1 Objectif

Déploiement des modèles prédictifs avec « Pentaho Data Integration Community Edition » (PDI-CE), utilisation du standard de description PMML.

Le déploiement des modèles est une étape importante du processus Data Mining. Dans le cadre de l'apprentissage supervisé, il s'agit de réaliser des prédictions en appliquant les modèles sur des observations non étiquetées. Nous avons décrit à maintes reprises la procédure pour différents outils (ex. <u>Tanagra, Sipina</u>, <u>Spad</u>, ou encore <u>R</u>). Ils ont pour point commun d'utiliser le même logiciel pour la construction du modèle et son déploiement.

Ce nouveau didacticiel se démarque des précédents dans la mesure où nous utilisons un logiciel tiers pour le classement des nouvelles observations. Il fait suite à une remarque qui m'a été faite par **Loïc LUCEL** (merci infiniment Loïc pour tes précieuses indications) qui m'a fait prendre conscience de deux choses : le déploiement donne sa pleine mesure lorsqu'on le réalise avec un outil dédié au management des données, nous prendrons l'exemple de PDI-CE (Kettle); nous accédons à une certaine universalité lorsque nous décrivons les modèles à l'aide de standards reconnus/acceptés par la majorité des logiciels, en l'occurrence le standard de description <u>PMML</u>.

J'avais déjà parlé à plusieurs reprises de PMML. Mais jusqu'à présent, je ne voyais pas trop son intérêt si nous n'avons pas en aval un outil capable de l'appréhender de manière générique. Dans ce didacticiel, nous constaterons qu'il est possible d'élaborer un arbre de décision avec différents outils (SIPINA, KNIME et RAPIDMINER), de les exporter en respectant la norme PMML, puis de les déployer de manière indifférenciée sur des observations non étiquetées via PDI-CE. L'adoption d'un standard de description des modèles devient particulièrement intéressante dans ce cas.

Un peu à la marge de notre propos, nous décrirons des solutions de déploiement alternatives dans ce didacticiel. Nous verrons ainsi que Knime possède son propre interpréteur PMML. Il est capable d'appliquer un modèle sur de nouvelles données, quel que soit l'outil utilisé pour l'élaboration du modèle. L'essentiel est que le standard PMML soit respecté. En ce sens, Knime peut se substituer à PDI-CE. Autre piste possible, Weka, qui fait partie de la suite « Pentaho Community Edition », possède un format de description propriétaire directement reconnu par PDI-CE.

2 Données

Nous utilisons les données « <u>heart</u> », bien connues de la communauté de l'apprentissage automatique¹. Nous souhaitons prédire la présence ou l'absence d'une maladie cardiaque (cœur) à partir des caractéristiques des patients (âge, sexe, etc.). Nous disposons de deux fichiers : **heart-train.arff** est le fichier d'apprentissage, il servira à la construction de l'arbre de décision ; **heart-unlabeled.txt** correspond aux observations non étiquetées sur lesquelles nous appliquerons le modèle prédictif. Au final, l'objectif est de produire un nouveau fichier comprenant les descripteurs repris de **heart-unlabeled.txt**, et une colonne supplémentaire « prédiction » indiquant la classe (présence ou absence de maladie cardiaque) affectée à chaque individu.

¹ <u>http://archive.ics.uci.edu/ml/datasets/Heart+Disease</u>

Déploiement avec PDI-CE 3

Installation de PDI-CE et du plug-in « WekaScoring » 3.1

Nous avons décrit l'installation de PDI-CE dans un précédent didacticiel². Pour le déploiement des modèles prédictifs, nous devons intégrer le plug-in « WekaScoring ». Comme son nom l'indique, il sert à manipuler les modèles issus de Weka (avec un format propriétaire, nous y reviendrons plus loin – section 5); mais plus largement, il sait également gérer les modèles au format PMML, quelle qu'en soit la provenance.

Dans un premier temps, il faut récupérer le plug-in sur le site web de Pentaho. Attention, il ne faut pas mélanger les versions. Pour notre part, nous utilisons PDI-CE 4.0.1, nous avons chargé le plug-in WekaScoring version 4.x, adapté aux modèles produits à l'aide de Weka version 3.7.2 et ultérieures.



Puis, nous le désarchivons dans le répertoire d'installation de PDI-CE. Nous devrions avoir une configuration ressemblant à celle-ci sur le disque dur.

A COMPANY AND A COMPANY AND A COMPANY						
G Programmes > pdi-ce-4.0.1-stable > data-integration	▶ pl	ugins 🕨 steps 🕨 WekaScori	ing D C	Rechercher		٩
Fichier Edition Affichage Outils ?						
🆣 Organiser 🔻 🏢 Affichages 🔻 🚯 Graver	_		_	_		0
Liens favoris		Nom	Date de modificati	Туре	Taille	
 Documents Images Musique Autres » 		distrib docs libext libext libswt src libext libex	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	Dossier de fichiers Dossier de fichiers Dossier de fichiers Dossier de fichiers Dossier de fichiers		
Dossiers spoon bit steps DummyPlugin S3CsvInput ShapeFileReader3 WekaScoring	• •	tests WekaScoringDeploy build.xml LICENSE.bt LICENSE-xpp3.bt LICENSE-XStream.bt README.bt	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 11/01/2008 14:41 11/08/2010 14:35	Dossier de fichiers Dossier de fichiers Document XML Document texte Document texte Document texte	4 Ko 35 Ko 3 Ko 2 Ko 1 Ko	
12 éléments (espace libre : 334 Go)				j 📮 0	rdinateur	

Ensuite, nous devons copier le fichier « weka.jar » issue de la distribution WEKA (la version 3.7.2 en ce qui me concerne) dans le sous-répertoire « WekaScoringDeploy ».

² http://tutoriels-data-mining.blogspot.com/2010/09/pentaho-data-integration.html

WebsConing & alwairs & store & WebsConing &	Wal	ra Scaring Danlay			
Fichier Edition Affichage Outlik ?	wer	asconingDepioy	• • • • • • • • • • • • • • • • • • •	nercher	~
Organiser III Affichages Reference Graver				_	0
Liens favoris		Nom	Date de modificati	Туре	Taille
Documents Images Musique Autres >>		plugin.xml scoring.jar weka.jar WS.png xpp3_min-1.1.3.4.0.jar xstream-1.2.2.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	Document XML Executable Jar File Executable Jar File Image PNG Executable Jar File Executable Jar File	1 Ko 44 Ko 5 902 Ko 2 Ko 25 Ko 365 Ko
Uossiers Image: Instant Sector Sect	4 × 4				
6 éléments (espace libre : 333-Go)-				j Ordinateur	r

Enfin, pour vérifier le tout, le plus simple est de lancer PDI-CE. L'icône WekaScoring doit être disponible dans la branche TRANSFORM de la palette de création (créez un nouveau projet de transformation vide pour la vérification – Fichier / Nouveau / Transformation).

🚫 Pentaho Data Integration - Tra	nsformation 1
Fichier Edition Vue Action	Tools Aide
1 6 🗉 🖬 🗃	Perspective: 💢 Data Integration
Palette de création ³⁰ 1	😪 Bienvenue 🛛 🐹 Transformation 1 🖾
Fonctions 🗉	III III R * P R - R - R - R - R - R - R - R - R - R
 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 	
 Expérimental Étapes obselètes Chargement en bloc 	~
Historique étapes	K

Pour pouvoir réaliser le déploiement via PDI-CE, il nous faut construire l'arbre de décision et le sauver au format PMML. Nous utilisons SIPINA pour cela.

3.2 Création et sauvegarde d'un modèle au format PMML avec SIPINA

Pour cette partie, assurez vous de disposer de SIPINA 3.4. Vous pouvez vérifier le numéro de version dans la barre de titre du logiciel.

Sipina Research Version 3.4	x
File Edit Data Statistics Induction method Analysis View Window Help	
🖸 🖄 🐏 🖕	
Attribute selection	Â
Var_1	E
Learning method	
MethodName=Improved ChAID (Tsc A MethodClassName=TArbreDecisionI	
Hdl=8 E	
Split=0.001 TypeBonferroni=1	
Sampling=0 -	
Examples selection	
0 examples idle	-
	P .
Improved ChAID (Tschuprow Goodness of Split)	11

Si ce n'est pas le cas, le plus simple est de récupérer la dernière version sur le site de distribution du logiciel (<u>http://sipina.over-blog.fr/</u>), puis de l'installer par-dessus la précédente.

Chargement des données. Après avoir démarré SIPINA, nous chargeons les données en actionnant le menu FILE / OPEN... Nous sélectionnons le fichier « heart-train.arff » au format Weka.

💦 Si	pina Research Version 3.4						
File	Edit Data Statistics Induction m	ethod Analysi	s View Window	v Help			
	New						
	Open	Jaamina cat cdi					-
	Save	💦 Ouvrir					x
	Save as	Regarder dans :	Domml 📗		• + 1	। 💣 🎟 🕶 ।	
	Export	Ca.	Nom Date	de m Type	Taille	Mots-clés	
	Subsample management	Emplacements récents	eart-train.arff				
	Exit						
		Bureau					
Meth	odName=Improved ChAID (Tsc 🔺						
Meth Hdl=	odClassName=TArbreDecision	Maison					
Merg	e=0.05						
Туре	Bonferroni=1	Ordinateur					
Samp	eBonferroni=1 bling=0	Ciuli lateur					
Exa	mples selection	<u></u>					
1 exa	mples selected	Réseau	Nom du fichier :	heart-train.arff		<u> </u>	Ouvrir
0.000			Types de fichiers :	WEKA File Format (*	arff)	·	Annuler
Impro	oved ChAID (Tschuprow Goodness of S	plit)					

Nous disposons de 13 variables et 270 observations.

5 🚯 😫 🕒				,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,		i icip			
1 <u>X</u>		age	sexe	type_douleu	pression	cholester	sucre	electro	tau:
ttribute selection	1	70.00	masculin	D	130.00	322.00	A	С	109
	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
earning method	10	63.00	feminin	D	150.00	407.00	A	С	154
fethodName=Improve 🔺	11	59.00	masculin	D	135.00	234.00	A	A	161
fethodClassName=TA	12	53.00	masculin	D	142.00	226.00	A	С	111
tore=0.05	13	44.00	masculin	С	140.00	235.00	A	С	180
plit=0.001	14	61.00	masculin	A	134.00	234.00	A	A	145
ypeBonferroni=1	15	57.00	feminin	D	128.00	303.00	A	С	159
alueBonterroni=1	16	71.00	feminin	D	112.00	149.00	A	A	125
Europhing=0	17	46.00	masculin	D	140.00	311.00	A	A	120
Examples selection	18	53.00	masculin	D	140.00	203.00	в	С	155

Construction de l'arbre. Nous devons spécifier le rôle des variables avant de démarrer la phase d'apprentissage. Pour ce faire, nous cliquons sur le menu ANALYSIS / DEFINE CLASS ATTRIBUTE. Nous plaçons CŒUR en CLASS, les autres variables (12) en ATTRIBUTES.

Sipina Research Version 3.4 - [Learning set editor]								
🔊 File Edit Data S	Statistics In	duction metho	A	analysis View Window Help	_ & ×			
🖰 🐴 📬				Define class attribute				
X		age se	ĸe	Select active examples	re electro tau -			
Attribute selection	1	70.00 ma	sc	Set weight field	C 105			
	2	67.00 fe	nir	Attribute selection				
	3	57.00 ma	sc		Variables			
	4	64.00 ma	sc	Llass	age			
	5	74.00 fe	nir	Icoeur	sexe			
	6	65.00 ma	sc	- Au.3. A	type_douleur pression			
	7	56.00 ma	sc	Attributes	cholester			
	8	59.00 ma	sc	sexe	sucre			
	9	60.00 ma	sc	type_douleur	taux_max			
Learning method	10	63.00 fe	nir	cholester =	angine			
MethodName=Improve 🔺	11	59.00 ma	sc	sucre	pic			
MethodClassName=TA	12	53.00 ma	sc	taux_max	vaisseau			
Merge=0.05	13	44.00 ma	sc	angine	COCU			
Split=0.001	14	61.00 ma	sc					
TypeBonferroni=1	15	57.00 fe	nir					
Sampling=0	16	71.00 fe	nir	C Only discrete				
Examples selection	17	46.00 ma	sc	C Only continuous				
	18	53.00 ma	sc	© Both				
	•		1					
J	Editing N	IEW.FDM			🗸 OK 🛛 🗶 Annuler			
Improved ChAID (Tschup	row Goodnes	is of Split)		15	Exec. Lime : 15 ms.			

Il ne nous reste plus qu'à actionner le menu ANALYSIS / LEARNING... pour construire l'arbre de décision. Nous obtenons un modèle relativement simple, 3 variables parmi les 12 candidates interviennent dans la définition de l'arbre.



Exportation de l'arbre au format PMML. Il faut exporter l'arbre au format PMML. Nous actionnons le menu TREE MANAGEMENT / EXPORT DECISION TREE. Nous nommons le fichier « heart.xml ».



Remarque : Choisir l'extension « .xml » semble plus judicieux pour PDI-CE. Mais en réalité, le contenu des fichiers « .xml » et « .pmml » est strictement le même.

Comprendre le format PMML. Voyons justement le contenu du fichier « heart.xml ». On peut le subdiviser en plusieurs parties : (A) la définition des données (« DataDictionnary ») ; (B) la description de l'arbre avec la définition du problème d'apprentissage (« MiningSchema ») -- c.-à-d. spécifier la variable cible et les variables prédictives -- et l'énumération des nœuds successifs de l'arbre (« Node »).

- Concernant le dictionnaire de données (Figure 1), nous observons par exemple que la variable « sexe » est catégorielle, avec deux valeurs (« value ») possibles : « masculin » et « féminin ». En revanche, la variable « pression » est quantitative, la valeur minimale « leftMargin » (resp. maximale, « rightMargin ») observée est 94 (resp. 200). Nous utilisons la notation scientifique.
- Le schéma d'apprentissage (Figure 2) spécifie le rôle particulier de « cœur » dans le modèle, il s'agit de la variable à prédire.
- Enfin, lors de la description de l'arbre (Figure 3), nous distinguons les correspondances entre le contenu du fichier PMML et la représentation graphique de la structure. Pour chaque nœud nous observons : l'effectif global, la proposition qui la défini, les effectifs par classe.

```
<DataDictionary numberOfFields="13">
<DataField name="age" optype="continuous" dataType="double">
</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="2.00000000000E+002"/>
</DataField>
<DataField name="cholester" optype="continuous" dataType="double">
<Interval closure="closedClosed" leftMargin="1.260000000000E+002" rightMargin="5.64000000000E+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="0.000000000000000E+000" rightMargin="6.19999980926514E+000"/>
</DataField>
<DataField name="pic" optype="continuous" dataType="double">
<Interval closure="closedClosed" leftMargin="1.0000000000000E+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>
</DataDictionarv>
```

Figure 1 - PMML - Description des données

```
<TreeModel modelName="DecisionTree" functionName="classification" s</pre>
<MiningSchema>
<MiningField name="age"/>
<MiningField name="sexe"/>
<MiningField name="type douleur"/>
<MiningField name="pression"/>
<MiningField name="cholester"/>
<MiningField name="sucre"/>
<MiningField name="electro"/>
<MiningField name="taux max"/>
<MiningField name="angine"/>
<MiningField name="depression"/>
<MiningField name="pic"/>
<MiningField name="vaisseau"/>
<MiningField name="coeur" usageType="predicted"/>
</MiningSchema>
```





Figure 3- Description de l'arbre

3.3 Classement via PDI-CE et le plug-in WekaScoring

Nous souhaitons appliquer cet arbre sur les observations à classer « heart-unlabeled.txt ». Nous démarrons PDI-CE, puis nous créons un nouveau projet « Transformation » (FICHIER / NOUVEAU / TRANSFORMATION).



Importation des observations à classer. Nous chargeons le fichier avec le composant « Extraction depuis fichier CSV » (branche *EXTRACTION*). Nous le paramétrons en actionnant le menu contextuel « Editer étape ».



Attention, le caractère tabulation est le séparateur de colonne (« Séparateur champs »). Il n'est pas visible dans l'éditeur, mais il faut veiller à cliquer sur le bouton adéquat pour le spécifier. Nous cliquons sur le bouton « Récupérer Champs » pour obtenir la description des variables.

Classement avec le composant WekaScoring. Nous rajoutons le composant « WekaScoring », nous lui relions l'outil précédent. Nous paramétrons WekaScoring en actionnant le menu contextuel « Editer étape ». Nous importons le modèle « heart.xml » (bouton Load/Import model).

🚫 Pentaho Data Integration - Transformation 1 (mo	odifiée)
Fichier Edition Vue Action Tools Aide	
1 📁 🖬 🔚	Perspective: 👿 Data Integration
🔯 Navigateur 🥖 Palette de création	🔁 Bienvenue 🛛 💥 Transformation 1 🛛
Fonctions	🕨 🖩 🖿 🖎 🌾 🗭 🔯 🗞 🦗 🥰 🔚 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 Expérimental Étapes obselètes 	Step name Weka Scoring odel file Fields mapping Model Load/import model r_soft_dev_and_comparison\pmml\heart.xm[Parcourir Update model Save updated model Save updated model Parcourir File name field Stache loaded models in memory Output probabilities
Chargement en bloc	OK Annuler
	< ×

Création du fichier de sortie. Nous souhaitons créer un fichier de sortie au format Excel. Il doit contenir, pour chaque observation, les descripteurs et la classe prédite. Nous ajoutons le composant « Alimentation fichier Excel» (branche *ALIMENTATION*). Nous le paramétrons de manière à spécifier le fichier de sortie « heart-predicted.xls ».



Exécution de la transformation. Aboutissement de toutes ces manipulations, nous lançons les opérations en actionnant le menu ACTION / EXECUTER (F9).



Nous retrouvons le fichier des observations étiquetées « heart-predicted.xls » dans le répertoire de destination idoine. Nous l'ouvrons dans le tableur Excel, les prédictions se situent dans la dernière colonne « cœur_predicted ».

1	heart-pre	licted.xls [Mo	de de compa	tibilité]									= x
	A	B	С	D	E	F	G	H		J	K	L	
1	age	sexe	type_doule	pression	cholester	sucre	electro	taux_max	angine	depression	pic	vaisseau	coeur predicted
2	70.	00 masculin	D	130.00	322.00	A	C	109.00	non	2.40	2.00	D	presence
3	67.	00 feminin	С	115.00	564.00	A	C	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	A	A	105.00	oui	.20	2.00	В	presence
6	74.	00 feminin	В	120.00	269.00	A	C	121.00	oui	.20	1.00	В	absence
7	65.	00 masculin	D	120.00	177.00	A	A	140.00	non	.40	1.00	A	absence
8	56.	00 masculin	С	130.00	256.00	В	C	142.00	oui	.60	2.00	В	absence
9	59.	00 masculin	D	110.00	239.00	A	С	142.00	oui	1.20	2.00	В	presence
10	60.	00 masculin	D	140.00	293.00	A	C	170.00	non	1.20	2.00	С	presence
11	63.	00 feminin	D	150.00	407.00	A	C	154.00	non	4.00	2.00	D	presence
12	59.	00 masculin	D	135.00	234.00	A	A	161.00	non	.50	2.00	A	absence
13	53.	00 masculin	D	142.00	226.00	A	C	111.00	oui	.00	1.00	A	absence
14	44.	00 masculin	С	140.00	235.00	A	С	180.00	non	.00	1.00	A	absence
14 -		Sheet1 ⁄ 🗘											► U.:

Le 1^{er} individu par exemple est affecté à la classe « cœur = présence », le second à « absence », etc.

3.4 Exportation des modèles au format PMML avec d'autres logiciels

Nous avons construit notre arbre de décision avec SIPINA. Dans les deux sous-sections qui suivent, nous montrons qu'il est possible de reproduire la même démarche avec d'autres logiciels. L'essentiel est qu'ils sachent exporter correctement les modèles en respectant la norme PMML.

3.4.1 Elaboration et exportation des modèles avec Knime

Nous avons élaboré le diagramme suivant sous Knime. Le composant PMML WRITER sert à exporter le modèle qui lui est connecté. Dans notre cas, il s'agit d'un arbre de décision.

A KAIDAF				
A KNIME				
File Edit View Search Run Node Help				
📑 • 🖬 🗟 🛷 • 🖅 🖓 • 🦘	Qa - 💛 👾 100%	- 🔍 😪 🕮 💻	9 0 🖶 📀 🖉	
🖄 Workflow Projects	🛆 *0: Heart - Decis	🔺 *2: Heart - Decis.		💁 Node Description 🛛 🗌 🗖
() () () () () () () () () () () () () (_			
A Heart - Decision Tree PMML - Deployment				PMML Writer
Heart - Decision Tree PMML - Learning	ARFF Reader	Decision		=
🛕 Node Repository			PMML Writer	Writes a model from a PMML model port into a
- ☆ ↔ ▼				PMML v3.1 compliant file. Select a file to write to. If the file already exist it is silently overwritten! Be
n IO n	Node 10			sure to have write access to the entered file. If a
Bread		Node 2		PMML file from another version is read by the
- Write			Node 3	PMML Reader and directly written by this hode, it is converted into PMML v3.1. If the model is not
The CSV Writer			44	valid (unknown data types, etc.) an exception is
ARFF Writer				thrown during execution.
able Writer				-
PHIL PMML Writer				Dialog Options
Artificial Data			<u>.</u>	
Cache	🗄 Outline 🖾		Console 🛛	📭 🚮 📑 🖬 🖛 🗂 🗖
		KN	ME Console	
Contraction State	AMT Sundar Baddin			
Q Data Views		MMU Wher	g file is located at: C	:\Program Files\Knime\Knime_2.1.1\Worksp
Σ Statistics	Node 10	A WA	NN File Reader	The file 'file:/D:/DataMining/Databases
Mining	Node 2	Node 3 W2	NN PMMI Writer	File evists and will be overwritten
Meta			A FILD WITCH	The calous and whit be overwritten
🔁 Misc 👻		•		•

3.4.2 Elaboration et exportation des modèles avec RapidMiner

De la même manière, sous RapidMiner (version 5.0.010; l'interface a beaucoup changé depuis nos précédents tutoriels). Nous pouvons construire le fichier PMML à partir d'un arbre de décision.



4 Déploiement avec d'autres logiciels (Knime)

PDI-CE est un outil destiné au management des données, son implication dans le déploiement de modèles est donc parfaitement naturel. Mais d'autres logiciels peuvent s'en charger, pourvu qu'ils sachent gérer les données et interpréter correctement les modèles. L'adoption d'un standard, le PMML, est certainement un atout indéniable dans ce cadre.

Nous montrons très succinctement comment déployer un modèle avec Knime. Nous avons utilisé le diagramme suivant. On notera la double source d'informations, données (File Reader) et

modèle (PMML Reader), en entrée de l'outil de prédiction (Decision Tree Predictor). Les observations sont directement visualisées avec le composant InteractiveTable.

A KNIME	Color	- 16 · A 4	
<u>File Edit View</u> Search <u>R</u> un Node <u>H</u> elp			
📑 • 🔚 🕲 🛷 • 🗄 • 🖗 • 🛛	🖌 🗧 🤣 🔤 100% 🕞 🖳 😽 🖬	💻 🖻 🔘 📾 🕲 🖉 🧟	
🚵 Workflow Projects	🛆 *0: Heart - Decis 🛛 🛆 *2: Heart - D	ecis »s	🗖 🗖 Node Description 🛛 📃 🗖
Cars - Visualization	PMML Reader	Decision Tree Predictor Interactive	Table Table This node reads any model from a file. This node reads any model from a file. This node reads any model requiring node. Enter the source location in the configuration dialog of the node. When executed, the node reads the model from the XML description in to the specified file and provides it at its output port.
A Model Reader	Node 4		Dialog Options T
- Write TEL CSV Writer	E Outline 🛛 🗖 🗆	Console 🛛	
ARFF Writer ArFF Writer Model Writer And Model Writer Artificial Data Cache Database	Rathat Rathathat Rathat	KNIME Console WARN PMML Writer WARN PMML Writer WARN PMML Writer WARN PMML Writer WARN PMML Writer	File exists and will be overwritten File exists and will be overwritten File exists and will be overwritten File exists and will be overwritten Please select a PMML file!

5 Format de description Weka et PDI-CE

Weka et PDI-CE appartiennent tous deux à la suite « Pentaho Community Edition ». Ils peuvent communiquer via un format de description natif (et binaire) des modèles.

5.1 Construction et sauvegarde du modèle sous Weka

Il faut faire très attention aux versions des logiciels et des plugins dans cette partie. Nous avons utilisé la version 3.7.2 de Weka, la version de WekaScoring doit être à l'avenant. Après avoir démarré Weka en mode « Explorer » et chargé le fichier « heart-train.arff », nous activons l'onglet « Classify ». Nous sélectionnons la méthode J48, puis nous actionnons le bouton START.



Pour sauver le modèle prédictif, nous réalisons un clic-droit sur l'arbre produit, et nous actionnons l'item SAVE MODEL du menu contextuel. Nous spécifions le nom « heart-weka.model ».

5.2 Exploitation du modèle sous PDI-CE

Nous revenons dans PDI-CE, nous utilisons le même diagramme que précédemment, sauf que nous modifions le modèle utilisé. Au lieu du fichier PMML issu de SIPINA, nous sélectionnons le fichier « .model » produit par Weka.



Il ne nous reste plus qu'à lancer le processus (menu ACTION / EXECUTER). Nous obtenons une nouvelle version du fichier « heart-predicted.xls ».

🚫 Pentaho Data Integration - heart		1 5 5 5 100								
Fichier Edition Vue Action Tools Aide										
1 🧉 🗉 🗃 🔚		Perspective:	😥 Data Integration							
🔯 Navigateur 🥖 Palette de création	Navigateur 🥟 Palette de création 💦 Bienvenue 💢 heart 🖂									
Fonctions	🕨 ii 🖩 b. 🏶 🖻 📑 🗞	🖇 🜊 🖪 100% 👻								
Transform Extraction Alimentation Alimentation base Access Alimentation cube Palo Alimentation fichier Alimentation fichier Alimentation fichier Alimentation fichier Excel	Extraction depuis fichier CSV	Weka Scoring Alimente	tion fichier Excel							
A B C	D E F	G H I	J K L	M						
1 age sexe type doule pr	ession cholester sucre	electro taux max angine	depression pic vaisseau	coeur predicted						
2 70.00 masculin D	130.00 322.00 A	C 109.00 non	2.40 2.00 D	presence						
3 67.00 feminin C	115.00 564.00 A	C 160.00 non	1.60 2.00 A	absence						
4 57.00 masculin B	124.00 261.00 A	A 141.00 non	.30 1.00 A	absence						
5 64.00 masculin D	128.00 263.00 A	A 105.00 oui	.20 2.00 B	presence						
6 74.00 feminin B	120.00 269.00 A	C 121.00 oui	.20 1.00 B	absence						
7 65.00 masculin D	120.00 177.00 A	A 140.00 non	.40 1.00 A	absence 🗸						
H + > > Sheet1										

Remarque : A titre de curiosité, nous avons voulu confronter les prédictions de SIPINA avec celles de WEKA. Nous constatons qu'elles sont plus ou moins cohérentes : 227 (90+137) prédictions sur 270 sont concordantes. Nous ne savons pas en revanche, sans investigations supplémentaires (ex. évaluation des performances en <u>validation croisée</u>), lequel des deux est le plus précis.

		WEKA		
	Nombre de sipin Étiquettes de 💌			
	Étiquettes de 💌 presence	absence	Total général	
ANIA	presence	90	25	115
SIF	absence	18	137	155
	Total général	108	162	270

6 Conclusion

Le déploiement des modèles est (d'une certaine manière) l'aboutissement du Data Mining. Dans le cas de l'apprentissage supervisé, il s'agit de classer des individus avec un modèle préalablement construit. Nous avons mis l'accent sur le format de description PMML. Il cherche à s'imposer comme un standard de description. C'est sa principale force. S'il est adopté par tous, l'utilisation d'un outil ETL tel que PDI-CE pour le déploiement devient alors indépendant du logiciel de Data Mining utilisé pour produire le modèle de prédiction.

Bien entendu, PMML ne constitue certainement pas la panacée. C'est une solution comme une autre. Comme nous avons pu le constater, lorsque les logiciels se reconnaissent (ou appartiennent à la même suite, comme c'est le cas de Weka et PDI-CE), l'utilisation d'un format d'échange natif est possible. Il devrait en résulter des performances optimisées... *sinon on ne voit pas trop l'intérêt de produire un format propriétaire*.

Ce dernier point reste à vérifier. Une idée intéressante pour un prochain tutoriel serait justement de comparer la vitesse d'exécution du composant WekaScoring lors de l'appréhension de très gros fichiers, selon que le modèle est décrit avec le format binaire Weka ou avec la norme PMML.