Université de Bouira
Département d'Informatique
Module : Fouille de Données
Niveau M1-ISIL

Série TP 4

KNN et Naive Bayes.

//TODO

- Data Mining Challenge.
- Chaque groupe est une startup. *Business* : Data Science, Big Data, ventes prédictives, recommandations pour le e-commerce, enchères dynamiques, stratégies commerciales, veilles tarifaires, analyse des données remontées des réseaux sociaux ou des objets connectés, etc.
- Le but est de fournir différents modèles de décision, les plus précis possible, à une banque voulant mener une campagne Marketing sur ses clients afin de promouvoir sa nouvelle plateforme web.

Enoncé

Une banque dispose des informations suivantes sur un ensemble de clients :

Client	M	A	R	E	I
00	moyen	adulte	village	oui	oui
01	élevé	adulte	commune	non	non
02	faible	âgé	commune	non	non
03	faible	adulte	commune	oui	oui
04	moyen	jeune	ville	oui	oui
05	élevé	âgé	ville	oui	non
06	moyen	âgé	ville	oui	non
07	faible	adulte	village	non	non

L'attribut client indique le numéro du client ; l'attribut **M** indique la moyenne des crédits sur le compte du client ; l'attribut **A** donne la tranche d'âge ; l'attribut **R** décrit la localité du client ; l'attribut **E** possède la valeur oui si le client possède un niveau d'études supérieur au bac ; l'attribut **I** (la classe) indique si le client exécute ses opérations de gestion de compte via Internet.

Règles d'association - Apriori

1. Calculer les règles solides correspondant à un seuil de confiance = 0.9 et un minsup=0.4.
2. En déduire un modèle de décision.

Arbre de décision – J48

1. Construire l'arbre de décision correspondant à cette base en utilisant l'algorithme J48.
2. Donner la précision de l'arbre construit sur la base de test suivante :

Client	M	A	R	E	I
01	moyen	âgé	village	oui	non
02	élevé	jeune	ville	non	oui
03	faible	âgé	village	non	non
04	moyen	adulte	commune	oui	non

3. Depuis les résultats, indiquer sur quelle classe le modèle est moins performant (i.e. a plus tendance de se tromper). Justifier.
4. Donner la précision de l'arbre construit avec une évaluation par validation croisée.
5. Représenter l'arbre obtenu sous une forme plus interprétable pour un être humain.

Classification bayésienne - NaiveBayes

1. Donner le modèle de décision déduit de cette base en utilisant la classification naïve bayésienne. (Weka Explorer : Classify = > Choose => Bayes => **NaiveBayes**).
2. Donner la précision de l'arbre construit sur la base de test précédente.
3. Donner la précision du modèle déduit avec une évaluation par validation croisée.

Les K plus proches voisins - IBk

1. Convertir les attributs catégoriels en numérique en utilisant le filtre NominalToBinary. Preprocess => Filter => Choose => Unsupervised = > Attributes => NominalToBinary.
2. Entrainement : Classify = > Choose => Lazy => **IBk**. Changer dans les paramètres de IBk la valeur de **K**NN à 5.
3. Donner la précision sur la base de test précédente (à convertir aussi en numérique).
4. Donner la précision avec une évaluation par validation croisée.

Utilisation/Classification

1. Trouver la classe de ses nouveaux exemples : - selon chacun des modèles, puis par vote

Exemple	Apriori	J48	NaiveBayes	IBk	Décision
Elevé, Adulte, Village, Oui					
Faible, Jeune, Ville, Non					

Naive Bayes – Explication du modèle de décision obtenu (**Conditional Probability Table**) :

```
                Class
Attribute         yes    no
                (0.4)  (0.6)
============================
M
  faible          2.0    3.0
  moyen           3.0    2.0
  eleve           1.0    3.0
  [total]         6.0    8.0

A
  jeune           2.0    1.0
  adulte          3.0    3.0
  age             1.0    4.0
  [total]         6.0    8.0

R
  ville           2.0    3.0
  village         2.0    2.0
  commune         2.0    3.0
  [total]         6.0    8.0

E
  yes             4.0    3.0
  no              1.0    4.0
  [total]         5.0    7.0
```

Probabilité a priori : P(C)

Exemple : P(Yes) = 0.4

Pour le calcul des probabilités de vraisemblance.

= (Nombre d'instances avec une valeur donné d'un attribut dans le dataset par rapport à chaque classe) + 1.

Exemple : R=ville 2.0(yes) 3.0(no)

2.0 : (nombre d'instances ayant la valeur ville et leur classe = yes) + 1 = (1) +1 = 2.0

3.0 : (nombre d'instances ayant la valeur ville et leur classe = no) + 1 = (2) +1 = 3.0

Donc : P(X=ville|C=yes) = 2.0/Total = 2.0/6.0= 0.33

Client	M	A	R	E	I
00	moyen	adulte	village	oui	oui
01	élevé	adulte	commune	non	non
02	faible	âgé	commune	non	non
03	faible	adulte	commune	oui	oui
04	moyen	jeune	ville	oui	oui
05	élevé	âgé	ville	oui	non
06	moyen	âgé	ville	oui	non
07	faible	adulte	village	non	non