TD3 : Anonymisation et Analyse de Données

M2 IMIS et MIAGE

Temps prévu : 2h00

Dans ce TD, nous allons travailler avec les logiciels ARX (logiciel d'anonymisation) et WEKA (logiciel d'analyse de données). Nous nous intéresserons à la qualité des opérations de classification et de clustering.

Préparation du TD :

Téléchargez les logiciels ARX et WEKA. Ces logiciels sont réalisés en Java, il est donc souvent plus simple de télécharger juste le JAR exécutable, qui fonctionnera peu importe votre plateforme. Il existe également des versions installables.

ARX : <u>https://arx.deidentifier.org/downloads/</u>

WEKA : https://waikato.github.io/weka-wiki/downloading_weka/

Téléchargez également le fichier contenant les données à analyser : il s'agit d'un fichier sur des analyses de diabete des indiennes de la tribu Pima.

Données : https://benjamin-nguyen.fr/ENS/4ASTI-EA-BIGDATA-SECU/pima-indians-diabetes.csv

Ce fichier peut être chargé tel quel dans WEKA et ARX.

I- Analyse de données brutes

Nous débutons ce TD en analysant le fichier contenant les micro-données brutes.

1) Lancement de WEKA GUI CHOOSER

Lancez WEKA et cliquez sur Explorer



2) Exploration du jeu de données

Dans la fenête Explorer ouvrez le fichier que vous venez de télécharger. Pensez bien à choisir le type de fichier CSV.

🜍 Open		×
Look In:	vekanx 💽 🕋	
i student bina-india bina-india bina-india bina bina bina bina bina bina bina bi	ns-diabetes.csv csv	Linvoke options dialog
File <u>N</u> ame:	C:\Users\Benjamin\Google Drive\ENS\wekarx	
Files of <u>T</u> ype:	CSV data files (*.csv)	•
		Open Cancel

Une fois chargé l'écran devrait ressembler à ceci :

🕼 Weka Explorer	– – ×
Preprocess Classify Cluster Associate Select attributes Visualize	
Open file Open URL Open DB Ger	nerate Undo Edit Save
Choose	Apply Stop
Current relation Relation: pima-indians-diabetes Attributes: 9 Instances: 768 Sum of weights: 768	Selected attribute Name: Pregnancy Type: Numeric Missing: 0(0%) Distinct: 17 Unique: 2 (0%)
Attributos	Statistic Value
All None Invert Pattern	Minimum 0 Maximum 16 Mean 3.844 StdDev 3.365
1 Pregnancy 2 Glucose 3 BP 4 TricepsThickness 5 Insulin 6 BM	
7 DiabetesPedigree 8 Age 9 Class	Visualize All
Remove	143 102 87 67 20 23 24 20 10 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
Status	8 16
0К	Log x0

On voit que ce jeu de

données comporte 9 attribut (incluant l'attribut class que l'on souhaite prédire) et 768 instances (lignes ou individus). Les 8 attributs que nous allons analyser sont :

- le nombre de fois que la patiente a été enceinte (Pregnancy)
- son taux de glucose après ingestion au bout de 2h (Glucose)

- sa tension artérielle (BP en mm Hg)
- l'épaisseur de la peau de son triceps (TricepsThickness en mm)
- la prise d'insuline au bout de 2h (Insulin en mu U/ml)
- l'indice de masse corporelle (BMI en (kg/m)^2)
- la fonction pedigree de diabete (DiabetesPedigree)
- son age en années (AGE)

Et ce, afin de prédire le diagnostic de diabete (Class, YES ou NO).

3) Classification avec kNN

Appuyez sur l'onglet « Classify », puis le bouton « choose ». Une fenêtre s'ouvre pour choisir le classifieur que vous voulez utiliser. Choisissez le répertoire *lazy* puis le classifieur *IBk*

C	🕽 Weka Explo	rer								
	Preprocess	Classify	Cluster	Associate	Select attributes	Visualize				
CI	assifier									
L	V 🚔 weka	ssifiers			r	NNSearch -A	"weka.core.Euclid	eanDistance - F	R first-last(""	
		bayes functions lazy				JUL				
		Bk KStar LWL								
		meta misc rules								
0		1000								
Re										
					Close					

Cliquez ensuite sur le classifieur comme indiqué sur la figure, pour ouvrir la fenêtre de configuration du classifieur.

			A 15	
Weka Explorer		—		×
Preprocess Classify Cluster Associate	Select attributes Visualize			
Classifier				
Choose IBk	oursearch.LinearNNSearch - A "weka.core EuclideanDistance - R use lastr"			
Test options	Classifier output			
◯ Use training set				
O Supplied test set Set				
Cross-validation Folds 10				
O Percentage split % 66				
More options				
(Nom) Class				
Start Stop				
Result list (right-click for options)				

Vous pouvez maintenant configurer le classifieur. Plusieurs paramètres peuvent nous intéresser ici : le paramètre KNN (nombre de voisins, nous allons le faire varier de 1 à 20). Le champ distanceWeighting permet de pondérer les voisins selon leur distance (dit autrement, on peut donner ou pas plus d'importance aux voisins selon leur proximité). D'autres éléments comme nearestNeighbourSearchAlgorithm permettent de configurer le type d'algorithme de recherche utilisé pour optimiser le temps d'exécution.

	🙆 weka gui GenericObjectEditor	× –
.n	Wekalgundenencobjecteurtor	
-	weka.classifiers.lazy.lBk	-
1	About	
	Manager and a statistic state of the second	
_	K-nearest neighbours classifier.	More
		Capabilities
	KNN	1
-		
	batchSize	100
_	crossValidate	False
	debug	False
1	distanceWeighting	No distance weighting
		Talan P
	doivotCheckCapabilities	raise
	meanSquared	False
	n a areathlaigh haur Caarch Algorithm	
	nearesuvergrooursearchAigonum	Choose Linear NNSear Ch - A weka.core.Euclid
	numDecimalPlaces	2
	windowSize	0
	Open Save	OK Cancel
_		

Choisissez KNN=1, distanceWeighting=no distance weighting et cliquez sur OK.

Choisissez dans Test options la *cross-validation* folds = 10 et vérifiez que c'est bien (Nom) Class que vous cherchez à prédire, puis appuyez sur le bouton Start. Vous observez les résultats suivants dans la fenêtre de droite :

=== Run information ===

Scheme: weka.classifiers.lazy.IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""

- Relation: pima-indians-diabetes
- Instances: 768

Attributes: 9

Pregnancy

Glucose

ΒP

TricepsThickness

Insulin

BMI

DiabetesPedigree

Age

Class

Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

IB1 instance-based classifier

using 1 nearest neighbour(s) for classification

Time taken to build model: 0 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances 536 69.7917 %

Incorrectly Classified Instances	232	30.2083 %
Kappa statistic	0.3211	
Mean absolute error	0.3027	
Root mean squared error	0.5488	
Relative absolute error	66.5897 %	
Root relative squared error	115.1446 %	
Total Number of Instances	768	

=== Detailed Accuracy By Class ===

Area H	PRC 2	Area	TP Rate Class	FP Rate	Precision	Recall	F-Measure	MCC ROC
0,646	l	0,465	0,522 YES	0,208	0,574	0,522	2 0 , 547	0,322
0,646	l	0,729	0,792 NO	0,478	0,756	0,792	2 0,773	0,322
Weighte 0,646	ed A	vg. 0,637	0,698	0,384	0,692	0,69	8 0,694	0,322

=== Confusion Matrix ===

a b <-- classified as 140 128 | a = YES 104 396 | b = NO

La ligne Weighted Avg. peut être utilisée pour mesurer la qualité de ce classifieur. Nous allons considérer la métrique F-Measure (qui vaut ici 0.694) comme indication de la qualité de la classification. Plus cette valeur est proche de 1, meilleure est la prédication.

Q1- QUESTION : Faites varier les paramètres KNN, nearestNeighbourSearchAlgorithm et distanceWeighting pour essayer d'améliorer la qualité de votre classifieur, que vous estimerez par la valeur F-MEASURE comme indiquée plus haut. Tracez par exemple la fonction représetant la F-Measure, en fonction de K. Pour quelle valeur observez-vous la meilleure qualité de classification ? Indiquez cette qualité.

Dans la suite, nous prendrons KNN = 7 et distanceWeighting = 1-distance. Notez la valeur de la F-Measure, qui servira de référence.

4) Classification avec J48

Choisissez le classifieur par arbre trees/J48



Le paramètre qui va nous intéresser ici est minNumObj qui indique le nombre minimum d'instances qu'on souhaite avoir dans une feuille de décision, ainsi que confidenceFactor (le facteur de confiance) de chaque classe.

Voici ce qui est affiché pour la valeur minNumObj=2 et confidenceFactor=0.25

```
=== Run information ===
```

Scheme:	weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:	pima-indians-diabetes
Instances:	768
Attributes:	9
	Pregnancy
	Glucose
	BP
	TricepsThickness
	Insulin
	BMI
	DiabetesPedigree

```
Class
Test mode: 10-fold cross-validation
=== Classifier model (full training set) ===
J48 pruned tree
_____
Glucose <= 127
| BMI <= 26.4: NO (132.0/3.0)
| BMI > 26.4
| | Age <= 28: NO (180.0/22.0)
| Age > 28
  | | Glucose <= 99: NO (55.0/10.0)
| | Glucose > 99
1
 | | | DiabetesPedigree <= 0.56: NO (84.0/34.0)
| | | DiabetesPedigree > 0.56
| | | Pregnancy <= 6
  | | | | Age <= 30: YES (4.0)
| | | Age > 30
| | Age <= 34: NO (7.0/1.0)
| | | Age > 34
          | | | BMI <= 33.1: YES (6.0)
| | | | | BMI > 33.1: NO (4.0/1.0)
1
| | | Pregnancy > 6: YES (13.0)
Glucose > 127
| BMI <= 29.9
| | Glucose <= 145: NO (41.0/6.0)
| | Glucose > 145
```

Age

```
| | Age <= 25: NO (4.0)
| | | Age > 25
 | | | Age <= 61
|
        | | BMI <= 27.1: YES (12.0/1.0)
L
  | | BMI > 27.1
        | | BP <= 82
Ι
  | | | DiabetesPedigree <= 0.396: YES (8.0/1.0)
| | | DiabetesPedigree > 0.396: NO (3.0)
| | | BP > 82: NO (4.0)
| | Age > 61: NO (4.0)
BMI > 29.9
1
| | Glucose <= 157
  | | BP <= 61: YES (15.0/1.0)
| | BP > 61
| | Age <= 30: NO (40.0/13.0)
| | | Age > 30: YES (60.0/17.0)
1
| | Glucose > 157: YES (92.0/12.0)
Number of Leaves : 20
Size of the tree : 39
Time taken to build model: 0.01 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances 567
                                            73.8281 %
Incorrectly Classified Instances 201
                                             26.1719 %
```

Kappa statistic	0.4164
Mean absolute error	0.3158
Root mean squared error	0.4463
Relative absolute error	69.4841 %
Root relative squared error	93.6293 %
Total Number of Instances	768

=== Detailed Accuracy By Class ===

Area P	PC Area	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC ROC
Alea I	NC AIEd	CIASS					
0 , 751	0 , 572	0,597 YES	0,186	0,632	0,597	0,614	0,417
0,751	0,811	0,814 NO	0,403	0,790	0,814	4 0,802	0,417
Woighto	d Arro	0 7 2 0	0 227	0 725	0 7 2	0 0 7 2 6	0 417
0,751	0,727	0,730	0,327	0,755	0,75	0 0,730	0,417

```
=== Confusion Matrix ===
```

a b <-- classified as
160 108 | a = YES
93 407 | b = NO</pre>

Regardez bien le champ J48 Pruned Tree qui représente l'arbre de décision. Vous pouvez également visualiser l'arbre via un clic droit sur votre exécution (dans Result List -> visualize tree)

_	14:34:37 - lazy.ll 14:37:22 - trees 14:38:06 - trees 14:38:15 - trees	J48 J48 J60 108 J	< classified as a = YES b = NO
	14:38:34 - trees	 View in separate window 	
	14:40:44 - trees	. Save result buffer	
		Delete result buffer(s)	
	Status	Load model	-
	ОК	Save model	
	L	Re-evaluate model on current test set	
		Re-apply this model's configuration	12
		Visualize classifier errors	- <u>-</u>
		Visualize tree	9 -
1	, I	Visualize margin curve	

Q2- QUESTION : Faite varier les valeurs minNumObj et confidenceFactor. Notez l'évolution de la qualité de la F-Measure en la traçant. Notez la valeur pour laquelle vous obtenez la meilleure qualité, ainsi que sa valeur.

Dans la suite nous prendrons minNumObj=20 et confidenceFactor=0.75

5) Classification par règles : RIPPER

Nous allons utiliser un 3^e type de classifieur : un classifieur à base de règles logique. Choisissez le classifieur Rules/JRIP



=== Run information ===

Nous allons simplement utiliser JRIP avec les paramètres de base. Nous obtenons le résultat suivant :

```
weka.classifiers.rules.JRip -F 3 -N 2.0 -O 2 -S 1
Scheme:
Relation:
              pima-indians-diabetes
              768
Instances:
Attributes:
               9
               Pregnancy
               Glucose
              ΒP
               TricepsThickness
               Insulin
              BMI
              DiabetesPedigree
               Age
               Class
```

Test mode: 10-fold cross-validation === Classifier model (full training set) === JRIP rules: _____ (Glucose >= 167) => Class=YES (79.0/11.0) (Glucose >= 112) and (DiabetesPedigree >= 0.529) and (BMI >= 30) and (Age >= 30) => Class=YES (48.0/9.0) (Glucose >= 115) and (BMI >= 42.1) => Class=YES (34.0/12.0)(Age >= 31) and (Glucose >= 108) and (BMI >= 28.2) => Class=YES (88.0/38.0) => Class=NO (519.0/89.0) Number of Rules : 5 Time taken to build model: 0.14 seconds === Stratified cross-validation === === Summary === Correctly Classified Instances 572 74.4792 % 196 25.5208 % Incorrectly Classified Instances Kappa statistic 0.4171 0.3461 Mean absolute error Root mean squared error 0.4351 Relative absolute error 76.1388 % 91.2746 % Root relative squared error Total Number of Instances 768

```
=== Detailed Accuracy By Class ===
```

```
TP Rate FP Rate Precision Recall F-Measure MCC
                                                               ROC
Area PRC Area Class
                                         0,556 0,603
               0,556
                      0,154
                               0,659
                                                              0,420
0,708
       0,576
                YES
                      0,444 0,780 0,846 0,812
               0,846
                                                              0,420
0,708
       0,782
                NO
Weighted Avg. 0,745 0,343 0,738
                                          0,745
                                                    0,739
                                                              0,420
0,708
       0,710
=== Confusion Matrix ===
  a b <-- classified as
149 119 | a = YES
 77 423 | b = NO
On s'intéresse plus précisément aux règles générées par JRIP qui sont les suivantes :
JRIP rules:
_____
(Glucose >= 167) => Class=YES (79.0/11.0)
(Glucose >= 112) and (DiabetesPedigree >= 0.529) and (BMI >= 30) and
(Age >= 30) => Class=YES (48.0/9.0)
(Glucose >= 115) and (BMI >= 42.1) => Class=YES (34.0/12.0)
(Age >= 31) and (Glucose >= 108) and (BMI >= 28.2) => Class=YES
(88.0/38.0)
=> Class=NO (519.0/89.0)
```

Number of Rules : 5

On voit qu'il serait également possible de construire un arbre de decision à partir de ces règles.

6) Classification avec des réseaux de neurones profonds

Nous utilisons un dernier type de classifieur : des réseaux de neurones profonds, disponible dans functions/multilayer Perceptron.

weka.gui.GenericObject	tEditor MultilaverPercentren	
About	mululayerPerceptron	
About		
A classifier that uses b perceptron to classify i	backpropagation to learn a multi-layer nstances.	More Capabilities
GUI	False	
autoBuild	True	
batchSize	100	
debug	False	
decay	False	
doNotCheckCapabilities	False	
hiddenLayers	a	
learningRate	0.3	
momentum	0.2	
normalizeAttributes	True	
normalizeNumericClass	True	
numDecimalPlaces	2	
reset	True	
resume	False	
seed	0	
trainingTime	500	
validationSetSize	0	
validationThreshold	20	

De nombreux paramètres sont disponibles, comme le nombre de nœuds dans les niveaux cachés (hiddenLayers, qui vaut « a » ou une valeur entière), le taux d'apprentissage (learningRate) ou encore l'inertie (momentum). Le temps d'apprentissage peut également être contrôlé (trainingTime). Lançons le modèle avec les paramètres par défaut.

```
=== Run information ===
Scheme: weka.classifiers.functions.MultilayerPerceptron -L 0.3 -M 0.2
-N 500 -V 0 -S 0 -E 20 -H a
Relation: pima-indians-diabetes
Instances: 768
Attributes: 9
Pregnancy
```

Glucose ΒP TricepsThickness Insulin BMI DiabetesPedigree Age Class Test mode: 10-fold cross-validation === Classifier model (full training set) === Sigmoid Node 0 Inputs Weights

Inputs	Weights
Threshold	3.628703319715117
Node 2	-6.313299284416023
Node 3	-2.568104564109483
Node 4	-2.8448541489871015
Node 5	-2.531131735140029
Node 6	-2.2169477046738026
oid Node 1	L

Sigmo

Inputs	Weights
Threshold	-3.6287033197148237
Node 2	6.313299284413041
Node 3	2.5681045641093068
Node 4	2.844854148987099
Node 5	2.531131735139932
Node 6	2.216947704673593
Sigmoid Node 2	2

Inputs Weights

Threshold -5.726636007684776 Attrib Pregnancy 0.47208258366115546 Attrib Glucose -6.5452774656896135 Attrib BP 1.9803888455019736 Attrib TricepsThickness -4.1742177349813785 Attrib Insulin 7.048470366054944 Attrib BMI 0.48531328921316397 Attrib DiabetesPedigree 1.9877172681824125 Attrib Age -7.597191482777374 Sigmoid Node 3 Inputs Weights Threshold -2.7031829318357508 Attrib Pregnancy 3.0657225750732575 Attrib Glucose -5.4957416963918035 Attrib BP -9.569096675837606 Attrib TricepsThickness 0.06271383682773789 Attrib Insulin -1.4401488539528373 Attrib BMI -9.10971393376813 Attrib DiabetesPedigree -3.174922522292794 Attrib Age 12.985230736208651 Sigmoid Node 4 Inputs Weights Threshold -2.202935559669145 Attrib Pregnancy -1.4493533000047487 Attrib Glucose -12.089311850014818 Attrib BP -1.3398675531560917 Attrib TricepsThickness 2.967829312368748 Attrib Insulin 1.1802832608623406 Attrib BMI -6.286813870465455 Attrib DiabetesPedigree -6.776007065799867

Attrib Age 2.8238051786015523 Sigmoid Node 5 Inputs Weights Threshold -6.414462475572164 Attrib Pregnancy 7.527527969290638 Attrib Glucose -11.389428921109165 Attrib BP 6.345771244438931 Attrib TricepsThickness 0.22924527140725157 Attrib Insulin -1.341715914555352 Attrib BMI -5.840278388013632 Attrib DiabetesPedigree -1.841522948252757 Attrib Age -16.80717183085853 Sigmoid Node 6 Inputs Weights Threshold 1.5331683098797988 Attrib Pregnancy -10.455311936699879 Attrib Glucose -4.7970387233467635 Attrib BP 3.3118562514122276 Attrib TricepsThickness 1.2804738190655798 Attrib Insulin -0.7339822735340477 Attrib BMI -10.25191137035137 Attrib DiabetesPedigree -2.88526462850306 Attrib Age 8.695516148522088 Class YES Input Node 0 Class NO Input Node 1

Time taken to build model: 0.73 seconds

```
=== Stratified cross-validation ===
=== Summary ===
```

Correctly Classified Instances	578	75.2604 %
Incorrectly Classified Instances	190	24.7396 %
Kappa statistic	0.4508	
Mean absolute error	0.2942	
Root mean squared error	0.4224	
Relative absolute error	64.7386 %	
Root relative squared error	88.6256 %	
Total Number of Instances	768	

=== Detailed Accuracy By Class ===

			TP Rate	FP Rate	Precision	Recall	F-Measure	MCC ROC
Area H	PRC	Area	Class					
0,794		0,658	0,627 YES	0,180	0,651	0,627	7 0,639	0,451
0,794		0,859	0,820 NO	0,373	0,804	0,820	0,812	0,451
Weighte 0,794	ed A	.vg. 0,789	0,753	0,306	0,751	0 , 75	3 0,751	0,451

=== Confusion Matrix ===

a b <-- classified as 168 100 | a = YES

90 410 | b = NO

Les informations de sortie sont récupérées dans les nœuds 0 et 1. La F-Measure vaut ici 0.751.

Q3 : QUESTION : Jouez un peu avec le paramétrage du réseau de neurones. Indiquez les valeurs de paramètres que vous utilisez pour obtenir une meilleure F-Measure.

7) Clustering avec K-Means

Nous allons maintenant effectuer des tâches de clustering. Choisissons l'onglet cluster, puis l'algorithme *SimpleKMeans*.

Cluster data using the K means algorithm.	Capabilities
canopyMaxNumCanopiesToHoldInMemory	100
canopyMinimumCanopyDensity	2.0
canopyPeriodicPruningRate	10000
canopyT1	-1.25
canopyT2	-1.0
debug	False
displayStdDevs	False
distanceFunction	Choose EuclideanDistance -R first-
doNotCheckCapabilities	False
dontReplaceMissingValues	False
fastDistanceCalc	False
initializationMethod	Random
maxIterations	500
numClusters	2
numExecutionSlots	1
preserveInstancesOrder	False
duceNumberOfDistanceCalcsViaCanopies	False
seed	10

Deux paramètres sont pertinents ici : *distanceFunction* qui permet de choisir le type de distance (Euclidienne, Manhattan, etc.) et numClusters qu'on va fixer à 2 (car il y a 2 clusters YES et NO). Lorsqu'on utilise beaucoup d'attibuts catégoriels, il est pertinent d'utiliser la distance Manhattan. Lorsqu'on a des données numériques, on peut préférer utiliser la distance euclidienne. Par ailleurs, lorsqu'on clique sur la métrique qu'on utilise, on peut choisir de ne l'appliquer que sur un sousensemble des dimensions. On peut également décider de normaliser les dimensions ou pas (si on met dontNormalize = false c'est en fait qu'on normalise, c'est-à-dire qu'on considère que toutes les dimensions ont le même poids. Si on met dontNormalize = true les dimensions ayant des valeurs très grandes seront privilégiées. Ici, comme on souhaite donner un poids équivalent à chaque dimension, on choisit dontNormalize = false.

🥥 weka.gui.Gene	ericObjectEditor	\times
weka.core.Euclide	eanDistance	
About		_
Implementing	a Euclidean distance (or similarity) function	
	geocidean distance (or similarity) function.	
attributaladicaa	first lost	_
aunoutennuices		
dontNormalize	False	•
invertSelection	False	•
	· · · · · · · · · · · · · · · · · · ·	
Open	Save OK Cancel	
	maanaaaaan maanaa maanaann	-

Afin d'avoir un retour sur la qualité de notre clustering, il faut régler le mode de clustering sur « Classes to clusters evaluation » et choisir l'attribut qui correspond aux classes, à savoir *Class*. Le bouton « *Ignore Attributes* » permet d'ignorer certaines dimensions. Dans un premier temps, on décide de toutes les conserver.

Choose SimpleKMeans -in	it 0 -max-candidates 10)0 -pe
Cluster mode		c
◯ Use training set		
O Supplied test set	Set	
O Percentage split	% 66	
Classes to clusters evaluation	on	
(Nom) Class	-	
Store Clustere for risualizatio		
lanara attribu	taa	5
Ignore attribu	les	
Start	Stop	
Result list (right-click for options)		

L'exécution de Simple K-Means donne le résultat suivant :

=== Run information ===

```
Scheme: weka.clusterers.SimpleKMeans -init 0 -max-candidates 100 -
periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A
"weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10
Relation: pima-indians-diabetes
Instances: 768
Attributes: 9
Pregnancy
Glucose
BP
TricepsThickness
```

Insulin BMI DiabetesPedigree Age Ignored: Class Test mode: Classes to clusters evaluation on training data === Clustering model (full training set) === kMeans _____ Number of iterations: 5 Within cluster sum of squared errors: 122.87437030193571 Initial starting points (random): Cluster 0: 1,126,56,29,152,28.7,0.801,21 Cluster 1: 8,95,72,0,0,36.8,0.485,57 Missing values globally replaced with mean/mode Final cluster centroids: Cluster# Attribute Full Data 0 1 (768.0) (513.0) (255.0) _____ Pregnancy 3.8438 2.0468 7.4588

Glucose	120.8945	115.4678	131.8118
BP	69.1055	65.9708	75.4118
TricepsThickness	20.5365	21.8285	17.9373
Insulin	79.7995	84.9025	69.5333
BMI	31.9926	31.7854	32.4094
DiabetesPedigree	0.4719	0.4709	0.4738
Age	33.2409	26.8168	46.1647

Time taken to build model (full training data) : 0.03 seconds

=== Model and evaluation on training set ===

Clustered Instances

- 0 513 (67%)
- 1 255 (33%)

Class attribute: Class Classes to Clusters:

0 1 <-- assigned to cluster 134 134 | YES 379 121 | NO

Cluster 0 <-- NO

Cluster 1 <-- YES

```
Incorrectly clustered instances : 255.0 33.2031 %
```

La qualité de la classification est indiquée sur la dernière ligne : 33% des instances sont mal classifies (et donc 66% donc bien classifiées). Le tableau « Classes to Clusters » permet de comprendre un peu mieux la qualité de la classification : ici le cluster 0 correspond aux individus avec la valeur « NO », il y a 379 corrects (vrai positifs) et 134 erreurs (faux positifs). Pour le cluster 1 (« YES ») il y a 134 corrects (vrai positifs) et 121 erreurs (faux positifs).

8) Clustering par densité

Nous allons utiliser un algorithme de clustering par densité, DBSCAN. Il faut commencer par l'installer, car il n'est pas chargé par défaut. Pour ce faire, fermez la fenêtre WEKA et retournez dans le GUI Chooser de WEKA et allez dans le menu Tools -> Package Manager ce qui ouvre une nouvelle fenêtre.



Tapez « dbscan » dans la ligne « *package search* » puis enter et installez le package « *optics_dbscan* ». Lorsqu'on recharge notre fichier csv et qu'on se rend dans l'onglet « clustering » on dispose maintenant de l'algorithme DBSCAN. Les paramètres à considérer comme vu dans le cours sont minPoints (nombre de voisins) et epsilon (la distance). On peut également changer de fonction distance comme pour KMeans si on le souhaite.

ka.clusterers.f	DBSCAN							
bout								
Basic implementation of DBSCAN clustering algorithm that More								
should *not* be used as a reference for runtime benchmarks:								
instances is	not suppo	prenentations exist clustering of new pred						
	dobug	Falsa						
	debug	Faise						
distance	Eupetion							
distance	Function	Choose EuclideanDistance -R first-last						
distance NotCheckCar	Function	Choose EuclideanDistance -R first-last						
distance pNotCheckCap	Function	Choose EuclideanDistance -R first-last						
distance oNotCheckCap	Function pabilities epsilon	Choose EuclideanDistance -R first-last False 0.9						
distance NotCheckCap	Function pabilities epsilon	Choose EuclideanDistance -R first-last False 0.9						

Lançons l'algorithme avec le paramétrage de base. On obtient le résultat suivant :

Time taken to build model (full training data) : 0.18 seconds

=== Model and evaluation on training set ===

Clustered Instances

0 768 (100%)

Class attribute: Class Classes to Clusters:

0 <-- assigned to cluster

268 | YES

500 | NO

Cluster 0 <-- NO

Incorrectly clustered instances : 268.0 34.8958 %

L'interprétation qu'on peut donner ici, c'est que toutes les instances ont été mises dans le même cluster ! Le résultat n'est donc pas du tout satisfaisant. Pour augmenter le nombre de clusters on peut agir de deux manières : augmenter le nombre de voisins et/ou réduire la distance.

Q4 : QUESTION : La configuration de DBSCAN est relativement délicate. Essayez de modifier ces paramètres pour obtenir un clustering de relativement bonne qualité (au moins 2 clusters, au moins 100 instances clusterées dont moins de 30% d'instances mal clusterées). Indiquez vos paramétrages et commentez la qualité de vos résultats.

Utilisez le paramétrage suivant : epsilon = 0.15 et minPoints = 4. Interprétez les résultats.

Voilà, vous avez pu expérimenter plusieurs techniques d'analyse de données classiques. Nous pouvons constater que leur paramétrage n'est pas si simple, et que la qualité obtenue est très variable. Les techniques d'analyse supervisées (classification) produisent de manière assez naturelle des résultats de meilleure qualité que les techniques non supervisées (clustering).

II- Anonymisation avec ARX

Nous allons maintenant utiliser ARX pour anonymiser les données en utilisant le modèle du *k*anonymat. Puis nous allons exécuter des analyses de données comme dans la partie I et comparer les résultats.

9) Mise en place

Lancez ARX et créez un nouveau projet (appelez le « Pima »), puis choisissez le menu File-> import data et chargez le .csv En principe les délimiteurs par défaut sont correctement configurés. Validez les paramètres de typage auto-détectés.

Please pro	ovide the inf	formation re	quested below						
Location	C:\Users\B	C:\Users\Benjamin\Google Drive\ENS\wekan\pima-indians-diabetes.csv							
Charset	WINDOWS	S-1252 (Syste	em default)			~			
Delimiter	,					~			
Quote						~			
-									
Escape						~			
inebreak	Windows								
	First row	v contains co	olumn names						
Pregnanc	First row	v contains co Glucose	blumn names BP	TricepsThickne	Insulin	BMI	D ^		
Pregnanc 6	First row	v contains co ilucose 48	BP 72	TricepsThickne 35	Insulin 0	BMI 33.6	D ^ 0.		
Pregnance 6 1	Cy G	v contains co Glucose 48 5	BP 72 66	TricepsThickne 35 29	Insulin 0 0	BMI 33.6 26.6	D ^ 0. 0.		
Pregnano 6 1 8	First row	v contains co ilucose 48 5 83	BP 72 66 64 55	TricepsThickne 35 29 0	Insulin 0 0 0	BMI 33.6 26.6 23.3 29.1	D ^ 0. 0. 0.		
Pregnance 6 1 8 1 0	First row Fy G 1. 8 1. 8 1. 8 1. 1. 8 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.	v contains co Glucose 48 5 83 9 37	BP 72 66 64 66 40	TricepsThickne 35 29 0 23 35	Insulin 0 0 94 168	BMI 33.6 26.6 23.3 28.1 43.1	D ^ 0. 0. 0. 0. 2.		
Pregnano 6 1 8 1 0 5	Cy G 11- 8 11- 8 11- 8 11- 11- 11- 11- 11- 1	v contains co ilucose 48 5 83 9 37 16	BP 72 66 64 66 40 74	TricepsThickne 35 29 0 23 35 0	Insulin 0 0 94 168 0	BMI 33.6 26.6 23.3 28.1 43.1 25.6	D ^ 0. 0. 0. 0. 2. 0.		
Pregnance 6 1 8 1 0 5 5 2 2 4	First row Gy G 14 14 14 14 14 14 14 17 14 17 14 14 14 14 14 14 14 14 14 14	v contains co ilucose 48 5 83 9 37 16 %	BP 72 66 64 66 40 74 50	TricepsThickne 35 29 0 23 35 0 42 35	Insulin 0 0 94 168 0 8 8	BMI 33.6 26.6 23.3 28.1 43.1 25.6 31.0	D ^ 0. 0. 0. 0. 2. 0. 0. >		

Nous devons maintenant créer 8 hiérarchies pour les données. Commençons avec l'attribut « *pregnancy* ». Par défaut tous les attributs sont classés comme *insensitive*, c'est-à-dire ignorés par ARX. Passons *pregnancy* en quasi-identificateur. La lumière devant lui devient jaune pour indiquer ce changement de statut.

1	ARX A	nonymization Tool -	Pima																
File	Ed	t View Help																	
_	尾 l		추 志 🗸 🗙	*	á 🗉 🗄														
*	Confi	gure transformation	Explore res	ults 🚧	Analyze uti	lity 🕂 Analyze risk													
Inp	out da	ta											Ţ	t I	8 8 0	Data	transfor	mation Attribute metadata	
		 Pregnancy 	 Glucose 	9	BP	TricepsThickness	•	Insulin	Θ	BMI	DiabetesPedigree	e 0	Age	Θ	Class ^	Ту	e:	Insensitive	 Transformat
	1	✓ 6	148	72		35	0		33.6		0.627	50		YES		11		Insensitive	-
	2	✓ 1	85	66		29	0		26.6		0.351	31		NO		Mi	nimum:	Sensitive	Maximum:
	3	8	183	64		0	0		23.3		0.672	32		YES				Quasi-identifying	
	4	✓ 1	89	66		23	94		28.1		0.167	21		NO				Identifying	_
	5	0	137	40		35	168		43.1		2.288	33		YES					
	6	✓ 5	116	74		0	0		25.6		0.201	30		NO					
	7	3	78	50		32	88		31.0		0.248	26		YES					
	8	10	115	0		0	0		35.3		0.134	29		NO					
	9	2	197	70		45	543		30.5		0.158	53		YES					
	10	✓ 8	125	96		0	0		0.0		0.232	54		YES					
	11	✓ 4	110	92		0	0		37.6		0.191	30		NO					
	12	10	168	74		0	0		38.0		0.537	34		YES					
	10	10	130	an .		n	n		27.1		1 // 1	57		NO					

Faisons de même pour tous les attributs sauf *class*.

En cliquant sur l'onglet Analyse Utility et Distribution, on peut voir les valeurs que prennent chaque attribut.

🧊 arx	Anonymization Tool	- Pima								
File E	dit View Help									
🖪 属	😹 😹 🗟 📓	[슈송] 🗸 🗙	a 4	1 🔲 🚸 🗏 🔠 😣						
🚠 Con	figure transformation	Steplore result	t 📯 An	alyze utility 🛛 🔱 Analyze ris	k					
Input o	ata Classification	performance Qua	ality mode	Is					J 🕇 🗎	E 0
	Pregnancy	 Glucose 		BP TricepsThickness	s o Insulin	 BMI 	DiabetesPedigree	o Age	0	Class ^
1	6	148	72	35	0	33.6	0.627	50	YES	
2	✓ 1	85	66	29	0	26.6	0.351	31	NO	
3	8	183	64	0	0	23.3	0.672	32	YES	
4	✓ 1	89	66	23	94	28.1	0.167	21	NO	
5	0	137	40	35	168	43.1	2.288	33	YES	
6	5	116	74	0	0	25.6	0.201	30	NO	
7	✓ 3	78	50	32	88	31.0	0.248	26	YES	
8	10	115	0	0	0	35.3	0.134	29	NO	
9	2	197	70	45	543	30.5	0.158	53	YES	
10	8	125	96	0	0	0.0	0.232	54	YES	
11	☑ 4	110	92	0	0	37.6	0.191	30	NO	
12	10	168	74	0	0	38.0	0.537	34	YES	
13	1 0	139	80	0	0	27.1	1.441	57	NO	
14	✓ 1	189	60	23	846	30.1	0.398	59	YES	
15	5	166	72	19	175	25.8	0.587	51	YES	
16	7	100	0	0	0	30.0	0.484	32	YES	
17	0	118	84	47	230	45.8	0.551	31	YES	
18	7	107	74	0	0	29.6	0.254	31	YES	
19	✓ 1	103	30	38	83	43.3	0.183	33	NO	
20	1	115	70	30	96	34.6	0.529	32	YES	
21	3	126	88	41	235	39.3	0.704	27	NO	
22	8	99	84	0	0	35.4	0.388	50	NO	
23	7	196	90	0	0	39.8	0.451	41	YES	
24	9	119	80	35	0	29.0	0.263	29	YES	
25	✓ 11	143	94	33	146	36.6	0.254	51	YES	
26	10	125	70	26	115	31.1	0.205	41	YES	
27	7	147	76	0	0	39.4	0.257	43	YES	
28	✓ 1	97	66	15	140	23.2	0.487	22	NO	
29	✓ 13	145	82	19	110	22.2	0.245	57	NO	
30	5	117	92	0	0	34.1	0.337	38	NO	
31	5	109	75	26	0	36.0	0.546	60	NO	
		100	70	he	D.45	D1 6	0.051	00	MEC	Ň
	6	~		\ \	· · · ·					-
Summ	ary statistics Distribut	tion Continger	ncy Class	sizes Properties Classificatio	n models					V
15	·									
P 2										
10 U		1	<u> </u>							
af .										
spo										
25			î.	······································		ii				
0						ųų			i,	<u> </u>
	0 1	2 3	4	5 6	7 8	9 10	11 12	13	14	15

On voit par exemple que l'attribut *pregnancy* varie de 0 à 15 avec une faible proportion des individus ayant une valeur > 10.

10) Construction de Hiérarchies

Construisons maintenant la hiérarchie de généralisation en retournant sur l'onglet *configure transformation* et en appuyant sur le bouton *create hierarchy* et utilisons des intervalles (premier choix proposé).

🤡 AR)	Anonymization Tool	- Pima				
File	dit View Help		\sim			
🖪 🎜	<i>B B B B</i>	🎄 🚲 🖌 🗙	🚓 📓 💷	∲目≣ 9		
- Co	figure transformation	Explore res	sults 🚧 Analyze	utility 🖖 Analyze ris	k	
<u> </u>					-	
Input	data					Uata transformation Attribute metadata
	 Pregnancy 	 Glucose 	 BP 	TricepsThicknes	s o Insulin	BMI DiabetesPedigree • Age • Class ^ Type: Quasi-identifying ~
1	6	148	72	35	0	🦻 Hierarchy wizard — 🗆 🗙 🚽
2		85	66	29	0	
3	⊻ 8	183	64	0	0	Create a generalization hierarchy
4		89	66	23	94	Specify the type of hierarchy
5		137	40	35	168	
6		110	/4	0	0	Use dates (for dates)
7	≥ 3	/8	50	32	88	Use intervals (for variables with ratio scale)
8	V 10	115	0	0	0	O Use ordering (e.g., for variables with ordinal scale)
9	⊻ 2	197	/0	45	543	O Use masking (e.g., for alphanumeric strings)
10	≥ 8	125	96	0	0	
11	⊻ 4	110	92	0	0	-
12	≥ 10	168	/4	0	0	
13	10	139	80	0	0	
14	⊻ 1	189	60	23	846	
15	⊻ 5	166	72	19	175	
16	7	100	0	0	0	
17	0	118	84	47	230	
18	7	107	74	0	0	
19	≥ 1	103	30	38	83	
20	≥ 1	115	70	30	96	
21	≥ 3	126	88	41	235	
22	≥ 8	99	84	0	0	
23	7	196	90	0	0	
24	9	119	80	35	0	
25	☑ 11	143	94	33	146	
26	✓ 10	125	70	26	115	
27	7	147	76	0	0	Help Load Save < Back Next > Finish Cancel
28	✓ 1	97	66	15	140	

Les intervalles sont disponibles dès lors qu'on traite des données numériques. Si on traite des données catégorielles, il faut utiliser « ordering » qui permet d'ordonner ses données comme on le souhaite, puis construire des regroupements à plusieurs niveaux.

Commençons par une construction automatique de la hiérarchie.

Hierarchy wizard			_	
reate a hierarchy by defining inte	ervals			
pecify the parameters				
10, 17[0, 17]				
General Range Interval Group				
Aggregate function: Default				~
Function Parameter:				
Min: 0				
Mars 17				
IVIDX:				

Cliquez sur le premier intervalle, puis mettez la valeur max à 1. Ensuite faites un clic droit sur ce nouvel intervalle et choisissez « add new level » ce qui va rajouter un niveau à la hiérarchie.

	u				
eate a hieraro pecify the parame	chy by defining ^{eters}	intervals			<
10 11 10 11					
R	emove				
A	dd before				
A	dd after				
N	ferge down				
N	/lerge up				
A	dd new level				
A	dd new level				
A	dd new level				
eneral Range In	dd new level				
eneral Range In Aggregate functio	udd new level Iterval Group Dn: Default				
eneral Range In Aggregate functio	dd new level iterval Group on: Default ier:				
A ieneral Range In Aggregate functio Function Paramet Min:	dd new level terval Group on: Default er: 0				
eneral [Range] In Aggregate functio 'unction Paramet Vin: Vax:	dd new level terval Group Default er: 0 1				
eneral [Range] In Aggregate function Function Paramet Viin: Max:	dd new level				

Cliquez ensuite sur ce nouveau niveau et passez la valeur *group* à 2, ce qui signifie qu'on veut grouper 2 intervalles précédents ensemble pour créer ce nouveau niveau. Un nouvel ensemble (l'intervalle [1;2[) est créé automatiquement (il est grisé pour indiquer qu'il est construit automatiquement.



Continuez à construire des niveaux jusqu'à arriver au niveau [0,16[Votre hiérarchie doit ressembler à ceci :



Validez ensuite.

Q5- QUESTION : Créez de même une hiérarchie pour l'attribut *Glucose* et pour l'attribut *Age.* Pour l'attribut Age, le premier intervalle à créer est l'intervalle [21-22]

Q6- QUESTION : Créez de même des hiérarchies pour BP, TricepsThickness et BMI. Vous créerez une hiérarchie de profondeur 5 comme pour *BP* (il suffit de s'arrêter après avoir créé 5 niveaux de profondeur).

Pour ce qui est d'Insulin, on voit que les valeurs varient jusqu'à 857, mais très peu de valeurs sont supérieures à 250. Nous allons donc utiliser le « snapfrom » c'est-à-dire remplacer tout ce qui dépasse une certaine valeur par la valeur maximale (topcoding). Ce champ n'est qu'à remplir une seule fois, puis on procède de la même manière que précédemment.

Lower bound	al Group	Upper bound		
Minimum value:	0	Snap from:	250	
Bottom coding from:	0	Top coding from:	847	
Snap from:	0	Maximum value:	847	

On voit après construction de la hiérarchie (à 5 niveaux) que toutes les valeurs supérieures à 249 sont mises dans l'intervalle 249-857 dès le premier niveau de généralisation.

view (n verview of	e nierarcny f groups and values						<
Groups	Table						
186	Level-0	Level-1	Level-2	Level-3	Level-4	Level-5	Level
136	237	[237, 238[[237, 238[[237, 238[[237, 240[[236, 240[[235, 240[
110	240	[240, 241[[240, 241[[240, 241[[240, 241[[240, 241[[240, 241[
94	245	[245, 246]	[244, 246[[244, 246[[245, 248[[245, 246[[241, 246[
84	249	[249, 847[[249, 847[[249, 847[[249, 847[[246, 847[[247, 847[
1	250	[249, 847[[249, 847[[249, 847[[249, 847[[246, 847[[247, 847[
	255	[249, 847[[249, 847[[249, 847[[249, 847[[246, 847[[247, 847[
	258	[249, 847[[249, 847[[249, 847[[249, 847[[246, 847[[247, 847[
	265	[249, 847[[249, 847[[249, 847[[249, 847[[246, 847[[247, 847[
	270	[249, 847[[249, 847[[249, 847[[249, 847[[246, 847[[247, 847[
	271	[249, 847[[249, 847[[249, 847[[249, 847[[246, 847[[247, 847[
	272	[249, 847[[249, 847[[249, 847[[249, 847[[246, 847[[247, 847[
	274	[249, 847[[249, 847[[249, 847[[249, 847[[246, 847[[247, 847[
	275	[249, 847[[249, 847[[249, 847[[249, 847[[246, 847[[247, 847[
	277	[249, 847[[249, 847[[249, 847[[249, 847[[246, 847[[247, 847[
	278	[249. 847]	[249, 847]	[249, 847]	[249, 847]	[246, 847]	[247, 847]

Q7- QUESTION : Créez la hiérarchie de profondeur 5 pour DiabetesPedigree, en utilisant un snapfrom à 1.6 et en créant le premier intervalle entre [0 ;0.1[.

Voici ce que vous devriez obtenir (ou quelque chose approchant) :

Hierarchy	wizard						
view the	e hierarchy						~
rerview of	groups and values						
Groups	Table						
517	Level-0	Level-1	Level-2	Level-3	Level-4	Level-5	
16	1.394	[1.3, 1.4]	[1.2, 1.4]	[1.2, 3.42]	[0.8, 3.42]	[0.078, 3.42]	
4	1.400	[1.4, 1.5]	[1.4, 3.42]	[1.2, 3.42]	[0.8, 3.42[[0.078, 3.42[
2	1.441	[1.4, 1.5]	[1.4, 3.42]	[1.2, 3.42]	[0.8, 3.42]	[0.078, 3.42]	
1	1.461	[1.4, 1.5]	[1.4, 3.42]	[1.2, 3.42]	[0.8, 3.42]	[0.078, 3.42]	
	1.476	[1.4, 1.5]	[1.4, 3.42]	[1.2, 3.42]	[0.8, 3.42]	[0.078, 3.42]	
	1.600	[1.5, 3.42]	[1.4, 3.42]	[1.2, 3.42]	[0.8, 3.42]	[0.078, 3.42]	
	1.698	[1.5, 3.42]	[1.4, 3.42]	[1.2, 3.42]	[0.8, 3.42]	[0.078, 3.42[
	1.699	[1.5, 3.42]	[1.4, 3.42]	[1.2, 3.42]	[0.8, 3.42]	[0.078, 3.42]	
	1.731	[1.5, 3.42]	[1.4, 3.42]	[1.2, 3.42]	[0.8, 3.42]	[0.078, 3.42]	
	1.781	[1.5, 3.42]	[1.4, 3.42]	[1.2, 3.42]	[0.8, 3.42]	[0.078, 3.42]	
	1.893	[1.5, 3.42]	[1.4, 3.42]	[1.2, 3.42]	[0.8, 3.42]	[0.078, 3.42]	
	2.137	[1.5, 3.42]	[1.4, 3.42]	[1.2, 3.42]	[0.8, 3.42]	[0.078, 3.42]	
	2.288	[1.5, 3.42[[1.4, 3.42]	[1.2, 3.42]	[0.8, 3.42]	[0.078, 3.42[
	2.329	[1.5, 3.42]	[1.4, 3.42]	[1.2, 3.42]	[0.8, 3.42]	[0.078, 3.42[
	2.420	[1.5, 3.42]	[1.4, 3.42]	[1.2, 3.42]	[0.8, 3.42]	[0.078, 3.42]	
	<						>
	Help	Load	Save	< Back	Next >	Finish	Cancel
				D			

11) Anonymisation selon le k-anonymat

Il ne nous reste plus qu'à définir le modèle d'anonymisation que nous voulons utiliser !

1.144	[1.1, 1.2]	[1, 1.2[[0.8, 1.2]	[0.8, 3.42]	[0.078, 3.42]			
1.154	[1.1, 1.2[[1, 1.2[[0.8, 1.2]	[0.8, 3.42]	[0.078, 3.42]			
1.159	[1.1, 1.2]	[1, 1.2[×
1.162	[1.1, 1.2]	[1, 1.2]						~
1.174	[1.1, 1.2[[1, 1.2[Ac	dd a new privacy	model				
1.182	[1.1, 1.2[[1, 1.2[p	lease select a privacy i	model which will b	e applied to the data set			
1.189	[1.1, 1.2]	[1, 1.2]						
1.191	[1.1, 1.2]	[1, 1.2[Ty	pe Model		Attribute			
1.213	[1.2, 1.3[[1.2, 1. 6	(ε δ)-Differen	tial privacy				
1.222	[1.2, 1.3[[1.2, 1. K) k-Anonymity					
1.224	[1.2, 1.3[[1.2, 1. (k) k-Map					
1.251	[1.2, 1.3[[1.2, 1. (8) δ-Presence	δ-Presence				
1.258	[1.2, 1.3[[1.2, 1. G) Profitability	Profitability				
1.268	[1.2, 1.3[[1.2, 1. (1) Average-reide	ntification-risk				
1.282	[1.2, 1.3[[1.2, 1. (1) Population-ur	Population-uniqueness				
1.292	[1.2, 1.3[[1.2, 1. (1) Sample-uniqu	ieness				
1.318	[1.3, 1.4[[1.2, 1.						
1.321	[1.3, 1.4[[1.2, 1.	ofiguration					- Q
1.353	[1.3, 1.4]	[1.2, 1.						_ ~~
1.390	[1.3, 1.4[[1.2, 1. K:	2					
1.391	[1.3, 1.4[[1.2, 1.						-
1.394	[1.3, 1.4[[1.2, 1.	Note: ye	ou can also enter va	alues by double-clicking the c	ontrol knobs		
1.400	[1.4, 1.5[[1.4, 3.						
1.441	[1.4, 1.5[[1.4, 3.	OK	Cancel				
<								
rivacy models	Population Co	osts and benefits						
Туре	Model				Attribute			
General settings	Utility measur	e Coding mode	Attribute weights					
2								
Suppression lim	t: 0%							
Approximate:	Assume pr	actical monoton	icity					
	p		,					
recomputation	: 🗌 Enable. Thi	reshold: 0%						

Appuyez sur le bouton + puis choisissez le k-anonymat et k=2. Rajoutons également une limite de suppression de 5%. Puis pressez le bouton « anonymize »

🤣 A	RX A	۱no	nymization Tool -	Pima				
File	Ed	it	View Help	\frown				
🖪 l	4	3	🛃 🐺 🐺 i	🖧 🛵 🗹 🏓 d	🕹 🖾 💷 🖗	目 ☷ 🛛 😢		
- C	Confi	gui	re transformation	Stepson Edit ->	Anonymize e utili	ty 🕂 Analyze risk		
Inpu	ut da	ta						
			 Pregnancy 	 Glucose 	• BP	TricepsThickness	 Insulin 	• BI
1	1	~	5	44	62	0	0	25.0
ž	2	~	7	62	78	0	0	32.6
3	3	~	0	57	60	0	0	21.7
4	4	~	1	0	48	20	0	24.7
-	5	✓	3	61	82	28	0	34.4
(5	~	9	57	80	37	0	32.8
7	7	~	1	0	68	35	0	32.0
8	В	~	5	0	80	32	0	41.0
(9	\checkmark	6	0	68	41	0	39.0

La fenêtre qui s'affiche donne un nombre de pas maximum pour l'anonymisation. Si votre problème d'anonymisation est difficile (i.e. possède beaucoup d'attributs) il faudra peut être augmenter cette valeur. Mais pour notre application, c'est inutile.

Au bout de quelques instants l'anonymisation aura eu lieu.

Nous pouvons visualiser les résultats en allant dans Analyze Utility, et en naviguant les onglets Summary statistics, distribution, etc. Comparons les données initiales et les données anonymes.

Summary statistics Distribution Contingency Class sizes Properties Classification models									
Measure	Value (incl. suppressed)	Value (excl. suppressed)							
Average class size	1 (0.13021%)	1 (0.13021%)							
Maximal class size	1 (0.13021%)	1 (0.13021%)							
Minimal class size	1 (0.13021%)	1 (0.13021%)							
Suppressed records	0 (0%)	0							
Number of classes	768	768							
Number of records	768	768							

Summary statistics Distribution Contingency Class sizes	Properties Classification models			
Measure	Value (incl. suppressed)	Value (excl. suppressed)		
Average class size	10.20833 (1.32921%)	10.20833 (1.38889%)		
Maximal class size	139 (18.09896%)	139 (18.91156%)		
Minimal class size	2 (0.26042%)	2 (0.27211%)		
Suppressed records	33 (4.29688%)	0		
Number of classes	72	72		
Number of records	768	735		

On voit que les données anonymes n'ont plus que 72 classes (par opposition aux 768 intiales, c'est-àdire que chaque individu était unique ...) et qu'on a supprimé 33 enregistrements (soit 4.3%).

On peut comparer aussi des distributions, par exemple celle de *glucose* (il faut appuyer sur l'attribut en haut de l'écran)



On voit que l'age a été totalement généralisé (ou supprimé).



Cela signifie que si on voulait utiliser l'age pour faire la prédiction c'est devenu impossible !

Pour changer les critères utilisés pour la généralisation, il faut se rendre dans l'onglet *explore results*. On voit qu'il est possible de changer certains paramètres, par contre ARX n'a pas trouvé de solution sans qu'on généralise l'age. Il serait certainement possible de trouver des solutions, mais il faudrait lui donner plus de temps pour faire le calcul. Pour effectuer une autre transformation, il suffit de cliquer sur la transformation de treillis qui nous intéresse et faire « apply transformation ». On peut par exemple changer le degré de généralisation de certains attributs.



12) Analyse de risque

L'analyse de risque est disponible via l'onglet Analyze Risk. Le risque initial sur chaque enregistrement de la base est de 100% car chaque enregistrement était unique sur les 8 QID utilisés. Après anonymisation, on peut voir une très grosse baisse de cette possibilité d'attaque via les QID. En effet, le risque maximal est de 50% (c'est-à-dire 1/k, comme k=2), et on voit que seuls 7% des enregistrements de la base atteignent ce risque.



Ce graphique permet de quantifier le risque lié à la réidentification par quasi-identifieur d'un attaquant. C'est au responsable de traitement de données d'accepter ou non ce risque. Souvent une valeur de risque maximale acceptable sera plutôt à moins de 20% (soit k>=5)

13) Export et Analyse de données

Exportons maintenant nos données (en format .csv).

🧭 Al	RX A	nonymization Tool -	Pima											
File	Edi	t View <u>Help</u>												
I	3 B B B I () ふふ イ×ふ ふ 回 今 目 II 9													
- h C	靠 Configure transfo File -> Export data jsults [🕪 Analyze utility 😽 Analyze risk]													
Inpu	uput data 🛛 Classification performance Quality models 🔹 🗈 🖻 🕑 Output data 🖉 Classification performance Quality models													
		 Pregnancy 	 Glucose 	• BP	TricepsThickness	 Insulin 	 BMI 	DiabetesPedigree	 Age 	 Class ^ 		 Pregnancy 	 Glucose 	o Bi
1		✓ 1	0	68	35	0	32.0	0.389	22	NO	1	 0, 8[[0, 32[•
2		5	0	80	32	0	41.0	0.346	37	YES	2	 0, 8[[0, 32[*
3		✓ 6	0	68	41	0	39.0	0.727	41	YES	3	 0, 8[[0, 32[*
4		7	147	76	0	0	39.4	0.257	43	YES	4	0, 8[[128, 160[*
5		7	133	84	0	0	40.2	0.696	37	NO	5	 0, 8[[128, 160[*
6		7	159	64	0	0	27.4	0.294	40	NO	6	 0, 8[[128, 160[*

Nous pouvons importer ces données dans WEKA. Nous voyons que les données sont désormais moins précises :

Current relation	Selected attr	ribute		
Relation: pima-anon Attributes: 9 Instances: 768 Sum of weights: 768	Name: Glucose Missing: 0 (0%) Distinct: 5			Type: Nominal Unique: 0 (0%)
Attributes	No.	Label	Count	Weight
All None Invert Pattern No. Name Image: Pregnancy 1 Pregnancy 2 Glucose 3 BP 4 TricepsThickness 5 Insulin 6 BMI 7 DiabetesPedigree 8 Age	Class: Class	(Nom)	9 251 13 462 33	9.0 251.0 13.0 462.0 33.0
9 Class	9	251	13	492

Q8- QUESTION : Effectuez les tâches de classification et de clustering avec les paramètres optimaux que vous avez déterminés dans la Partie I. Commentez la qualité des résultats, en particulier les règles générées avec JRIP ou encore l'arbre de décision généré avec J48.

Q9- QUESTION : Choisissez un classifieur. Comparez la qualité de ce classifieur en utilisant plusieurs sortes d'anonymisation : k=4, k=20, des taux de suppression allant jusqu'à 25%, et en privilégiant certains attributs par rapport à d'autres. Commentez la possibilité d'utiliser l'anonymat pour faire de l'analyse de données.