on a data file containing unlabeled instances. We see that the use of the PMML standard enhances dramatically the powerful of both the data mining tool and the ETL tool.

In addition, we will describe other solutions for deployment in this tutorial. We will see that Knime has its own PMML reader. It is able to apply a model on unlabeled datasets, whatever the tool used for the construction of the model. The key is that the PMML standard is respected. In this sense, Knime can be substituted to PDI-CE. Another possible solution, Weka, which is included into the Pentaho Community Edition suite, can export the model in a proprietary format that PDI-CE can handle.

2 Dataset

We use the <u>heart</u>⁵ dataset in this tutorial. We want to predict the occurrence of the heart disease from the patient characteristics. "heart-train.arff" is to the training set used for the construction of

¹ <u>http://en.wikipedia.org/wiki/Predictive_Model_Markup_Language</u>

² <u>http://www.dmg.org/</u>

³ <u>http://en.wikipedia.org/wiki/Pentaho</u>

⁴ <u>http://data-mining-tutorials.blogspot.fr/2012/04/pentaho-data-integration-kettle.html</u>

⁵ <u>http://archive.ics.uci.edu/ml/datasets/Heart+Disease</u>

the model. "heart-unlabeled.txt" it is the data file containing the unlabeled instances. The goal of the process is to add the assigned class membership to this dataset using the predictive model.

3 Deployment with PDI-CE

3.1 Installing PDI-CE and the "WekaScoring" plugin

We have described the installation and the use of PDI-CE in a previous tutorial⁶. For the model deployment, we must install the "WekaScoring" plugin. It enables to handle models produced by Weka in a proprietary format, but more generally it can also handle models in the PMML format whatever the tool used for the model creation.

First, we download the plugin from the Pentaho website. We must check carefully the versions.



We unpack the file into the PDI-CE directory. We should have a configuration similar to below.

Programmes > pdi-ce-4.0.1-stable > data-integration)	▶ pli	ugins 🕨 steps 🕨 WekaScorir		Rechercher		
Fichier Edition Affichage Outils ? ① Organiser				.,	0	
Liens favoris Documents Musique Autres >> Dossiers Spoon Signal Steps DummyPlugin SigCsvinput ShapeFileReader3	*	Nom distrib docs libext libswt src tests WekaScoringDeploy build.xml LICENSE.txt LICENSE.txt LICENSE.xp3.txt LICENSE-XStream.txt README.txt	Date de modificati 04/09/2010 07:31 04/09/2010 07:31 04/09/2010 07:32 04/09/2010 07:31 04/09/2010 07:31 04/09/2010 07:31 04/09/2010 07:32 22/09/2009 10:05 19/11/2007 16:13 17/01/2008 14:41 17/01/2008 14:41 11/08/2010 14:35	Type Dossier de fichiers Dossier de fichiers Dossier de fichiers Dossier de fichiers Dossier de fichiers Dossier de fichiers Dossier de fichiers Document XML Document texte Document texte Document texte Document texte	Taille 4 Ko 35 Ko 3 Ko 2 Ko 1 Ko	
12 éléments (espace libre : 334 Go)	Ŧ			🜉 Ordinateur		

Then, we must copy the "weka.jar" file from the Weka distribution⁷ (version 3.7.2 for this tutorial) into the sub-directory "WekaScoringDeploy".

⁶ <u>http://data-mining-tutorials.blogspot.fr/2012/04/pentaho-data-integration-kettle.html</u>

⁷ <u>http://www.cs.waikato.ac.nz/ml/weka/</u>

C→ ↓ ≪ data-integration → plugins → steps → WekaScor	ring ▶ Wek	aScoringDeploy		hercher	<u>× 0 م</u>
Fichier Edition Affichage Outils ?					
🍓 Organiser 👻 🏢 Affichages 👻 🚯 Graver	_				0
Liens favoris		Nom ***	Date de modificati	Type Document VMI	Taille
 Documents Images Musique Autres >> 		scoring.jar weka.jar WS.png Spg min-1.1.3,4.0.jar	11/08/2010 14:31 11/08/2010 14:31 27/07/2010 05:22 11/08/2010 14:31 11/08/2010 14:31	Executable Jar File Executable Jar File Image PNG Executable Jar File	44 Ko 5 902 Ko 2 Ko 25 Ko
Dossiers	~	📧 xstream-1.2.2.jar	11/08/2010 14:31	Executable Jar File	365 Ko
 ↓ plugins ↓ databases ↓ jobentries ↓ repositories ↓ steps ↓ DummyPlugin ↓ StapeFileReader3 ↓ WekaScoring ↓ docs ↓ libswt ↓ src ↓ tests ↓ WekaScoringDeploy 	A III				
6 éléments (espace libre : 333 Ge) — 🦯				👰 Ordinateu	r

Last, to check the installation of the plugin, we launch PDI-CE. The WekaScoring icon must be visible into the TRANSFORM branch of the creation palette.

🚫 Pentaho Data Integration - Tra	ansformation 1
Fichier Edition Vue Action	Tools Aide
1) 💪 🗉 🖥 🖥	Perspective: 👿 Data Integration
Palette de création »1	👊 Bienvenue 🛛 💥 Transformation 1 🛛
Fonctions 🛛 🕒	🕨 🏢 🖿 📐 🎋 🖻 🔛 🗞 🦾 🔂 🔚 100% 🗸
Transform	·
🌈 Weka Scoring	
Extraction	
Alimentation	
🛅 Transformation	
📄 Divers	
📄 Contrôle de flux	
💼 Exécution de scripts	
💼 Recherche	
📄 Jointure lignes	
📄 Entrepôt de données	
📄 Validation	
💼 Statistiques	
🗀 Lien avec Tâche	
💼 Sous-tranformation	
💼 En ligne	
Expérimental	
📄 Étapes obselètes	
🗀 Chargement en bloc	
📄 Historique étapes	< >

To perform the deployment with PDI-CE, we must learn a decision tree and save it in the PMML format. We use the SIPINA tool for that.

3.2 Learning and storing a model in the PMML format with SIPINA

We use SIPINA 3.4 or later version for this task.

Sipina Research Version 3.4		
File Edit Data Statistics Induct	tion method Analysis View Window Help	
🕂 👿 👹 🖻		
Attribute selection	St. Learning set editor	
	Var_1	Ξ
Learning method MethodName=Improved ChAID (Tsc A MethodClassName=TArbreDecisionI HdI=8 Merge=0.05 Split=0.001 TypeBonferroni=1 ValueBonferroni=1 Sampling=0		
Examples selection 1 examples selected 0 examples idle	< III	Ŧ
, Improved ChAID (Tschuprow Goodnes	ss of Split)	11.

If we have not the right version, we can download the setup file on the Sipina website (<u>http://eric.univ-lyon2.fr/~ricco/sipina.html</u>).

Importing the data file. After we launch SIPINA, we click on the FILE / OPEN to import the dataset. We pick the "heart-train.arff" data file.

💦 Si	pina Research Version 3.4										
File	Edit Data Statistics Induction n	method Analysi	s View Windo	w Help)						
	New										
	Open	Learning cet edit	tor								
	Save	S Ouvrir									
	Save as	Regarder dans :	🔐 pmml			🔹 🔶 🔁	▥▾				
	Export	Ca	Nom Date	e de m	Туре	Taille	Mots-clés				
	Subsample management	Emplacements	📄 heart-train.arff								
	Exit	lecents									
J		Bureau									
Lear	ning method										
Meth	odClassName=TArbreDecisionI	Maison									
Merg	o je=0.05										
Type	=0.001 Bonferroni=1	Ordinatour									
Valu Sam	eBonferroni=1 pling=0	Ordinated									
Exa	mples selection										
1 exa	amples selected	Réseau	Nom du fichier :	heart-train	n.arff		<u> </u>	Ouvrir			
			Types de fichiers :	WEKA Fi	le Format (*.arff)	•	Annuler			
Impro	oved ChAID (Tschuprow Goodness of S	Split)									

The dataset contains 13 variables and 270 instances.

🔉 Sipina Research Versi	on 3.4 - [Lea	rning set ed	litor]		1				
🔉 File Edit Data	Statistics In	duction me	thod Ana	ysis View	Window	Help			- 8 ×
🐺 🐺 📑 🖻									
<u> </u>		age	sexe	type_douleu	pression	cholester	sucre	electro	tau: 🔺
Attribute selection	1	70.00	masculin	D	130.00	322.00	A	С	105
	2	67.00	feminin	С	115.00	564.00	A	С	160
	3	57.00	masculin	В	124.00	261.00	A	A	141
	4	64.00	masculin	D	128.00	263.00	A	A	105
	5	74.00	feminin	в	120.00	269.00	A	С	121
	6	65.00	masculin	D	120.00	177.00	A	A	140
	7	56.00	masculin	С	130.00	256.00	в	С	142
	8	59.00	masculin	D	110.00	239.00	A	С	142
	9	60.00	masculin	D	140.00	293.00	A	С	170
Learning method	10	63.00	feminin	D	150.00	407.00	A	С	154
MethodName=Improve 🔺	11	59.00	masculin	D	135.00	234.00	A	A	161
MethodClassName=TA	12	53.00	masculin	D	142.00	226.00	A	С	111
Merge=0.05	13	44.00	masculin	С	140.00	235.00	A	С	180
Split=0.001	14	61.00	masculin	A	134.00	234.00	A	A	145
TypeBonferroni=1	15	57.00	feminin	D	128.00	303.00	A	С	159
Sampling=0	16	71.00	feminin	D	112.00	149.00	A	A	125
Examples selection	17	46.00	masculin	D	140.00	311.00	A	A	120
Enamples selection	18	53.00	masculin	D	140.00	203.00	В	С	155 👻
	•		1		1				1
	Editing	NEW.FDM				Attrib	utes : 13	Examples	270
Improved ChAID (Tschup	row Goodne	ss of Split)					Exec.	lime : 15 ms.	

Decision tree learning. We define the role of the variables by clicking on the ANALYSIS / DEFINE CLASS ATTRIBUTE menu. We set COEUR as target variable, the others as input ones.

🔉 Sipina Research Versio	on 3.4 - [Learning	set editor]	ar management				
💦 File Edit Data S	Statistics Induction	on method	Analysis View Window Help	_ 8 ×			
🖰 搔 🖺 🕒			Define class attribute				
<u>×</u>	age	sexe	Select active examples	electro tau:			
Attribute selection	1 70.00) maso	Set weight field	C 105			
	2 67.00) femir	Attribute selection				
	3 57.00) maso		Variables			
	4 64.00) maso					
5 74	5 74.00) femir	Icoeur	sexe			
	6 65.00) maso		type_douleur			
	7 56.00) maso	Ambutes	cholester sucre			
	8 59.00) maso	sexe				
	9 60.00) maso	type_douleur	taux_max			
Learning method	10 63.00) femir	cholester =	angine			
MethodName=Improve 🔺	11 59.00) maso	sucre	pic			
MethodUlassName=1A	12 53.00) maso	taux_max	Vaisseau			
Merge=0.05	13 44.00) maso	angine	,			
Split=0.001	14 61.00) maso					
TypeBonferroni=1	15 57.00) femir					
Sampling=0	16 71.00) femir	C Only discrete				
Examples selection	17 46.00) maso	C Only continuous				
	18 53.00) maso	© Both				
	Editing NEW.F	DM		🗸 OK 🛛 🗶 Annuler			
Improved ChAID (Tschup	row Goodness of S	plit)	Exec. I II	me : 15 ms.			

Then, we click on the ANALYSIS / LEARNING menu to launch the learning process. We obtain a rather simplistic model. Only 3 predictive variables among 12 are selected.

Sipina Research Version 3.4 - [Decision tree]
🕻 Induction method Analysis Tree management View Window Help
Induction method Analysis Tree management View Window Help I Induction method Analysis Tree management View Window Help I Induction method Analysis Tree management View Window Help I Induction method Analysis Tree management View Window Help I Induction method Analysis Tree management View Window Help I Induction method Analysis Tree management View Window Help I Induction method Analysis Tree management View Window Help I Induction method Analysis Tree management View Window Help I Induction method Analysis Tree management View Window Help I Induction Method Induction Ind
examples selected -
proved ChAID (Tschuprow Goodness of Split) Exec.Time : 31 ms

Exporting the tree in the PMML format. We want to export the decision tree in the PMML format. We click on the TREE MANAGEMENT / EXPORT DECISION TREE menu. We set the file name "heart.xml".



Note : Choosing the xml extension seems more appropriate for PDI-CE. In fact, both ".xml" and ".pmml" extensions refer to the PMML file format.

Understanding the PMML format. The "heart.xml" file is subdivided in several parts: (A) the data dictionary; (B) the classification tree, including the "Mining Schema" which describes the role of the variables (class and predictive attributes).

- About the data dictionary (Figure 1), we observe for instance that "sex" is a discrete attribute with two values "masculin" and 'feminin" (in French). On the other hand, we observe that "pression" is a continuous attribute. The minimum (resp. maximum) value "leftMargin" ("rightMargin") is 94 (200). We use the scientific notation.
- Into the mining schema (Figure 2), we observe that "coeur" is the target attribute.
- Last, we have the description of the decision tree (Figure 3). We note the correspondence between the graphical representation of the tree and the PMML description. For each node, we observe the number of instances covered by the node, the rule associated to the node, the number of instances corresponding to each class value.

```
<DataDictionary numberOfFields="13">
<DataField name="age" optype="continuous" dataType="double">
<Interval closure="closedClosed" leftMargin="2.900000000000E+001" rightMargin="7.70000000000E+001"/>
</DataField>
<DataField name="sexe" optype="categorical" dataType="string">
<Value value="masculin"/>
<Value value="feminin"/>
</DataField>
<DataField name="type douleur" optype="categorical" dataType="string">
<Value value="D"/
<Value value="C"/>
<Value value="B"/>
<Value value="A"/>
</DataField>
<DataField name="pression" optype="continuous" dataType="double">
<Interval closure="closedClosed" leftMargin="9.400000000000E+001" rightMargin="2.000000000000E+002"/>
</DataField>
<DataField name="cholester" optype="continuous" dataType="double">
<Interval closure="closedClosed" leftMargin="1.260000000000E+002" rightMargin="5.640000000000E+002"/>
</DataField>
<DataField name="sucre" optype="categorical" dataType="string">
<Value value="A"/>
<Value value="B"/>
</DataField>
<DataField name="electro" optype="categorical" dataType="string">
<Value value="C"/>
<Value value="A"/>
<Value value="B"/>
</DataField>
<DataField name="taux max" optype="continuous" dataType="double">
<Interval closure="closedClosed" leftMargin="7.100000000000E+001" rightMargin="2.020000000000E+002"/>
</DataField>
<DataField name="angine" optype="categorical" dataType="string">
<Value value="non"/
<Value value="oui"/>
</DataField>
<DataField name="depression" optype="continuous" dataType="double">
<Interval closure="closedClosed" leftMargin="0.00000000000000E+000" rightMargin="6.19999980926514E+000"/>
</DataField>
<DataField name="pic" optype="continuous" dataType="double">
<Interval closure="closedClosed" leftMargin="1.000000000000E+000" rightMargin="3.00000000000E+000"/>
</DataField>
<DataField name="vaisseau" optype="categorical" dataType="string">
<Value value="D"/>
<Value value="A"/>
<Value value="B"/>
<Value value="C"/>
</DataField>
<DataField name="coeur" optype="categorical" dataType="string">
<Value value="presence"/>
<Value value="absence"/>
</DataField>
</DataDictionary>
```

Figure 1 - PMML – Data dictionary



Figure 2 - Mining Schema





3.3 Deployment using PDI-CE and the WekaScoring plugin

We want to use this tree to label the instances from the "heart-unlabeled.txt" data file. We start PDI-CE. We create a new transformation project "FICHIER / NOUVEAU / TRANSFORMATION".



Importing the unlabeled instances. We import the data file with unlabeled instance using the "**Extraction depuis fichier CSV**" component. We set the following parameters:



The tabulation character is the column separator. We click on the "Recupérer champs" button to obtain the columns description.

Labelling with the WekaScoring component. We add the **WekaScoring** component into the workflow. We set the model file name "heart.xml" into the Settings dialog.

🚫 Pentaho Data Integration - Transformation 1 (modifiée)									
Fichier Edition Vue Action Tools Aide									
1 🖻 🗉 🖬 🖥	Perspective: 👿 Data Integration								
🔯 Navigateur 🧪 Palette de création	🔓 Bienvenue 🛛 💥 Transformation 1 🛛								
Fonctions	🕨 III 🔳 🖎 🎋 🖻 🌚 🗞 🥾 🕰 🔚 100% 👻								
 Transform Weka Scoring Extraction Alimentation Transformation 	Extraction depuis fichier CSV Weka Scoring								
Divers	Weka Scoring								
Exécution de scripts Recherche Jointure lignes Entrepôt de données Validation Statistiques Lien avec Tâche Sous-tranformation En ligne Expérimental Ca	Step name Weka Scoring odel file Fields mapping Model Load/import model jr_soft_dev_and_comparison\pmml\heart.xml & Parcourir Update model Save updated model File name field ache loaded models in memory								
Chargement en bloc	Output probabilities								
	< ×								

Creating the output file. We want to create the output file containing the values of the predictive variables and the assigned class value in the Excel file format. We add the "Alimentation Fichier Excel" into the workflow. We set the output file name "heart-predicted.xls".

Extraction depuis fichier CSV	Weka Scoring Alimentatio	n fichier Excel	
	Alimentation fichier Excel		
	Nom étape	Alimentation fichier Excel	
	Fichier Contenu Champs	/	,
	Nom fichier	r soft dev and comparison\p	mml\heart-predicted 🕸 Parcourir
	Ne pas créer le fichier au démarrage		
	Extension	xls	\$
	Inclure n°étape dans nom de fichier		
	Inclure date dans nom de fichier		
	Inclure heure dans nom de fichier		
	Spécifier format date		
	Format date		-
		Visualiser Nom de fichier(s)]
	Ajouter fichier au résultat		
		OK Annuler	

Launching the process. We launch the process by clicking on the ACTION / EXECUTER menu (F9).

🚫 Pentaho Data Integration - heart				and the second second	
Fichier Edition Vue Action Tools Aide					
1 🖻 🗉 🔚 🗃				Perspective: 🕃	Data Integration
🔁 Navigateur 🧪 Palette de création	읂 Bier	nvenue 🛛 💥 heart 🛛 📃			
Fonctions		🔲 📐 🎋 🖄 🔡 🔂 🕹 🕄	100%	•	
Transform Transform Extraction Alimentation base Access Alimentation cube Palo Alimentation dimension Palo Alimentation fichier Alimentation fichier Excel Alimentation fichier propriétés Alimentation fichier XML		Inction depuis fichier CSV Weka S Weka S Weka S tats exécution torique I Trace I Statistiques	coring	Alimentatio	n fichier Excel
Sénération flux RSS	÷	Nom étape	N°Copie	Lignes lues	Lignes écrites
Insertion / Mise à jour table	1	Extraction depuis fichier CSV	0	0	270
S Insertion dans Salesforce	2	Weka Scoring	0	270	270
JI Insertion dans table	3	Alimentation fichier Excel	0	270	270
Mise à jour de Salesforce	•				•

We obtain the following data file. The predicted values are in the last column.

B	Meart-predicted.xls [Mode de compatibilité]											x		
	A	В	С	D	E	F	G	Н		J	K	L		-
1	age	sexe	type_doule	pression	cholester	sucre	electro	taux_max	angine	depression	pic	vaisseau	coeur predicted	
2	70.0) masculin	D	130.00	322.00	A	С	109.00	non	2.40	2.00	D	presence	
3	67.0) feminin	С	115.00	564.00	A	С	160.00	non	1.60	2.00	A	absence	
4	57.0) masculin	В	124.00	261.00	A	Α	141.00	non	.30	1.00	Α	absence	
5	64.0) masculin	D	128.00	263.00	A	Α	105.00	oui	.20	2.00	В	presence	
6	74.0) feminin	В	120.00	269.00	A	С	121.00	oui	.20	1.00	В	absence	
7	65.0) masculin	D	120.00	177.00	A	Α	140.00	non	.40	1.00	Α	absence	
8	56.0) masculin	С	130.00	256.00	В	С	142.00	oui	.60	2.00	В	absence	
9	59.0) masculin	D	110.00	239.00	A	С	142.00	oui	1.20	2.00	В	presence	
10	60.0) masculin	D	140.00	293.00	A	С	170.00	non	1.20	2.00	С	presence	
11	63.0) feminin	D	150.00	407.00	A	С	154.00	non	4.00	2.00	D	presence	
12	59.0) masculin	D	135.00	234.00	A	A	161.00	non	.50	2.00	A	absence	
13	53.0) masculin	D	142.00	226.00	A	С	111.00	oui	.00	1.00	A	absence	
14	44.0) masculin	С	140.00	235.00	A	С	180.00	non	.00	1.00	A	absence	\mathbf{T}
H	< ► H S	1eet1 🦯 🗘											►	L .::

For instance, the value "presence" is assigned to the first individual, "absence" for the second individual, etc.

3.4 Exporting PMML from other tools

We have built our model with SIPINA above. In the two subsections that follow, we show that it is possible to perform the same approach with other software. The key is that they can correctly export models in accordance with the PMML format.

3.4.1 Exporting PMML from Knime

We have defined the following learning process under Knime. The PMML WRITER component is used to export the predictive model.

File Edit View Search Run Node Help				
<u> 📑 - 🖩 🕼 🖋 - ½ - ½ - 1 🌾 (</u>	🍒 ▼ 🛛 🤣 😓 🕴 100%	- 9 % # =		
🚵 Workflow Projects 🛛 🗆 🗆	🛓 *0: Heart - Decis	🛓 *2: Heart - Decis	🛛 🚬 🖓 5 👘 🗖	💁 Node Description 🛛 📃 🗖
(
A Heart - Decision Tree PMML - Deployment				PMML Writer
A Heart - Decision Tree PMML - Learning	ARFF Reader	Decision		
		Tree Learner	PMML Writer	Writes a model from a DMML model port into a
				PMML v3.1 compliant file. Select a file to write to.
			PHIL	If the file already exist it is silently overwritten! Be
IO 🔺	Node 10		t 👝 t	sure to have write access to the entered file. If a
state and the second se		Node 2		PMML file from another version is read by the
Write			Node 3	is converted into PMML v3.1. If the model is not
APEE Writer			4	valid (unknown data types, etc.) an exception is
Table Writer			1	thrown during execution.
PMMI Writer				
Model Writer			-	Dialog Options
🔁 Artificial Data		- B)[F		
🛢 Cache				
🟮 Database		KN	IME Console	*****
式 Data Manipulation	ANT hander Dediton Tree Learner	MML Wher LC	g file is located at: C	:\Program Files\Knime\knime 2.1.1\worksp
Q Data Views	······································	WA	RN File Reader	The file 'file:/D:/DataMining/Databases
Σ Statistics	Node 10 Node 2	🚣 WA	RN File Reader	The file 'file:/D:/DataMining/Databases_
Wining S.a. Mate		WA	RN PMML Writer	File exists and will be overwritten
				T
- mac				

3.4.2 Exporting PMLL from RapidMiner

In the same way, we define the same process under RapidMiner (5.0.010). The WRITE PMML component enables to export the classification tree in the PMML format.



4 Model deployment with Knime

PDI-CE is intended for the data management, its use in the model deployment is therefore selfevident. But other tools can do it also, as long as they are able to manage the data files and to correctly interpret the models. The adoption of the PMML standard is a valuable asset in this process.

We show briefly how to deploy a model with Knime here. The model is the one created with SIPINA previously. We define the following workflow. We notice the two sources of information: FILE READER for the data file containing the unlabeled instances; PMML READER for the model in the

PMML format. We apply the model on the dataset using the DECISION TREE PREDICTOR. We visualize the instances including the assigned labels with the INTERACTIVE TABLE component.



5 Weka model file and PDI-CE

Both Weka and PDI-CE belong to the "Pentaho Community Edition" suite. They can communicate via a proprietary (and binary) file format.

5.1 Creating and storing a model with Weka

We use the Weka 3.7.2 version in this section. The WekaScoring version must be compatible. After we launch Weka in the "Explorer" mode, we load the "heart-train.arff" data file. We activate the "Classify" tab and we select the J48 induction tree algorithm. We start the learning process by clicking on the START button.



To store the predictive model, we right-click on the model and we select the SAVE MODEL menu item. We set "heart-weka.model" as file name.

5.2 Deployment under PDI-CE

We are back in PDI-CE. We use the same diagram as above, except that we change the model. Instead of the PMML file from SIPINA, we select the "heart-weka.model" file created by Weka.

Extraction depuis fichier CSV We	ka Scoring Alimentation fichier Excel
🍞 Weka Scoring	
Step name	Weka Scoring
Model file Fields mapping Mod	el)
Load/import model	:v_and_comparison\pmml\heart-weka.model 🏼 Parcourir
Update model	
Save updated model	Parcourir
Accept model filename from	
File name field	*
Cache loaded models in memory	
Output probabilities	
	OK Annuler

We launch again the process (F9). We obtain a new version of the "heart-predicted.xls" output file.

🚫 Pe	ntaho D	ata Integratio	on - heart			1			1	State of the second				
Fichie	Fichier Edition Vue Action Tools Aide													
1	🗎 👶 🗉 🗟 Perspective: 😥 Data Integration										tion			
	🔯 Navigateur 🥢 Palette de création 💦 🔓 Bienvenue 🔀 heart 🛛													
Fond	Fonctions 🕒 🕒 🕨 🔳 🗮 🗮 🎋 🇭 🚱 🥵 🕵 🖪 100% 🗸													
	Transform Extraction Alimentation base Access Alimentation fichier Excel Alimentation fichier Excel Alimentation fichier Excel													
		Α	В	С	D	E	F	G	Н	- I	J	K L	M	
	1	age	sexe	type_doule	pression	cholester	sucre	electro	taux_max	angine	depression p	ic vaisse	au coeur predicted	
	2	70.00	masculin	D	130.00	322.00	Α	С	109.00	non	2.40	2.00 D	presence	
	3	67.00	feminin	С	115.00	564.00	Α	С	160.00	non	1.60	2.00 A	absence	
	4	57.00	masculin	В	124.00	261.00	A	A	141.00	non	.30	1.00 A	absence	
	5	64.00	masculin	D	128.00	263.00	Α	Α	105.00	oui	.20	2.00 B	presence	
	6	74.00	feminin	В	120.00	269.00	Α	С	121.00	oui	.20	1.00 B	absence	
	7	65.00	masculin	D	120.00	177.00	Α	Α	140.00	non	.40	1.00 A	absence	-
	14 4	→ >I Sh	eet1 ⁄ 🖓	/			-	-		Í 4				1.:

Note: Interestingly, we have compared the predictions of SIPINA and WEKA. We observe that they are inconsistent for 43 cases (18 +25) about 270.

		WE	KA	
		Étiquettes de 🗾		
	Étiquettes de 🗾	presence	absence	Total général
INA	presence	90	25	115
SI	absence	18	137	155
	Total général	108	162	270

6 Conclusion

The deployment of the models is the last step of the data mining process in many situations. In the case of supervised learning, we want to assign labels to unseen instances using the model learning from the labeled dataset. We emphasize the PMML format because it tries to establish a standard for the model description. If this standard is recognized by the most of the tools, it becomes really powerful. In this tutorial, we show that we can create the model using a data mining tool such as SIPINA (or Knime, RapidMiner...) and deploy it using PDI-CE, a tool intended for the data management and the data transformation.

Of course, PMML is not the unique format for the models description. When the tools are included in the same software suite, we can use a proprietary format (e.g. Weka and PDI-CE). We expect that the performances (computation time mainly) are better in this case. It would be interesting to compare the performances (PMML vs. proprietary format) in an upcoming tutorial.