Université Abdelhamid Ibn Badis – Mostaganem
Faculté des Sciences Exactes et d'Informatique
Département de Mathématiques et informatique
Filière Informatique
Master 1 option Ingénierie des systèmes d'information

UNIVERSITE
Abdelhamid Ibn Badis
MOSTAGANEM

Module : fouille de données

Rapport de projet de fouille de données

Travaille réalisé par :

BELHAMITI AYA (ISI)
FATAH MAROUA (ISI)
GANI FATIMA (ISI)
FERHI ASMAA (ISI)
BELBAHI ILYES (ISI)
DJEZZAM MOHAMED AMINE AISSA (RESYS)

2021 / 2022

Introduction :

Dans ce projet nous avons examiné un recensement de revenus réalisé aux États-Unis en 1994. Le but Étant de prédire si le revenu annuel d'un individu sera plus que 50.000$ en se basant sur plusieurs attributs dataset est composé de 14 variables prédictives (colonnes) et d'une variable à prédire (classe) ainsi que de plus de 32000 enregistrements.

Ce travail sera réalisé sous langage PYTHON.

Le travail et disponible sur le lien :

https://github.com/asmaa-aa/Workshop-test

Les différentes étapes réalisées:

Étape 1 : Exploration du dataset :

L'importation des bibliothèques nécessaires :

- numpy --> pour le calcul scientifique.
- pandas --> pour les structures de données et les outils d'analyse de données.
- matplotlib --> pour outils de visualisation de données.

Affichage des dataset /chargement :

Pour la lecture d'un fichier csv on utilise 'pandas.read_csv'

Affichage de nombre des lignes et colonnes de cette data set par 'data.shape'

On voie que set dataset contient 15 attribut, 32560 individus

Quelques informations sur notre dataset :

Ensemble de données sur des observations concernant des différents paramètres financiers pour des adultes : âge, sexe, état civil, pays, revenu, éducation, profession, gain en capital, et d'autres paramètres défini par suites (attributs).

Les attributs sont :

- Age : l'Age de l'individu, valeur continue positive.
- workclass : la classe d'œuvre, les valeurs possibles : Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
- fnlwgt : le poids final, qui est le nombre d'unités dans la population cible que l'unité répondante représente, valeur continue positive.
- education : le niveau d'études atteint par l'individu, les valeurs possibles: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.
- education_num : le nombre total d'années d'études, qui est une représentation continue de la variable discrète éducation.
- marital-status : l'état civil, les valeurs possibles : Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.

- occupation : la profession, les valeurs possibles : Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.

- relationship : le rôle de l'unité répondante dans la famille, les valeurs possibles : Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.
- race : la race de l'individu, les valeurs possibles: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.
- sex : Female, Male.
- capital_gai / capital_loss : le revenus provenant de sources d'investissement autres que les salaires/salaires (gain /perte), valeur continue.
- hours_per_week : nombres des heures de travail par semaine, valeur continue positive.
- country : le pays , les valeurs possibles: United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands.
- income : le revenu, les valeurs possibles: >50K , <=50K .

Voire les attributs numérique par la command 'data**.**describe' pour le but de calculer des statistiques sur les données numérique

- le max = .max()
- le min = .min()
- la moyenne = . mean()
- l'écart-type de l'âge = .sdt()

On diviser les gens selon leur sexe en deux groupes : les hommes et les femmes. Apres on affiche le nombre d'individus selon chaque groupe qui gagnent plus et mois de 50.000$ et ceux qui gagnent moins ainsi que leurs pourcentages, ainsi le nuage de points montrant la distribution des enregistrements basent par plusieurs attribue choisies les quelles (age, eduction_num, occupation, workclass)

Et on a fait quelques Représentation de la distribution de ses attribues par un histogramme. Et d'autres représentations significatives.

- un graphique de barres empilées présentant la relation entre le nombre d'heures travaillées et le revenu.
- un graphique de barres empilées présentant la relation entre la race et le revenu
- un graphique de barres empilées présentant la relation entre le sexe et le revenu
- âpres on a passé vers autre niveau est le prétraitement des données.

Étape : Prétraitement du dataset :

Sélection d'attributs :

On a exécuté une boucle for sur toutes les colonnes en utilisant la fonction value_counts de la bibliothèque Pandas qui retourne le nombre des valeurs uniques, après ca on sélectionne les colonnes qui ont beaucoup de valeurs distinctes comme l'attribut fnlgwt qui a environ 2000+ valeurs. (Les attributs qui ont beaucoup de valeurs distinctes considéré comme attribut Noisy, doit être supprimé)

La fonction Pandas.drop est utilisée pour supprimer des colonnes ou des lignes spécifiées,(axis=1représente que nous avons l'intention de supprimer la colonne elle-même, inplace=True est de mentionner que nous remplaçons la trame de données d'origine).

Réduction de dimension :

Trouver la corrélation entre les attributs, fusionné ou supprimé les attributs qui ont une corrélation ≈ 1

Trouve le données manquantes/erronées :

Chercher les valeurs qui ont hors d'ensembles des valeurs possibles des attributs (garde que les valeurs possible)

Transformation des données :

Convertir les valeurs (attributs) nominales en valeurs numériques en utilisant la bibliothèque Pandas

Normalisation/Standardisation des données

On utilisant la formule de normalisation :

$$z = \frac{x \text{ - } \min(x)}{\max(x) \text{ - } \min(x)}$$

Séparation des données :

On a séparé les donne a 2 partie, 70% pour l'entrainement, et le reste est pour le teste , chaque partie est bien devise on 2 partie encore x qui sont les donnes , y est la classe a prédire .

Étape 3 : Prédiction et classification :

Régression logistique

La régression logistique est l'un des algorithmes d'apprentissage automatique supervisé les plus simples et les plus couramment utilisés pour la classification catégorielle. Les concepts fondamentaux de base de la régression logistique sont faciles à comprendre et peuvent être utilisés comme algorithme de base pour tout problème de classification binaire (0 ou 1), pour nous (<50, =>50)

- Créer un classifieur à base de l'algorithme Régression logistique
- Appliquer le classifieur sur l'ensemble de données
- appliquer le classifieur résultant sur les data test
- calculer la Précision
- la matrice de confusion pour test la qualite du modele ou classieur

Naïve Bayes

Naive Bayes est un algorithme de classification pour les problèmes de classification binaire (à deux classes) et multiclasses. On l'appelle Bayes naïf ou Bayes idiot parce que les calculs des probabilités pour chaque classe sont simplifiés pour rendre leurs calculs traitables.

Créer un classifieur à base de l'algorithme Naïve Bayes.

- Créer un classifieur à base de l'algorithme Naïve Bayes.
- Appliquer le classifieur sur l'ensemble de données
- appliquer le classifieur résultant sur les data test
- calculer la Précision
- la matrice de confusion

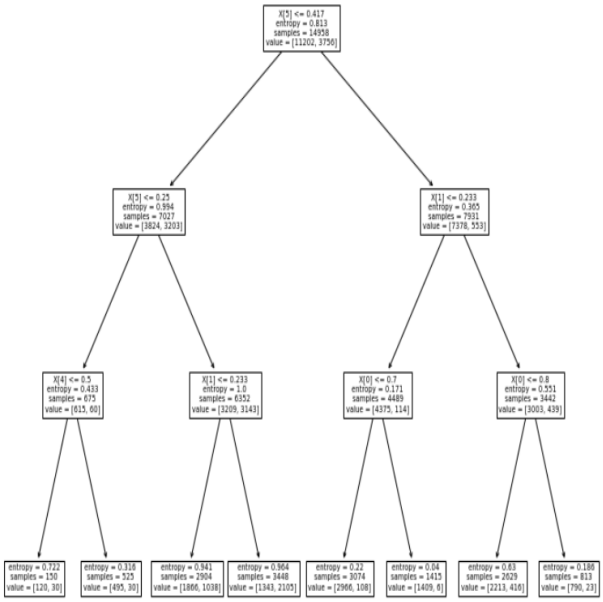
Arbres de décision

L'algorithme d'arbre de décision appartient à la catégorie des algorithmes d'apprentissage supervisé. Cela fonctionne à la fois pour les variables de sortie continues et catégorielles.

- Créer un classifieur à base de l'algorithme Arbres de décision
- Appliquer le classifieur sur l'ensemble de données
- Afficher l'arbre résultant sous forme textuelle

```
|--- feature_5 <= 0.42
|   |--- feature_5 <= 0.25
|   |   |--- feature_4 <= 0.50
|   |   |   |--- class: 0
|   |   |--- feature_4 >  0.50
|   |   |   |--- class: 0
|   |--- feature_5 >  0.25
|   |   |--- feature_1 <= 0.23
|   |   |   |--- class: 0
|   |   |--- feature_1 >  0.23
|   |   |   |--- class: 1
|--- feature_5 >  0.42
|   |--- feature_1 <= 0.23
|   |   |--- feature_0 <= 0.70
|   |   |   |--- class: 0
|   |   |--- feature_0 >  0.70
|   |   |   |--- class: 0
|   |--- feature_1 >  0.23
|   |   |--- feature_0 <= 0.80
|   |   |   |--- class: 0
|   |   |--- feature_0 >  0.80
|   |   |   |--- class: 0
```

- Enregistrer l'arbre résultant dans un fichier et Produire une représentation graphique de l'arbre résultant



- appliquer le classifieur résultant sur les data test
- calculer la Précision
- la matrice de confusion

K plus proches voisin

L'algorithme des k plus proches voisins est un algorithme d'apprentissage supervisé, il sera possible de classer (déterminer le label) d'une nouvelle donnée, par calcule les distances entre ces donnée

On répéter les mêmes étapes pour 'K plus proches voisin'
On a déroulé l'algorithme pour plusieurs valeurs de k .

Et à la fin afficher les résultats (la Précision) sur graph pour mieux compare les résultats de prédiction (pour les data teste et data train)