# **Analyse de données médical avec l'Analyse en Composantes Principales**

L'objet de ce projet est de réduire significativement la dimension d'un jeu de données utilisé pour un problème de classification. J’utilise principalement des techniques de représentations de réduction de dimensions tels que L'Analyse en Composantes Principales (ACP), l'algorithme K-means et du T-SNE (t-distributed stochastic neighbor embedding) qui permettent d'avoir une visualisation proche de la réalité quand elle n'est pas initialement représentable sur deux axes par exemple.

 L'analyse en composantes principales (ACP) est une méthode de réduction de dimension qui consiste à transformer des variables corrélées en nouvelles variables décorrélées les unes des autre. Il s’agit de résumer l’information contenue dans un ensemble de données en un certain nombre de variables synthétiques, combinaisons linéaires des variables originelles : ce sont les Composantes Principales. L'enjeu est généralement de réduire de manière significative la dimension du jeu de données tout en conservant au maximum l'information véhiculée par les données pour pouvoir d'identifier des structures intéressantes et d'adapter au mieux le choix de son algorithme d'apprentissage par la suite.

Les packages utilisés seront numpy, pandas, matplotlib, scikit-learn et ses sous-packages.

## **1.Jeu de données**

Le jeu de données est issu d'une étude menée sur un groupe de 400 patients agés de 2 à 90 ans atteint ou non d'une maladie des reins. Pour chaque observation, une personne est décrite selon 24 variables. Description des variables du fichier kidney\_disease :

id : Identifiant unique du patient

**age :** Age du patient

**bp :** Pression du sang (Blood Pressure)

**sg** : Densité spécifique de l’urine (Specific Gravity)

**al :** Taux d’Albumin

**su :** Taux de sucre (Sugar)

**rbc :** Nombre de globules rouges (red blood cell count)

**pc :** Taux de cellules de pus (Pus Cells)

**pcc :** Présence ou non de Chlorochromate de pyridinium

**ba :** Bactérie (Présence ou Non)

**bgr :** Glycémie aléatoire (mgs/dL) (Blood Glucose Random)

**bu :** Urée sanguine (mgs/dL) (Blood Urea)

**sc :** Créatine sérique(mgs/dL) (Serum Creatinine)

**sod :** Sodium (mEq/L)

**pot :** Potassium (mEq/L)

**hemo :** Hémoglobine (gms)

**pcv :** Volume de globules (Packed Cell Volume)

**wc :** Nombre de globules blancs (White blood cell count)

**htn :** Hypertension

**dm :** Diabète Mellitus

**cad :** Maladie de l’artère coronaire

**appet :** Appétit (Faible ou normal)

**pe :** Œdème (Oui ou Non)

**ane :** Anémie (Oui ou Non)

**Classification :** Label du patient (0 à non malade / 1 à malade

Le lien sur la Base des données est :

<https://www.kaggle.com/datasets/akshayksingh/kidney-disease-dataset/data>

## **2.Première exploration du jeu de données**

Avant de mettre des données dans un algorithme de machine learning, il est nécessaire d'en faire une première analyse pour ne pas avoir de résultats faussés ou non-représentatif de la réalité (il faut faire attention à la valeur manquante et aberrante par exemple.).

### Les étapes de nettoyage sont :

* Correction de valeurs erronées (\t, \ etc) ;
* Séparation des données en 2 catégories par type des données les variables numériques ou catégorielles ;
* Nettoyage des valeurs manquantes par la moyenne de chaque variable pour les variables numériques ;
* Encodage des variables catégorielles et remplacement par la valeur la plus fréquente de chaque modalité .

Une fois la normalisation des données faite, nous pouvons commencer notre Analyse en Composantes Principales.

## 3. L’analyse

L’observation la matrice de corrélation et de la distribution des variables

La matrice permet par exemple d’observer la relation positive forte entre les variables 'classification' et 'al'(Taux d’Albumin), htn (Hypertension) et Diabète Mellitus (dm) et négative entre avec sg (Densité spécifique de l’urine ) et Hémoglobine.

Toutes ces corrélations entre variables vont conditionner la composition des axes factoriels dont le sens et la signification s'interpréteront en fonction de leur corrélation avec chaque variable.

Je vais utiliser l'analyse en composantes principales (PCA) est une technique couramment utilisée en statistiques et en apprentissage automatique pour réduire la dimensionnalité des données. Elle vise à transformer un ensemble de données contenant un grand nombre de variables en un nouvel ensemble de variables (les composantes principales) qui sont non corrélées les unes aux autres.

Les composantes principales sont classées en fonction de leur capacité à expliquer la variance des données d'origine. La première composante principale explique la plus grande partie de la variance, la deuxième composante principale explique la deuxième plus grande partie, et ainsi de suite. En réduisant la dimensionnalité des données, la PCA permet de simplifier leur analyse, de réduire le bruit et de mettre en évidence les tendances ou les structures sous-jacentes.

Donc je vais créer une instance de la classe PCA nommée model et je vais stocker la représentation de data obtenue par Analyse en Composantes Principales dans un array nommé coord\_acp.

### Les étapes:

* Choix le nombre de facteurs.

Le nombre de composantes entre 2-6 semble être optimale pour une compression des données par ACP.

* Affichage et l’analyse d’un cercle de corrélation, qui est souvent utilisé pour interpréter les relations entre les variables originales et les composantes principales générées par l'analyse en composantes principales (ACP).

Ici, on peut voir les deux premières composantes principales (PC1 et PC2) en combinaisons linéaires des variables originales et capturent la variance maximale dans les données.

Chaque variable originale est représentée par une flèche qui part du centre (0,0) du cercle. La direction de la flèche indique la corrélation entre la variable originale et la composante principale respective.

La variable le plus corrélée positivement au premier axe est :

-pot - Potassium avec la corrélation le plus forte

- bp - Pression du sang

- ba - bactérie

Par contre les variables qui contribuent davantage à la variance de la composante principale sont:

-sc - Créatine sérique(mgs/dL) (Serum Creatinine)

-bu -Urée sanguine (mgs/dL) (Blood Urea)

-al - Taux d’Albumin

-htn - Hypertension

-dm - Diabète Mellitus

-bgr - Glycémie aléatoire

-su-Taux de sucre

La variable le plus corrélée négativement au premier axe sont :

- pcv -Volume de globules

- hemo - Hémoglobine

Les variables contribuent davantage à la variance d'un deuxieme axe :

-sc - Créatine sérique(mgs/dL) (Serum Creatinine)

-bu -Urée sanguine (mgs/dL) (Blood Urea)

La variable le plus corrélée négativement au deuxième axe sont :

-su-Taux de sucre

-bgr - Glycémie aléatoire

Presque tous les variable sang (blood ) ont une corrélation positive (BU, bgr, bp)

Toutes les variables avec les valeurs normales ont une corrélation négative.

* Utilisation d’l'algorithme des K-moyennes

Dans notre cas, nous avons le diagnostic final du patient. Dans certains cas, cette variable n'est pas disponible. Il faut alors déterminer quel point appartient à quel groupe. Pour cela, nous utilisons l'algorithme des K-moyennes. Et nous avons obtenu Silhouette Score 24%

* L'algorithme du T-SNE

L'objectif de T-SNE est similaire à celui de l'ACP : représenter nos données dans une dimension plus petite.

Et nous avons obtenu Silhouette Score jusqu’à 55%

### Conclusions

Les résultats obtenus avec le T-SNE semblent être meilleurs que ceux obtenus lors de l'ACP. Les deux groupes de personnes malades ou non sont un peu plus distanciés, mais on aimerait avoir une distinction clairement visible sans coloration des coordonnées.

Le Silhouette Score est une mesure qui varie de -1 à 1, où une valeur plus élevée indique une meilleure séparation des clusters. Dans notre cas, un Silhouette Score de 0,58 suggère une séparation relativement bonne des clusters.

En conclusion l'utilisation de l'ACP suivie par l'algorithme t-SNE (t-Distributed Stochastic Neighbor Embedding) est une approche courante pour la visualisation des données de grande dimension. Cette combinaison permet de réduire la dimensionnalité des données de manière plus efficace tout en préservant les structures locales et globales.

Ici nous avons réussi à obtenir 2 clusters, malades et pas malades et nous avons trouvé 42 personnes en plus malades.