Faculté des Sciences Economiques et de Gestion de Sfax

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique Université de Sfax

Projet Fouille de Données

Master de Recherche : Système d'Information et Nouvelles Technologies

Sujet :

Crédit allemand

Réalisé par :

CHOUCHENE BOUHMILA Khawla

Proposé par :

Mme. Lamia Hadrich Belguith

Année universitaire : 2018-2019

Introduction

Les banques analysent une myriade de critères avant d'accorder un crédit. Celles-ci vont de l'âge, du montant et de l'objet du crédit, de la profession, du statut d'emploi et bien d'autres choses encore. Ils utilisent également des algorithmes sophistiqués de régression et d'apprentissage automatique logistique pour prédire si le demandeur de crédit est crédible ou non.

L'objectif de ce projet est de produire une application interactive brillante dont l'utilisation est double:

Tout d'abord par les directeurs de banque pour obtenir un aperçu visuel préliminaire de la solvabilité d'un demandeur de crédit lors de l'analyse d'une demande de crédit. Étant donné que leurs modèles prédictifs exigent beaucoup de ressources et qu'ils peuvent durer longtemps, cette visualisation préliminaire peut être utile pour éliminer les demandes carrément insatisfaisantes avant d'exécuter les modèles sur le reste.

Deuxièmement, par les demandeurs de crédit pour estimer leur solvabilité avant de faire officiellement une demande à leurs banques. Cela les aidera à adapter leur demande de crédit en conséquence ou à la reporter jusqu'à ce qu'ils obtiennent une cote de crédibilité favorable.

L'ensemble de données utilisé pour ce projet a été obtenu à partir du référentiel UCI ML (ensemble de données GCD). Cet ensemble de données a été donné au public par le professeur Hans Hofmann de l'Université de Hambourg en 1994. Obtenir des données de crédit à jour est apparemment impossible, car cette information est très précieuse pour toute banque et la rendre publique donnera un avantage commercial.

L'ensemble de données contient 1000 observations avec 20 variables dont 14 sont catégoriques et une colonne supplémentaire classant les candidats comme crédibles ou non crédibles.

Présentation de Dataset

Les données contiennent 1000 entrées avec 21 attributs catégorique(symboliques).

Dans cet ensemble de données, chaque entrée représente une personne qui reçoit un crédit d'une banque. Chaque personne est classée en risques de crédit bons ou mauvais en fonction de l'ensemble des attributs.

Les attributs sont :

Statut du compte courant existant (qualitatif)

A11 : ... < 0 DM

A12 : 0 <= ... < 200 DM

A13 : ... >= 200 DM / Affectations salariales pour au moins 1 an

A14 : pas de compte courant

Durée en mois (numérique)

Antécédents de crédit (qualitatifs)

A30 : aucun crédit pris/ tous les crédits dûment remboursés

A31 : tous les crédits de cette banque dûment remboursés

A32 : crédits existants dûment remboursés jusqu'à présent

A33 : retard de remboursement dans le passé

A34 : compte critique / autres crédits existants (pas dans cette banque)

Objectif (qualitatif)

A40 : voiture (neuve)

A41 : voiture (occasion)

A42 : mobilier/équipement

A43 : radio/télévision

A44 : appareils ménagers

A45 : réparations

A46 : éducation

A47 : (vacances - n'existe pas ?)

A48 : recyclage professionnel

A49 : affaires

A410 : autres

Montant du crédit (numérique)

Compte d'épargne/obligations (qualitatif)

A61 : .... < 100 DM

A62 : 100 <= ... < 500 DM

A63 : 500 <= ... < 1000 DM

A64 : >= 1000 DM

A65 : inconnu/ pas de compte d'épargne

Emploi actuel depuis (qualitatif)

A71 : chômeurs

A72 : ... < 1 an

A73 : 1 <= ... < 4 ans

A74 : 4 <= ... < 7 ans

A75 : 7 ans >= 7 ans

Taux d'acomptes provisionnels en pourcentage du revenu disponible (numérique)

Statut personnel et sexe (qualitatif)

A91 : homme : divorcé/séparé

A92 : femme : divorcée/séparée/mariée

A93 : mâle : simple

A94 : hommes : mariés/veufs

A95 : femelle : simple

Autres débiteurs / garants (qualitatifs)

A101 : aucun

L102 : codemandeur

A103 : garant

Résidence actuelle depuis (numérique)

Immobilier (qualitatif)

A121 : biens immobiliers

A122 : sinon A121 : contrat d'épargne-logement / assurance-vie

A123 : sinon A121/A122 : voiture ou autre, hors attribut 6

A124 : inconnu / pas de propriété

Âge en années (numérique)

Autres plans de paiements échelonnés (qualitatifs)

A141 : banque

A142 : magasins

A143 : aucun

Logement (qualitatif)

A151 : loyer

A152 : propre

A153 : gratuit

Nombre de crédits existants dans cette banque (numérique)

Emploi (qualitatif)

A171 : chômeur/non qualifié - non-résident

A172 : non qualifié - résident

A173 : employé qualifié / fonctionnaire

L174 : cadres, travailleurs autonomes, travailleurs autonomes

Employé/agent hautement qualifié

Nombre de personnes tenues d'assurer l'entretien de (numérique)

Téléphone (qualitatif)

A191 : aucune

A192 : oui, enregistré sous le nom du client

Travailleur étranger (qualitatif)

A201 : oui

A202 : non

Classe : (crédible ou non)

A211 : good

A212 : bad

Vous trouverez ci-dessous le résumé de certaines des variables de l'ensemble de données :

Les algorithmes

Le k-NN est le diminutif de **k Nearest Neighbors**. C'est un algorithme qui peut servir autant pour la classification que la régression. Il est surnommé « nearest neighbors » (plus proches voisins en français) car le principe de ce modèle consiste en effet à choisir les **k**données les plus proches du point étudié afin d'en prédire sa valeur.

L'algorithme K-NN (K-nearest neighbors) est une méthode d'apprentissage supervisé. Il peut être utilisé aussi bien pour la régression que pour la classification. Son fonctionnement peut être assimilé à l'analogie suivante *"dis-moi qui sont tes voisins, je te dirais qui tu es…"*.

Pour effectuer une prédiction, l'algorithme K-NN ne va pas calculer un modèle prédictif à partir d'un *Training Set* comme c'est le cas pour la régression logistique ou la régression linéaire. En effet, K-NN **n'a pas besoin de construire un modèle prédictif.** Ainsi, pour K-NN il n'existe pas de phase d'apprentissage proprement dite. C'est pour cela qu'on le catégorise parfois dans le Lazy Learning. Pour pouvoir effectuer une prédiction, K-NN **se base sur le jeu de données** pour produire un résultat.

Comment K-NN effectue une prédiction ?

Pour effectuer une prédiction, l'algorithme K-NN va **se baser sur le jeu de données en entier**. En effet, pour une observation, qui ne fait pas parti du jeu de données, qu'on souhaite prédire, l'algorithme va chercher les **K instances du jeu de données les plus proches de notre observation.** Ensuite pour ces voisins, l'algorithme se basera sur leurs variables de sortie (output variable) pour calculer la valeur de la variable de l'observation qu'on souhaite prédire.

Par ailleurs :

Si K-NN est utilisé pour la régression, c'est la moyenne (ou la médiane) des variables des plus proches observations qui servira pour la prédiction

Si K-NN est utilisé pour la classification, c'est le mode des variables des plus proches observations qui servira pour la prédiction

On peut schématiser le fonctionnement de K-NN en l'écrivant en pseudo-code suivant :

Début Algorithme

Données en entrée :

Un ensemble de données .

Une fonction de définition distance .

Un nombre entier

Pour une nouvelle observation dont on veut prédire sa variable de sortie Faire :

Calculer toutes les distances de cette observation avec les autres observations du jeu de données

Retenir les observations du jeu de données les proches d'en utilisation la fonction de calcul de distance

Prendre les valeurs de des observations retenues :

Si on effectue une régression, calculer la moyenne (ou la médiane) de retenues

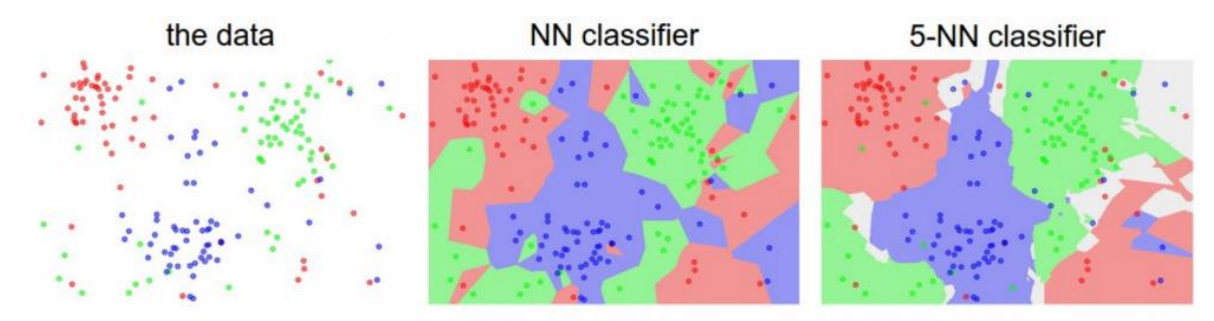
Si on effectue une classification, calculer le mode de retenues

Retourner la valeur calculée dans l'étape 3 comme étant la valeur qui a été prédite par K-NN pour l'observation .

Fin Algorithme

Comment choisir la valeur K ?

Le choix de la valeur à utiliser pour effectuer une prédiction avec K-NN, varie en fonction du jeu de données. En règle générale, moins on utilisera de voisins (un nombre petit) plus on sera sujette au sous apprentissage (underfitting). Par ailleurs, plus on utilise de voisins (un nombre K grand) plus, sera fiable dans notre prédiction. Toutefois, si on utilise ombre de voisins avec et étant le nombre d'observations, on risque d'avoir du overfitting et par conséquent un modèle qui se généralise mal sur des observations qu'il n'a pas encore vu.



L'image ci-dessus à gauche représente des points dans un plan 2D avec trois types d'étiquetages possibles (rouge, vert, bleu). Pour le 5-NN classifieur, les limites entre chaque région sont assez lisses et régulières. Quant au N-NN Classifier, on remarque que les limites sont "chaotiques" et irrégulières. Cette dernière provient du fait que l'algorithme tente de faire rentrer tous les points bleus dans les régions bleues, les rouges avec les rouges etc… c'est un cas d'overfitting.

Pour cet exemple, on préférera le 5-NN classifier sur le NN-Classifier. Car le 5-NN classifier se généralise mieux que son opposant.

Limitations de K-NN

K-NN est un algorithme assez simple à appréhender. Principalement, grâce au fait qu'il n'a pas besoin de modèle pour pouvoir effectuer une prédiction. Le contre coût est qu'il doit **garder en mémoire l'ensemble des observations** pour pouvoir effectuer sa prédiction. Ainsi il faut faire attention à **la taille du jeu d'entrainement.**

Également, le choix de la méthode de calcul de la distance ainsi que le nombre de voisins peut ne pas être évident. Il faut essayer plusieurs combinaisons et faire du tuning de l'algorithme pour avoir un résultat satisfaisant.

KStar est un classificateur basé instance qui classifie une instance de test selon la classe des instances, qui lui sont semblables, et ce en utilisant une fonction de similitude (Cleary, 1995). Il diffère des autres algorithmes d'apprentissage basés instance parce qu'il emploie une fonction de distance basée sur l'entropie. L'utilisation de l'entropie (Witten, 2000) comme mesure de distance a plusieurs avantages. Elle fournit une approche cohérente à la manipulation des attributs symboliques, des attributs à valeurs réelles et des valeurs manquantes. En utilisant une telle mesure, K* fournit des résultats rivalisant favorablement avec plusieurs algorithmes d'apprentissage automatique.

Les outils utilisés

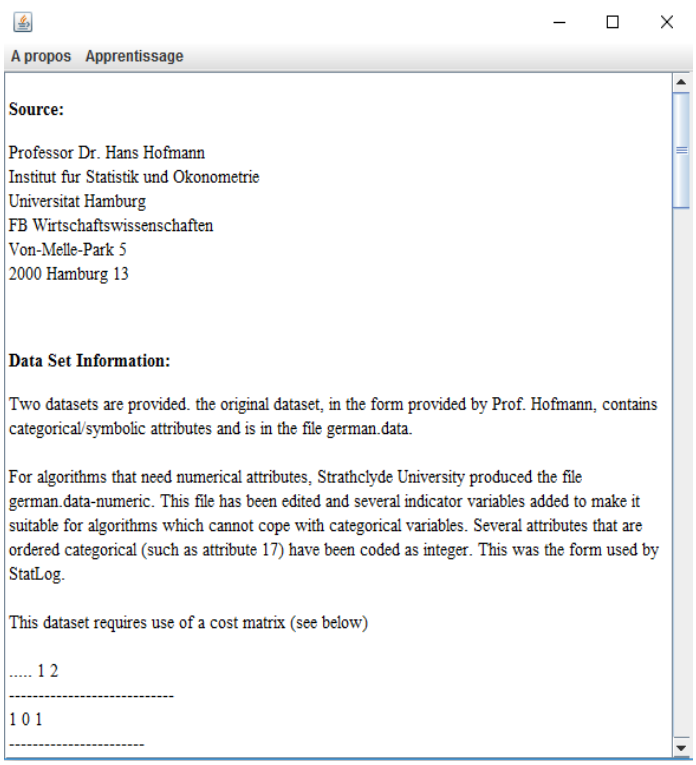
- **WEKA 3.8 :** Weka est une suite populaire de logiciels d'apprentissage automatique. Écrite en Java, développée à l'université de Waikato, Nouvelle-Zélande. Weka est un Logiciel libre disponible sous la Licence publique générale GNU.
- **PYTHON 3.**7 : Python est un langage de programmation interprété, multiparadigme et multiplateformes. Il favorise la impérative structurée, fonctionnelle et orientée objet
- **JAVA :** Java est un langage de programmation orienté objet créé par James Gosling et Patrick Naughton. La particularité et l'objectif central de Java est que les logiciels écrits dans ce langage doivent être très facilement portables sur plusieurs systèmes d'exploitation.

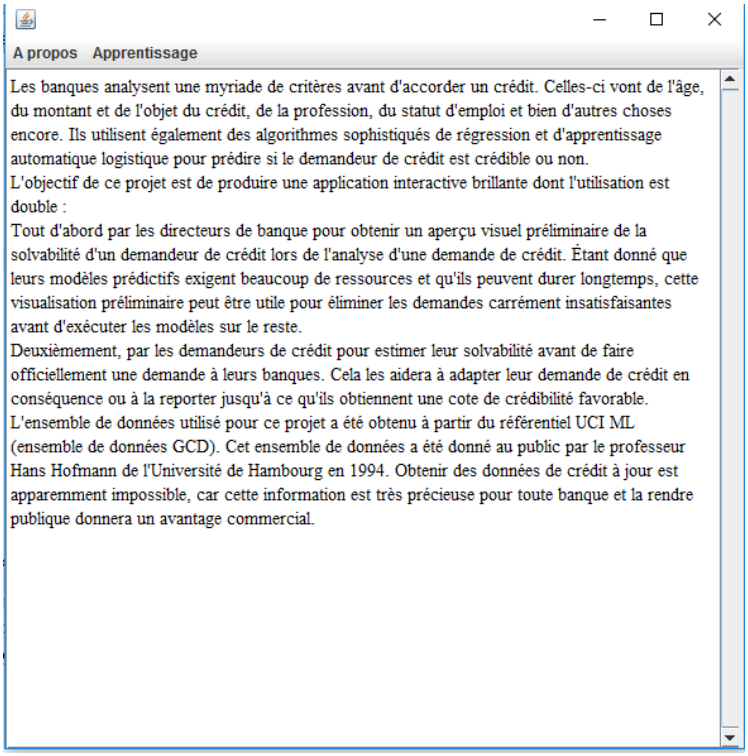
Application

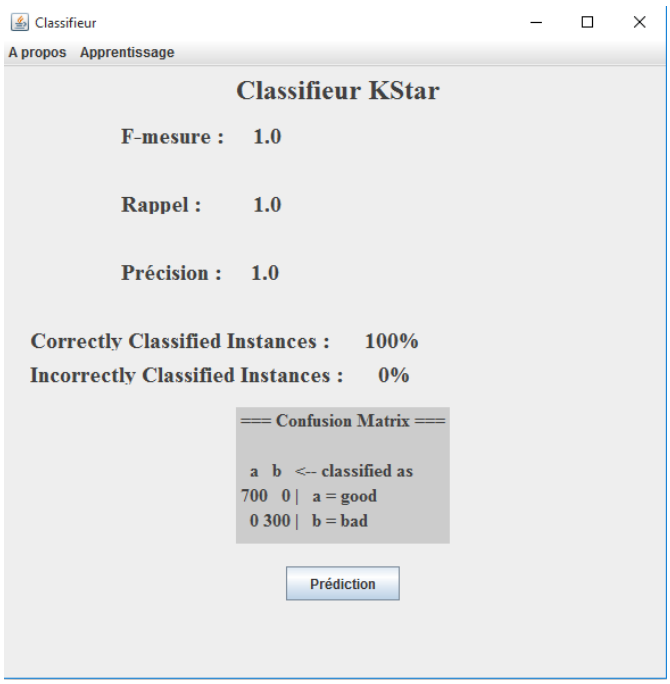
Mon application consiste en première étape à faire l'apprentissage de données en utilisant les 2 algorithmes que j'est citée auparavant puis le test et les prédictions.

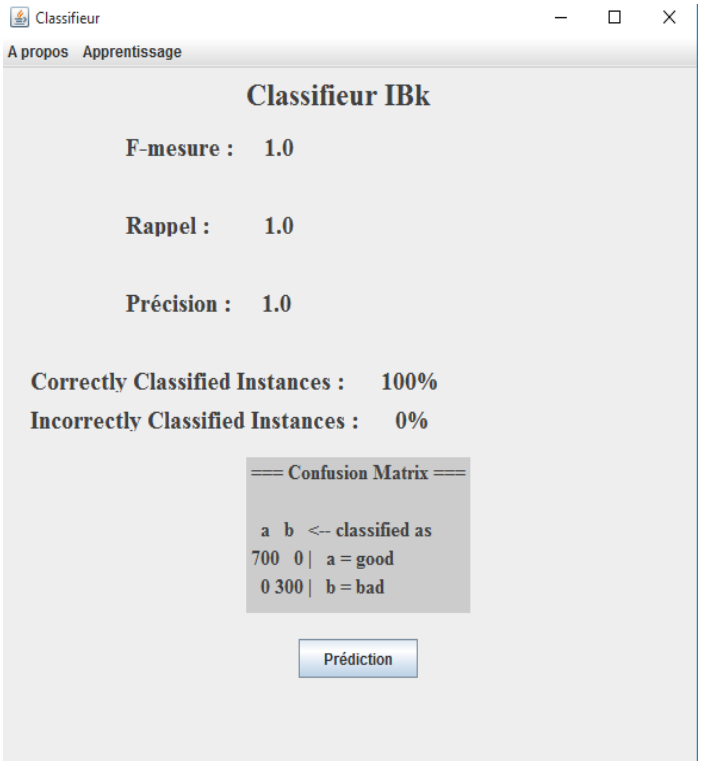
Version JAVA

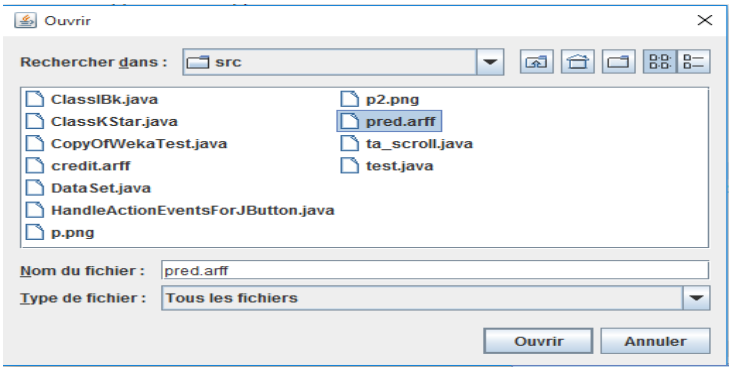
Pour découvrir en quoi consiste le projet il faut cliquer sur le menu « A propos -> projet » et pour les informations sur les données il faut cliquer sur le menu « A propos -> DataSet »

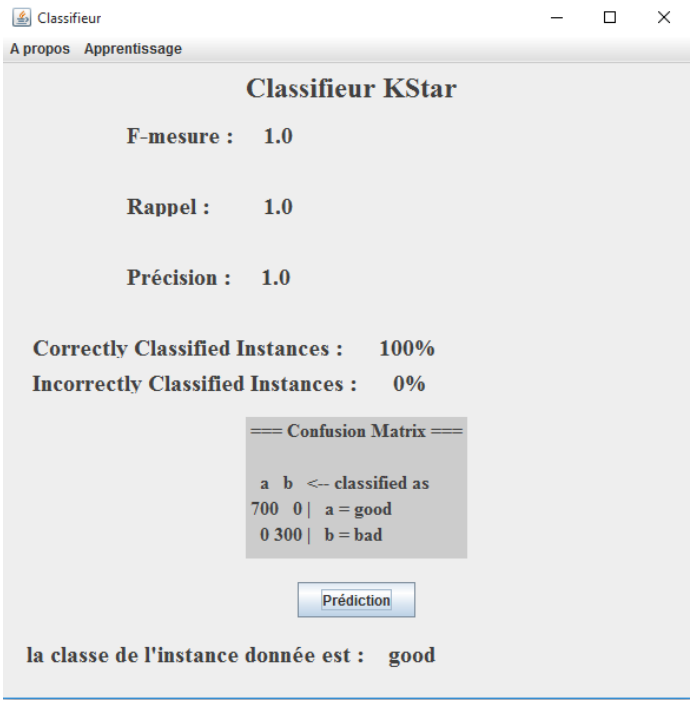


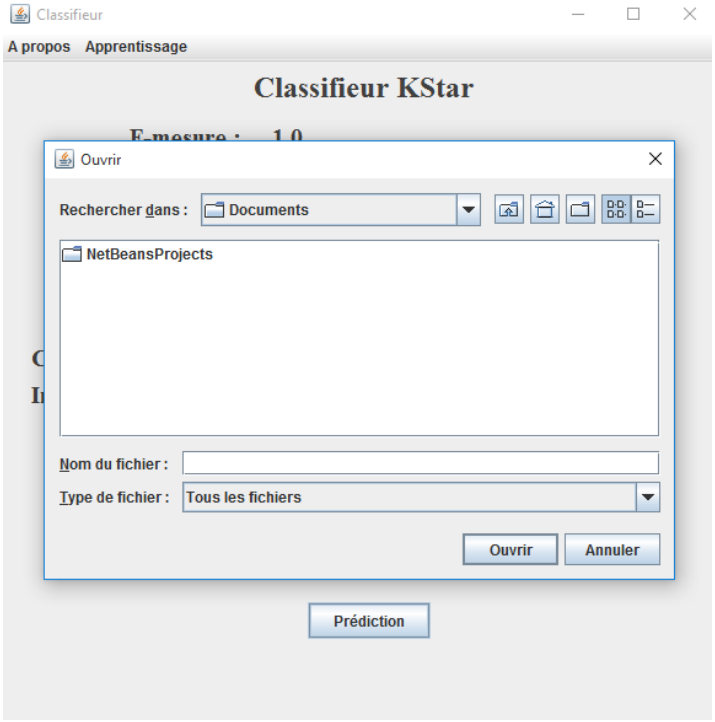




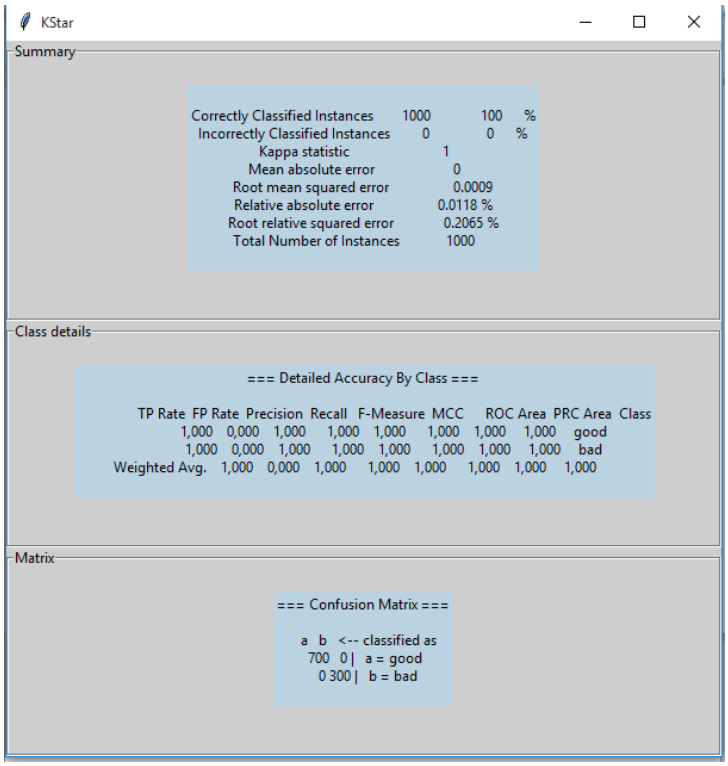

Classifieur
A propos Apprentissage
Classifieur IBk
F-mesure : 1.0
Rappel : 1.0
Précision : 1.0
Correctly Classified Instances : 100%
Incorrectly Classified Instances : 0%
=== Confusion Matrix ===
a b <-- classified as
700 0 | a = good
0 300 | b = bad
Prédiction

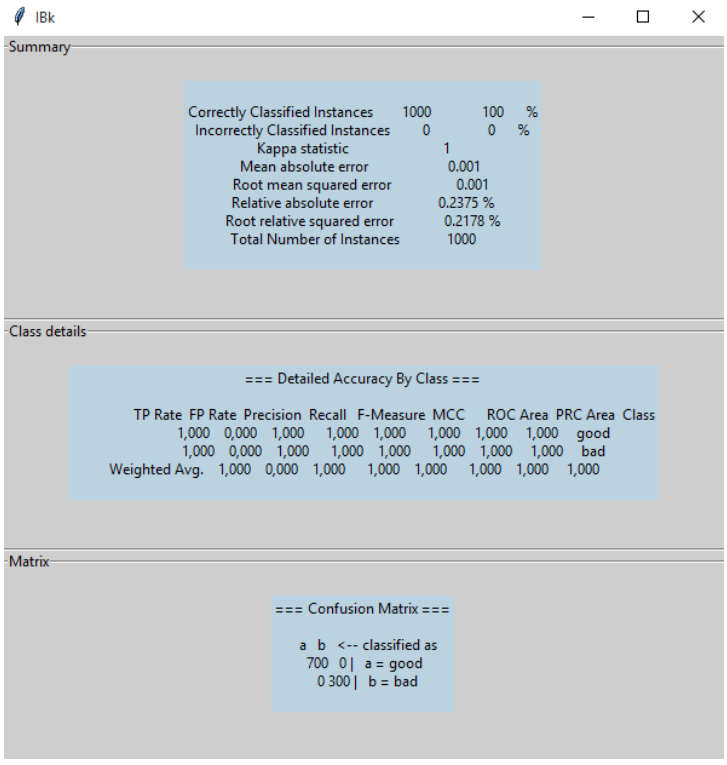

Ouvrir
Rechercher dans : src
ClassIBk.java
ClassKStar.java
CopyOfWekaTest.java
credit.arff
DataSet.java
HandleActionEventsForJButton.java
p.png
p2.png
pred.arff
ta_scroll.java
test.java
Nom du fichier : pred.arff
Type de fichier : Tous les fichiers
Ouvrir
Annuler


Classifieur
A propos Apprentissage
Classifieur KStar
F-mesure : 1.0
Rappel : 1.0
Précision : 1.0
Correctly Classified Instances : 100%
Incorrectly Classified Instances : 0%
=== Confusion Matrix ===
a b <-- classified as
700 0 | a = good
0 300 | b = bad
Prédiction
la classe de l'instance donnée est : good


Classifieur
A propos Apprentissage
Classifieur KStar
Ouvrir
Rechercher dans : Documents
NetBeansProjects
Nom du fichier :
Type de fichier : Tous les fichiers
Ouvrir
Annuler
Prédiction

Version PYTHON





Conclusion

Le projet pourrait être étendu pour attribuer des pondérations aux diverses variables avant d'élaborer et de former un modèle prédictif qui simulera si un candidat est digne de foi ou non.

Les analyses de crédit modernes utilisent de nombreuses variables supplémentaires comme le casier judiciaire des demandeurs, leurs renseignements médicaux, le solde net entre le revenu mensuel et les dépenses. Un ensemble de données contenant ces variables pourrait être acquis ou des variables complémentaires pourraient être ajoutées à l'ensemble de données. Cela rendra les simulations de crédit beaucoup plus réalistes, semblables à ce qui est fait par les banques avant qu'un crédit ne soit approuvé.