TP 1 : Prétraitement et Visualisation des données avec Weka

Ce TP est à réaliser sur Linux

Partie I : **Weka : Présentation et installation**

1. Présentation
   Weka (Waikato Environment for Knowledge Analysis) est un ensemble d'outils permettant de manipuler et d'analyser des fichiers de données, implémentant la plupart des algorithmes d'intelligence artificielle, dont les arbres de décision et les réseaux de neurones. Il est écrit en java, disponible sur le web [1], et s'appuie sur le livre
   `Data Mining, practical machine learning tools and techniques with Java implementations`
   `Witten & Frank`
   `Editeur : Morgan Kauffman`

   Il se compose principalement :
   — De classes Java permettant de charger et de manipuler les données.
   — De classes pour les principaux algorithmes de classification supervisée ou non supervisée.
   — D'outils de sélection d'attributs, de statistiques sur ces attributs.
   — De classes permettant de visualiser les résultats.
   On peut l'utiliser à trois niveaux :
   — Via l'interface graphique, pour charger un fichier de données, lui appliquer un algorithme, vérifier son efficacité.
   — Invoquer un algorithme sur la ligne de commande.
   — Utiliser les classes définies dans ses propres programmes pour créer d'autres méthodes, implémenter d'autres algorithmes, comparer ou combiner plusieurs méthodes.
2. Installation, initialisation
   Weka est installé dans les salles TP. Si vous voulez l'employer chez vous :
   — Charger l'archive zip à partir du site de Weka
   — Décompressez-la.
   — C'est tout.. . .
   — Les classes sont dans weka.jar
   — Les sources dans weka-src.jar
   — WekaManual.pdf est une présentation assez poussée des fonctionnalités de Weka.
3. Interface graphique
   Elle se lance avec la commande **java -jar** $\$WEKAHOME/weka.jar$
   Après l'avoir lancé, vous obtenez la fenêtre intitulée Weka GUI Chooser : choisissez l'Explorer. La nouvelle fenêtre qui s'ouvre alors (Weka Knowledge Explorer) présente six onglets (vous pouvez y ajouter des onglets en installant d'autres packages via l'outil package manager) :
   — **Preprocess** : pour choisir un fichier, inspecter et préparer les données.
   — **Classify** : pour choisir, appliquer et tester différents algorithmes de classification : là, il s'agit d'algorithmes de classification supervisée.
   — **Cluster** : pour choisir, appliquer et tester les algorithmes de segmentation.
   — **Associate** : pour appliquer l'algorithme de génération de règles d'association.
   — **Select Attributes** : pour choisir les attributs les plus prometteurs.
   — **Visualize** : pour afficher (en deux dimensions) certains attributs en fonctions d'autres.
4. Fichier **ARFF**
   WEKA utilise (entre autres) le format de fichier arff pour enregistrer les données. Un fichier arff est composé d'une liste d'exemples définis par leurs valeurs d'attributs.
   Un fichier arff comprend toujours trois types d'informations :
   — le nom de la relation : @relation permet de donner un nom à la relation
   — la liste des noms d'attributs et du type de valeurs : @attribute permet de définir un attribut
   — la liste des instances : @DATA permet de définir les données. chaque ligne représente une description, par la liste des valeurs de chacun de ses attributs. Une valeur manquante est remplacée par un point d'interrogation.
   — le caractère « % » marque les lignes de commentaires.

1. www.cs.waikato.ac.nz/ml/weka/

5. On peut aussi charger un fichier au format CSV (Open File), ou encore des données à partir d'une requête SQL (Open DB), ou sur le web (Open URL).

Partie II : **Prétraitement et Visualisation des données**

Exercice 1 :

Informations sur les prêts Japonais (un échantillon à partir de l'historique de prêt d'une base de données d'une banque japonaise) :

Clients : s1,..., s20

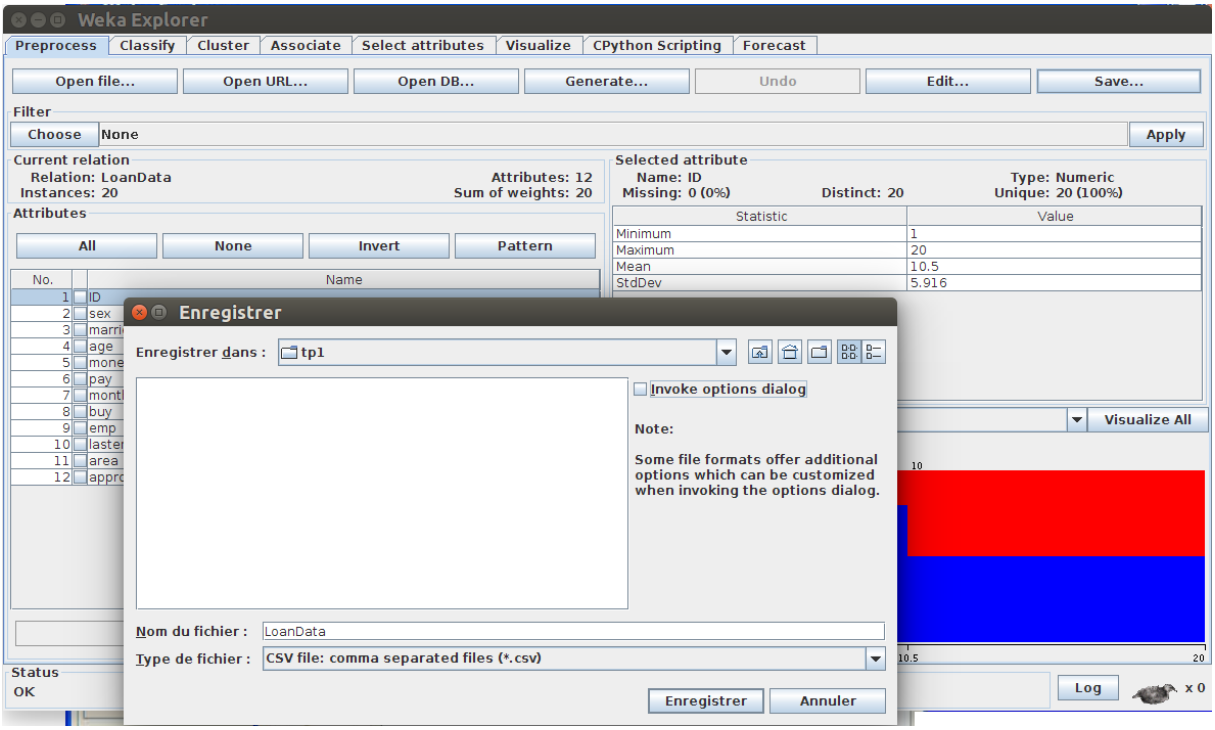
— Approved loan : s1, s2, s4, s5, s6, s7, s8, s9, s14, s15, s17, s18, s19
— Rejected loan : s3, s10, s11, s12, s13, s16, s20

Clients data :

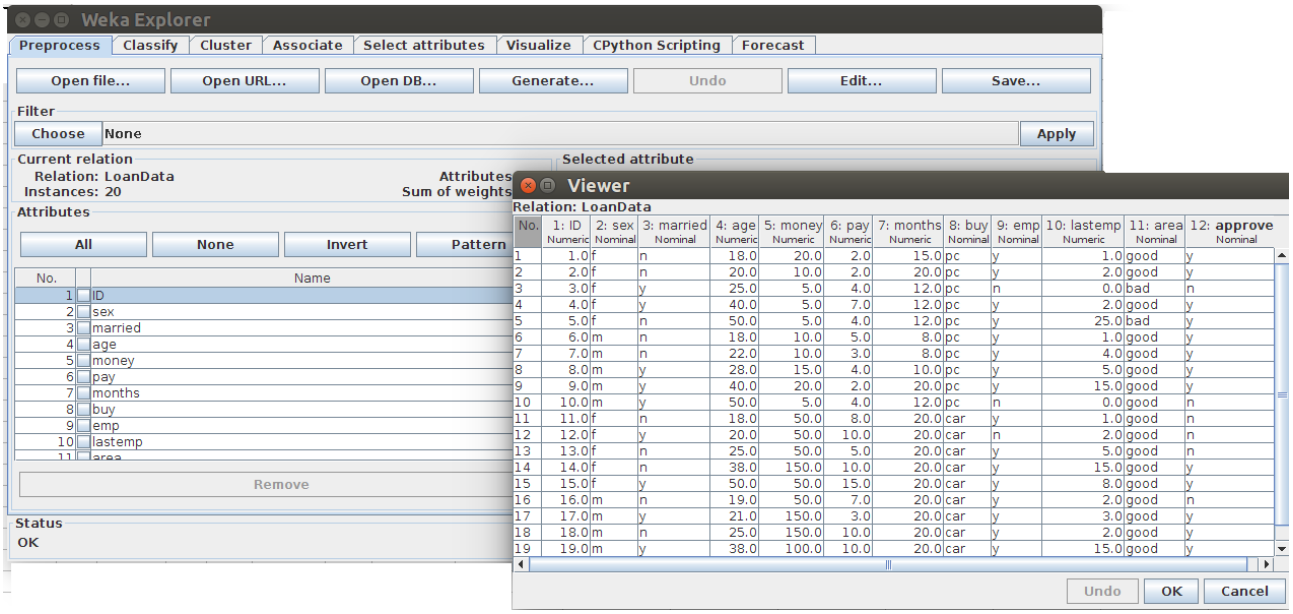
— unemployed clients : s3, s10, s12
— loan is to buy a personal computer : s1, s2, s3, s4, s5, s6, s7, s8, s9, s10
— loan is to buy a car : s11, s12, s13, s14, s15, s16, s17, s18, s19, s20
— male clients : s6, s7, s8, s9, s10, s16, s17, s18, s19, s20
— not married : s1, s2, s5, s6, s7, s11, s13, s14, s16, s18
— live in problematic area : s3, s5
— age : s1=18, s2=20, s3=25, s4=40, s5=50, s6=18, s7=22, s8=28, s9=40, s10=50, s11=18, s12=20, s13=25, s14=38, s15=50, s16=19, s17=21, s18=25, s19=38, s20=50
— money in a bank (x10000 yen) : s1=20, s2=10, s3=5, s4=5, s5=5, s6=10, s7=10, s8=15, s9=20, s10=5, s11=50, s12=50, s13=50, s14=150, s15=50, s16=50, s17=150, s18=150, s19=100, s20=50
— monthly pay (x10000 yen) : s1=2, s2=2, s3=4, s4=7, s5=4, s6=5, s7=3, s8=4, s9=2, s10=4, s11=8, s12=10, s13=5, s14=10, s15=15, s16=7, s17=3, s18=10, s19=10, s20=10
— months for the loan : s1=15, s2=20, s3=12, s4=12, s5=12, s6=8, s7=8, s8=10, s9=20, s10=12, s11=20, s12=20, s13=20, s14=20, s15=20, s16=20, s17=20, s18=20, s19=20, s20=30
— years with the last employer : s1=1, s2=2, s3=0, s4=2, s5=25, s6=1, s7=4, s8=5, s9=15, s10=0, s11=1, s12=2, s13=5, s14=15, s15=8, s16=2, s17=3, s18=2, s19=15, s20=2

Q 1 . Construire le fichier ARFF correspondant avec pour nom "LoanData". (Attribute-Relation File Format - http ://www.cs.waikato.ac.nz/˜ml/weka/arff.html)

Q 2 . Ouvrir le fichier avec Weka et le convertir en un fichier CSV (Comma-Separated Values).



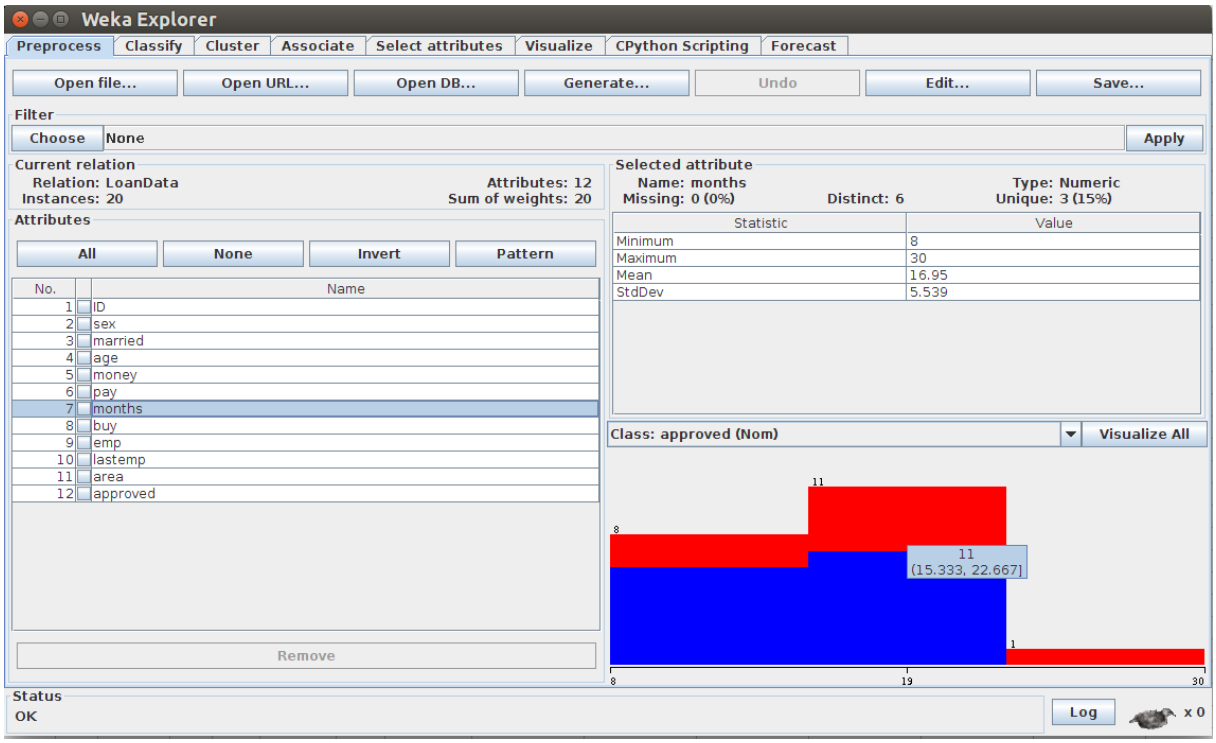
Q 3 . Editer le fichier.

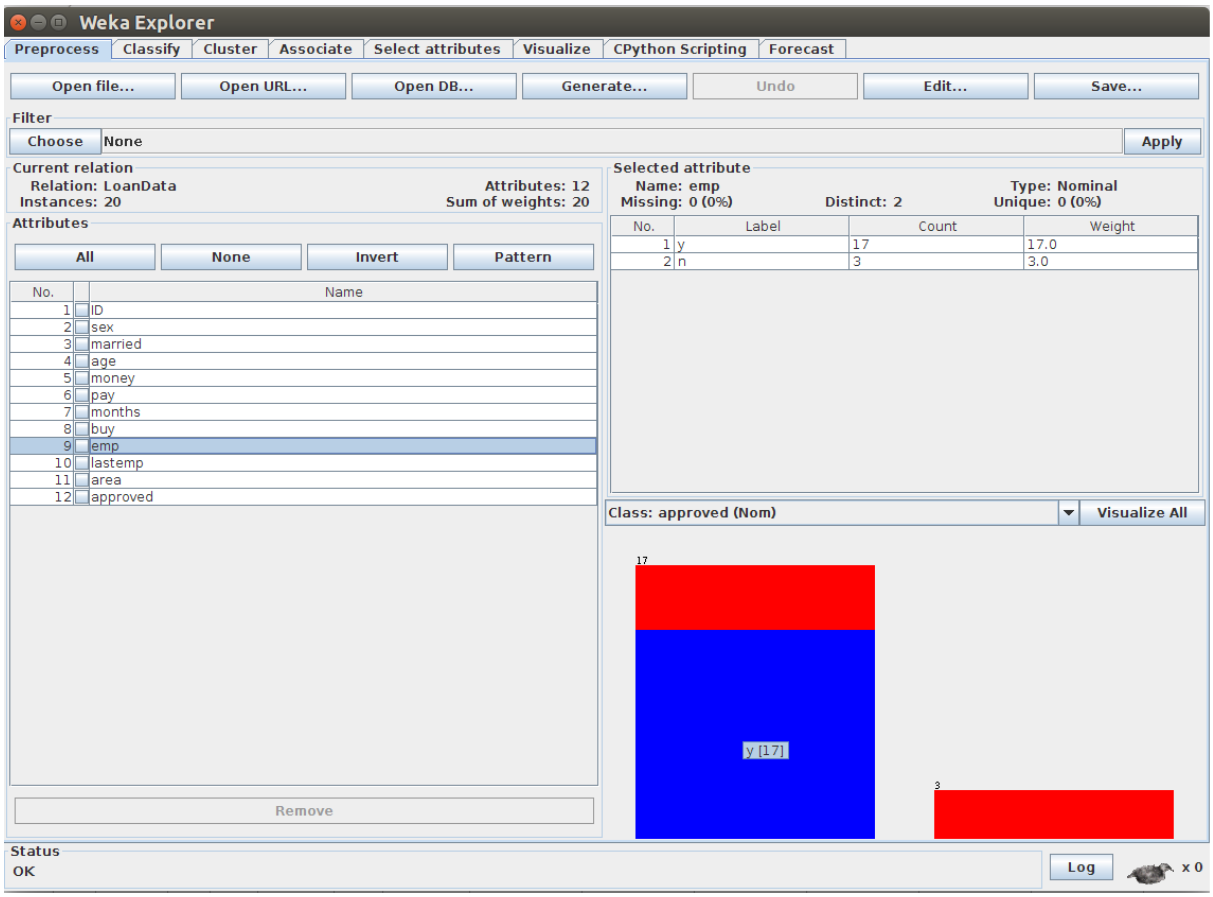


Q 4 . Examiner et visualiser les données

— Type et propriétés des attributs

— Distribution par rapport aux classes





Exercice 2 :

Q 1 . Télécharger le fichier doc_string.arff du répertoire TP sur l'ENT.

Q 2 . Visualiser doc_string.arff à l'aide d'un éditeur et l'examiner.

Q 3 . Ouvrir le fichier à l'aide de Weka et convertir le premier et le dernier attributs en attributs nominaux (filters/unsupervised/attribute/StringToNominal).

Q 4 . Convertir le texte contenu dans le deuxième attribut en un vecteur de mots. Chaque mot devient un attribut numérique. (filters/unsupervised/attribute/StringToWordVector)

Q 5 . Remettre l'attribut classe en fin de la liste des attributs :
— filters/unsupervised/attribute/Copy
— Mettre l'index à 2
— Supprimer le deuxième attribut

Q 6 . Changer le format numérique des attributs en format binaire (filters/unsupervised/attribute/NumericToBinary) et sauvegarder les données dans un fichier.

Q 7 . Sauvegarder une autre version sparse du fichier (filters/unsupervised/instance/NonSparseToSparse)