ESILV-Alt 3–A3
Année : 2023-2024
Module : MESIIN360523
Durée : 1h30

TD noté en analyse de données

Type du rendu : un notebook

Charger le jeu de donnée « auto.csv » présent sur DVL.

Nous allons effectuer une analyse exploratoire des données (EDA) avant d'appliquer la PCA. Notre étude comprend fondamentalement trois tâches principales :

Étape 1 : Aperçu de l'ensemble de données et analyse descriptive

Étape 2 : Prétraitement et nettoyage des données brutes

Étape 3: Application de la PCA et visualisation des données

Étape 1 : Aperçu de l'ensemble de données et analyse descriptive

1. Afficher les 10 premières lignes du dataframe. Que remarquez-vous ?

2. ***Surprise au premier rang :*** première ligne ici n'est pas un en-tête ; c'est juste plus de données. Ajouter des noms aux colonnes comme suit puis afficher les 10 premières lignes :

   col_names = ["symboling" , "normalized-losses" , "make" , "fuel-type" , "aspiration" , "num-of-doors","body-style", "drive-wheels" , "emplacement du moteur" , "empattement" , "longueur" , "largeur" , "hauteur" , "poids à vide" , "type de moteur" , "nombre de cylindres" , "taille du moteur" , "système de carburant" , "alésage" , "course","taux de compression","puissance", "crème-rpm" , "ville-mpg" , "autoroute-mpg" , "prix"].

3. Explorer votre dataframe en utilisant les fonctions suivantes :
   - **df.head() ,df.tail(), df.info():** Vu globale de l'ensemble de jeu de donnée
   - **df.shape :** Nombre d'observations
   - **df.dtypes :** Nombre et types d'entités
   - **df.describe(), df.describe(include='object'):** Statistique descriptive de l'ensemble de données

Etape 2 : prétraitement et nettoyage des données

En général, cette étape comprend ce qui suit :

- **Gestion de types**
  - **df.dtypes : pour vérifier le type de données**
  - **Df.dtypes.unique() :pour donner le nb de variables de chaque type**
  - **df.astype() pour changer le type de données**
- **Suppression des colonnes inutiles (df.drop())**
- **Suppression des valeurs manquantes (df.dropna())**

- **Remplacement des valeurs numériques manquantes par : moyenne, mode, médiane, min, max, etc… (df.fillna(), df.replace())**
- **Suppression des doublons. (df.duplicates() , df.drop_duplicates())**
- **Gestion du format de l'heure.**
- **Gérer les valeurs aberrantes.**

4. Compter le nombre de valeurs manquantes en utilisant *df.isna().sum().* Que remarquez-vous ?
5. Une deuxième surprise : des colonnes contiennent un « ? ». La fonctions *isna()* ne peut pas les détecter. Remplacer les « ? » par des *np.na*
6. Compter les doublons et supprimer les si besoin.
   df.duplicated().sum() *:* retourne le nombre d'observations en doublons

   df.drop_duplicates() *:*supprime les doublons

   - Compter à nouveau les valeurs manquantes et les doublons en utilisant les fonctions suivantes **Nombre ou taux des valeurs manquantes**
     - *df.isna().sum() #ou bien df.isnull().sum()*
     - *df.isna().sum().sum()* # nombre totale de cellules manquantes
     - *round(df.isna().sum().sum() / df.size * 100, 1)* # percentage de cellules manquantes

7. En déduire le pourcentage des cellules manquantes. Afficher un tableau qui ne montre que les colonnes avec des cellules manquantes.
8. Inspecter les types de chacune des colonnes ? utiliser df.dtypes.unique().
9. Donner le diagramme circulaire des types de données. Utiliser df.dtype.value_counts() pour obtenir le nb des variables appartenant à chaque type.
10. Nettoyer les données comme suit :
    a. Supprimer les colonnes qui contiennent plus de 15% de données manquantes (utiliser le paramètre *thresh* de la fonction ***dropna***). Vérifier que la colonne *'normalized-losses'* est supprimée.
    b. Dans chacune des colonnes « alésage », « course », « puissance » et "crème-rpm » :
       i. Convertir les données numérique de type 'O' en 'float64' en utilisant df.astype()
       ii. Remplacer les NA par la moyenne
    c. Dans la colonne « num-of-doors » : remplacer les na par « four »
11. Supprimer toutes les lignes manquantes s'il y en a.

12. Sélectionner pour la suite les colonnes numériques du dataframe. Calculer son coefficient de corrélation et afficher son heatmap. Quelles sont les variables les plus correlées entre elles ?

13. Sélectionner pour la suite les colonnes numériques ['longueur', 'largeur', 'hauteur',"prix"]. On note X ce nouveau dataframe. Standardiser le avec le SdandardScaler
14. Appliquer le skee plot pour déterminer le meilleur nombre de composante principale

15. Appliquer la PCA en utilisant 2 composantes.
16. Afficher les données avec les projections et les axes principaux.
17. Commenter et analyser les résultats.