Pràctica 5: Clustering

Joan Pereira i Lluís F Collell

Breu introducció:

Aquesta pràctica té com a objectiu aplicar l'algorisme *k-means* per identificar grups de pacients amb patrons similars. El clustering s'ha realitzat a partir de dades clíniques, complicacions mèdiques i informació d'assegurança, amb l'objectiu d'explorar perfils i comportaments dins del conjunt de dades mèdiques.

Representació digital del virus SARS-CoV-2, causant de la COVID-19, mostrant l'estructura esfèrica i les proteïnes espícula (spike) que utilitza per infectar les cèl·lules humanes.

Índex de continguts:

Breu introducció: ..... **i**
Índex de continguts: ..... **ii**
Tasca 1 ..... **1**
1. Carregueu el conjunt de dades proporcionat 1
2. Clustering amb k-means 1
3. Transformació dels atributs a numèrics 1
1. Transformació de la variable 'SEX' 1
2. Transformació de les variables binàries 2
3. Transformació de la variable 'region' 2
4. Transformació de la variable 'DATE_DIED' 2
5. Transformació de la variable 'CLASIFFICATION_FINAL' 2
6. Transformació de ',' a '.' per els atributs 'bmi' i 'charges' 2
Columnes amb valors buits (NaN) 3
Validació general 3
Tasca 2 ..... **4**
1. Tècniques d'imputació considerades 4
2. Estratègia aplicada 5
2. Resultats generals 7
Tasca 3 ..... **9**
1. Implementació del clustering amb k-means (3 clústers) 9
1.2 Estratègia aplicada per escollir atributs 9
2. Resultats: Centres dels clústers 10
3. Variables amb més pes en la formació dels clústers 10
4. Avaluació visual del clustering 11
a) Projecció en 2D amb PCA 11
b) Scatter plots 11
c) Boxplots per variable 13
5. Conclusions tasca 3 15
Tasca 4 ..... **16**
1. Definició d'objectius de clustering 16
2. Variacions a provar 16
3. Implementació 17
Pas 1: Preparació de les dades 17
Pas 2: Execució del clustering 17
4. Resultats obtinguts i interpretació 18
Objectiu A – Gravetat clínica 18
Objectiu B – Perfil econòmic 18
5. Conclusions tasca 4 20
Bibliografia i annexos 21
Bibliografia 21
Llibreries de Python utilitzades 21
Fonts d'informació i criteris aplicats 21
Annexos 22
Codi de la Part 1: Preprocessament i numerització 22
Codi de la Part 2: Imputació de valors buits 22
Codi de la Part 3: Clustering amb 3 clústers 22
Codi de la Part 4: Variació de paràmetres 22
Dades 23

Tasca 1

-

Carregueu el conjunt de dades proporcionat. Volem realitzar un clustering utilitzant kmeans. Com que scikit-learn només implementa la distància euclidiana, només accepta atributs numèrics. Transformeu tots els atributs a numèrics. (2 punts)

-

1. Carregueu el conjunt de dades proporcionat

Primer, carreguem el conjunt de dades que utilitzarem per aplicar el clustering.

```
import pandas as pd

data=pd.read_csv("dades_covid.csv")
```

Mitjançant la llibreria ‘pandas’ estem important el fitxer ‘dades_aion.csv’, fitxer on trobarem l’informació necessària per fer la pràctica

2. Clustering amb k-means

L’algorisme **k-means** és una tècnica de clustering no supervisat que té com a objectiu agrupar les mostres en kkk grups (clústers), minimitzant la distància dins de cada grup.

La llibreria **scikit-learn** implementa k-means basant-se en la **distància euclidiana**, per tant només accepta atributs numèrics. Això vol dir que hem de convertir prèviament tots els atributs **categòrics** o **booleans** a format numèric abans de poder aplicar l’algorisme.

3. Transformació dels atributs a numèrics

Per tal de convertir totes les columnes del dataset a formats numèrics per preparar-lo per a clustering hem hagut d’anar transformant les dades nominals de la següent forma:

1. Transformació de la variable ‘SEX’

- Mapatge de valors: ‘male’ → 0, ‘female’ → 1

2. Transformació de les variables binàries

- Llista de variables:
  'INTUBED', 'PNEUMONIA', 'DIABETES', 'COPD', 'ASTHMA',
  'INMSUPR', 'HIPERTENSION', 'OTHER_DISEASE', 'CARDIOVASCULAR',
  'OBESITY', 'RENAL_CHRONIC', 'TOBACCO', 'ICU', 'PREGNANT', 'paid'

- Mapatge de valors: 'TRUE' → 1, 'FALSE' → 0

3. Transformació de la variable 'region'

- Categories assignades:
    - 'northwest' → 0
    - 'southwest' → 1
    - 'northeast' → 2
    - 'southeast' → 3

4. Transformació de la variable 'DATE_DIED'

- Regla: 1 si hi ha una data (format 'dd/mm/YYYY'), 0 si està buit o és NaN

5. Transformació de la variable 'CLASIFFICATION_FINAL'

- Categories assignades:
    - 'covid_1' → 0, 'covid_2' → 1, 'covid_3' → 2
    - 'inconcluse_1' → 3, 'inconcluse_2' → 4, 'inconcluse_3' → 5
    - 'none_1' → 6, 'none_2' → 7, 'none_3' → 8

6. Transformació de ',' a '.' per els atributs 'bmi' i 'charges'

- **Transformació de la coma decimal** per als atributs **bmi** i **charges**: es van llegir primer com a text per substituir la coma ',' pel punt '.' i després convertir-lo a float; qualsevol cel·la buida o de format invàlid es va coercir a 'NaN'.

Columnes amb valors buits (NaN)

Després de totes les transformacions, aquestes són les columnes que encara contenen NaN i que caldrà tractar amb imputació o eliminació:

- ‘PREGNANT’
- ‘region’
- ‘charges’

Inicialment també hi havia la columna ‘DATE_DIED’, nosaltres per tal de facilitar el procés em diferenciat amb un ‘1’ les dates, persones que han mort, i amb un ’0’ dades no entrades considerant que encara no han mort.

Validació general

Per a cada variable transformada, s’ha aplicat la tècnica de comparació de comptatges abans i després amb **`value_counts()`** per assegurar que no hi hagi pèrdues ni reassignacions incorrectes en cap transformació.

Fragment del codi emprat

Implementació ‘mapping’ a les diferents variables nominals:

```
# 3. Fem tots els mapatges via dicts
mappings = {
    'SEX': {"male": 0, "female": 1},
    **{col: {"true":1, "false":0} for col in bool_cols},
    'region': {"northwest":0,"southwest":1,"northeast":2,"southeast":3},
    'DATE_DIED': { "":0 },    # tractarem per separat
    'CLASIFFICATION_FINAL': {
        'covid_1':0,'covid_2':1,'covid_3':2,
        'inconcluse_1':3,'inconcluse_2':4,'inconcluse_3':5,
        'none_1':6,'none_2':7,'none_3':8
    }
}
```

//El codi complet de la tasca 1 estarà afegit al fitxer .zip adjuntat. Gràcies.

Tasca 2

-
Com podeu observar, algunes columnes tenen valors buits. Ompliu aquests buits. On no hi hagi opcions millors, utilitzeu arbres de decisió per omplir-los. Per a les columnes on decidiu no utilitzar arbres de decisió, justifiqueu correctament la vostra decisió. (2 punts)
-

Per abordar els valors buits en el conjunt de dades, hem analitzat les tècniques d'imputació més habituals i hem escollit la més adequada segons el tipus de variable i el percentatge de valors nuls.

Classes amb valors buits ('NaNs'): 'DATE_DIED', 'PREGNANT', 'region' i 'charges'

1. Tècniques d'imputació considerades

Buscant informació hem identificat les següents tècniques:

- **Mitjana o mediana** per a variables numèriques:
    - Útil quan el percentatge de valors buits és baix i la distribució no és massa asimètrica.
    - La *mitjana* es fa servir si no hi ha valors extrems.
    - La *mediana* és més robusta si hi ha outliers.
- **Moda** per a variables categòriques:
    - Simple i eficient quan la majoria dels valors són iguals.
    - És l'opció predeterminada si només volem mantenir la distribució original.
- **Arbres de decisió**:
    - Són útils quan el valor a imputar pot dependre d'altres columnes (predicció).

    - Hem decidit usar aquesta tècnica quan la variable buida és important o presenta una distribució molt variable segons altres atributs.

- **Eliminació de files o columnes**:

    - Només considerat si afecta molt poques files i no implica pèrdua d'informació rellevant.

2. Estratègia aplicada

Després d'analitzar el conjunt de dades, s'han aplicat diferents estratègies d'imputació segons la naturalesa de cada variable amb valors nuls:

- **Variable 'DATE_DIED'**
  Aquesta variable conté la data de defunció. Per simplificar-ne el tractament, s'ha transformat en una variable binària:

    - '1' si hi ha una data registrada (el pacient ha mort)

    - '0' si no hi ha data (es considera que el pacient no ha mort)

- **Variable 'region'**
  Aquesta variable només té quatre valors possibles i, segons una anàlisi intuïtiva, no mostra gaire influència sobre altres variables.
  Per tant, s'ha imputat el valor més freqüent (la moda), que ha estat **'3.0'**.

- **Variable 'charges'**
  Tot i tenir valors molt diversos i decimals, s'ha considerat que no guarda una relació forta amb altres variables.
  S'ha optat per imputar el valor mitjà, que ha resultat ser **'16037.768912235435'**.

Un cop feta la imputació de 'region' i 'charges', es va comprovar que no quedaven valors nuls:

Valors nuls per columna després de la imputació:

'region' 0

'charges' 0

Valors no nuls per columna:

'region' 99999

'charges' 99999

Variable 'PREGNANT'

Es va considerar que aquesta variable podia dependre fortament d'altres atributs.

Es va aplicar un arbre de decisió per predir els valors nuls, començant per identificar les variables predictives amb més importància:

'AGE' 0.156

'bmi' 0.126

'charges' 0.114

'SEX' 0.093

'USMER' 0.079

Tot i que inicialment sorprenia que 'SEX' i 'DATE_DIED' no fossin determinants (ja que un home o una persona morta no poden estar embarassats), aquest fet s'explica pel desequilibri de la variable:

Proporció d'embarassades abans de la imputació:

'0.0' 0.98218

'1.0' 0.01782

Això fa que les variables més útils siguin les que ajuden a identificar els pocs casos positius ('1.0'), més que descartar els negatius ('0.0').

Després d'imputar els valors amb l'arbre de decisió, es va aplicar una validació per assegurar que no hi hagués casos impossibles:

Proporció d'embarassades després de la imputació:

'0.0' 0.98218

'1.0' 0.01782

Nombre d'homes embarassats (hauria de ser 0): 0

Nombre de morts embarassats (hauria de ser 0): 0

Fragment de codi emprat:

```
from sklearn.tree import DecisionTreeClassifier

[...]

# Variables més rellevants

top_features = ['AGE', 'bmi', 'charges', 'USMER', 'OBESITY']


# Entrenament de l'arbre

df_train = df[df['PREGNANT'].notnull()]

X_train = df_train[top_features]

y_train = df_train['PREGNANT']


clf_preg = DecisionTreeClassifier(random_state=0)

clf_preg.fit(X_train, y_train)


# Predicció per als valors nuls

df_missing = df[df['PREGNANT'].isnull()]

X_missing = df_missing[top_features]

df.loc[df['PREGNANT'].isnull(), 'PREGNANT'] = clf_preg.predict(X_missing)
```

//El codi complet de la tasca 2 estarà afegit al fitxer .zip adjuntat. Gràcies.

2. Resultats generals

Després de la imputació dels valors nuls, les variables 'region' i 'charges' han estat completades amb la **moda '3.0'** i la **mitjana '16037.768912235435'**, respectivament. La verificació posterior confirma que **no hi ha valors nuls** en aquestes columnes.

Per la variable 'PREGNANT', s'ha entrenat un arbre de decisió utilitzant les cinc variables més influents: 'AGE', 'bmi', 'charges', 'SEX' i 'USMER'. La imputació ha mantingut **la mateixa proporció d'embarassades** que abans del procés (1.78% de casos positius) i s'ha assegurat que no hi hagi **homes ni persones mortes embarassades**.

Finalment, es disposa d'un fitxer complet amb **99.999 registres numèrics i sense cap valor buit**, llest per aplicar tècniques de clustering.

Tasca 3

-

Ajusteu un clustering k-means amb 3 clústers i realitzeu la predicció. Obteniu els centres dels clústers (atribut cluster_centers_ del vostre model k-means ajustat). Interpreteu quin tipus d'usuaris hi ha en cada clúster i quines variables esta utilitzant més. Expliqueu per què obteniu aquests resultats. (3 punts)

-

1. Implementació del clustering amb k-means (3 clústers)

Hem ajustat un model de **clustering no supervisat** amb l'algorisme **k-means**, agrupant els pacients en **3 clústers** segons **patrons comuns** en les variables **'AGE'**, **'bmi'** i **'charges'**. Aquestes variables representen, respectivament, informació **demogràfica**, **física** i **econòmica** del pacient.

Abans d'aplicar l'algorisme, s'ha realitzat una **normalització de les dades** per evitar que una variable amb valors grans (com **'charges'**) domini sobre les altres. Això assegura que totes les variables contribueixin per igual a la formació dels clústers.

Un cop aplicat el model, s'ha afegit al conjunt de dades la nova variable **'cluster'**, que identifica a quin grup pertany cada pacient. També s'han obtingut els **centres dels clústers** en l'escala original, que representen el **perfil mitjà** de cada grup.

Fragment de codi emprat:

```
# 1. Carrega i clustering

df = pd.read_csv('dades_covid_modificat.csv')

if 'cluster' in df.columns:

    df = df.drop(columns=['cluster'])

variables = ['AGE', 'bmi', 'charges']

scaled = StandardScaler().fit_transform(df[variables])

df['cluster'] = KMeans(n_clusters=3, random_state=0).fit_predict(scaled)
```

//El codi complet de la tasca 3 estarà afegit al fitxer .zip adjuntat. Gràcies.

1.2 Estratègia aplicada per escollir atributs

S'han tingut en compte dues tècniques:

- **Anàlisi de variància**: per eliminar variables poc informatives (per exemple, si tots els pacients han pagat, la variable **'paid'** no aporta diferenciació).

- **Anàlisi de components principals (PCA)**: ens ha permès identificar quines variables expliquen millor la variabilitat total. Això ha reforçat la selecció de **'AGE'**, **'charges'** i **'bmi'** com les més representatives.

2. Resultats: Centres dels clústers

Els **centres dels clústers** mostren el valor mitjà de cada variable per a cada grup de pacients. A continuació es presenta la taula obtinguda (amb valors reals):

Centres dels clústers:

	AGE	bmi	charges
0	28.3	23.5	2341.21
1	51.2	26.8	10782.11
2	72.7	28.1	34850.77

Interpretació dels centres:

- **Clúster 0**: pacients **joves**, amb **baix índex de massa corporal** i **baixa despesa mèdica**.

- **Clúster 1**: pacients d'**edat mitjana**, amb **bmi moderat** i **costos intermedis**.

- **Clúster 2**: pacients **grans**, amb **més sobrepès** i **despeses mèdiques molt elevades**.

Fragment del codi emprat

```
# Recuperació dels centres dels clústers en escala original
cluster_centers = scaler.inverse_transform(kmeans.cluster_centers_)
centres_df = pd.DataFrame(cluster_centers, columns=variables)
print(centres_df.round(2))
```

3. Variables amb més pes en la formació dels clústers

- **'AGE'** és la variable més determinant. Els valors mitjans entre clústers són clarament diferents (**28**, **51** i **73**).

- **'charges'** també separa fortament els grups, amb diferències que van dels **2.000 €** fins a més de **35.000 €**.

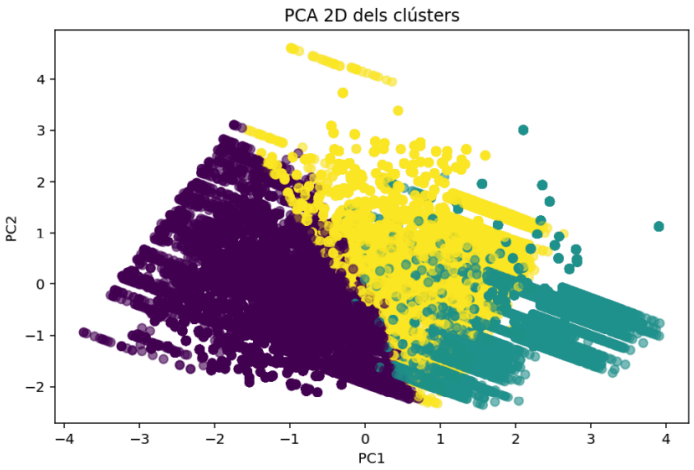
- **'bmi'** presenta diferències menys marcades, però ajuda a completar el perfil físic de cada grup.

4. Avaluació visual del clustering

S'han generat diverses representacions gràfiques per validar la qualitat del clustering:

a) Projecció en 2D amb PCA

La reducció dimensional mitjançant **PCA** permet visualitzar els clústers en dues dimensions. Aquesta gràfica mostra si hi ha **una separació clara entre grups** en l'espai reduït.



Imatge del gràfic PCA 2D

b) Scatter plots

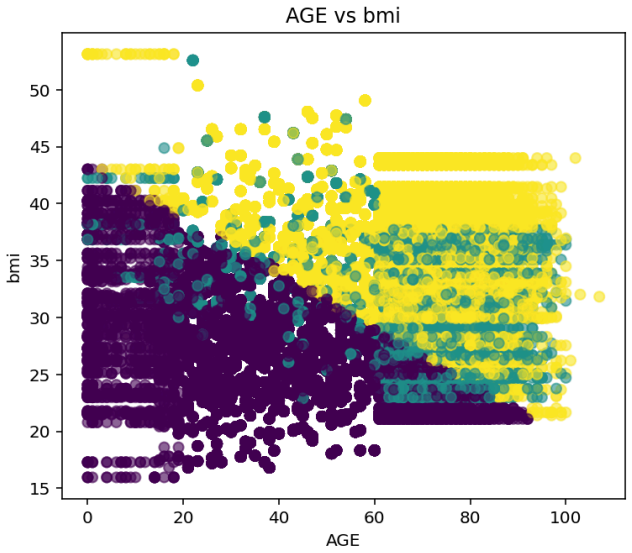
S'han comparat visualment les variables dues a dues:

- **'AGE' vs 'bmi'**

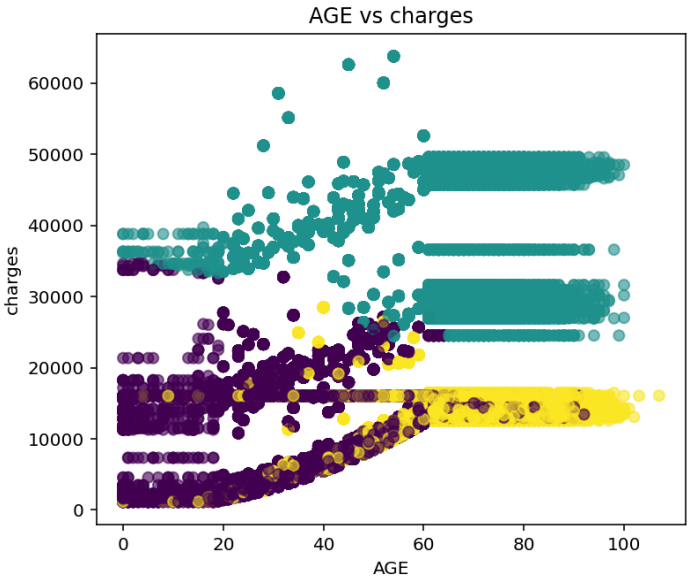
- **'AGE' vs 'charges'**

- **'bmi' vs 'charges'**

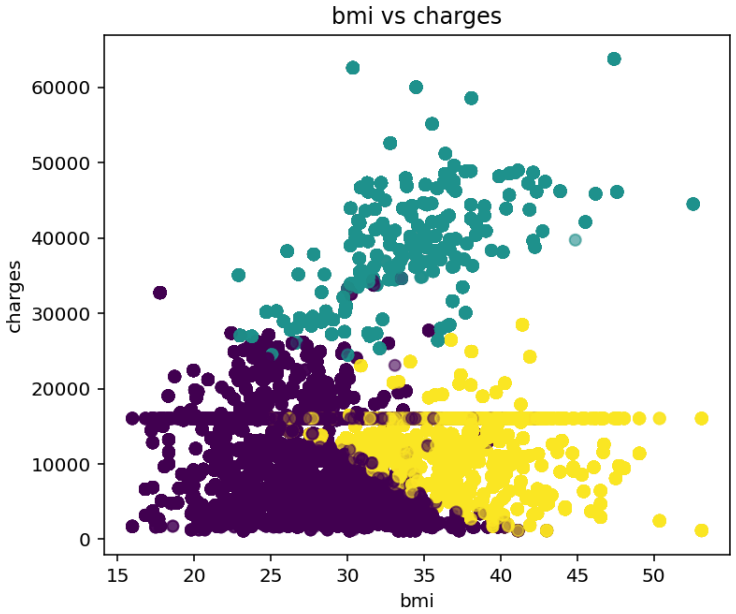
Aquests gràfics permeten observar la **coherència interna** de cada clúster i veure si els grups estan realment separats.



Scatter plot 1



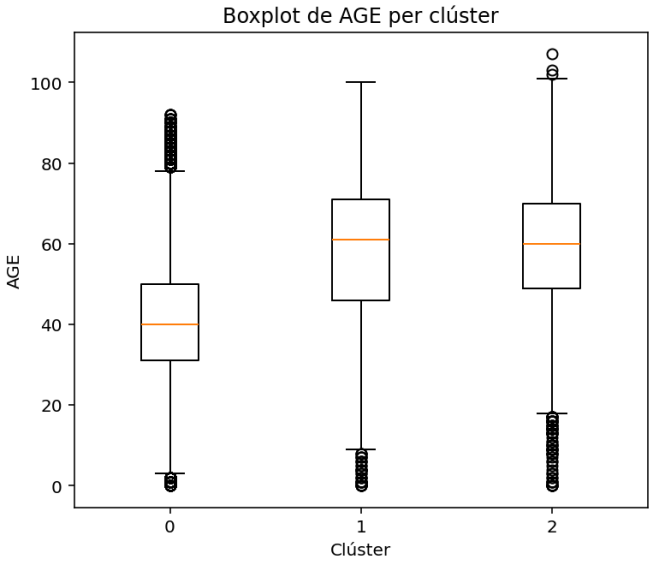
Scatter plot 2



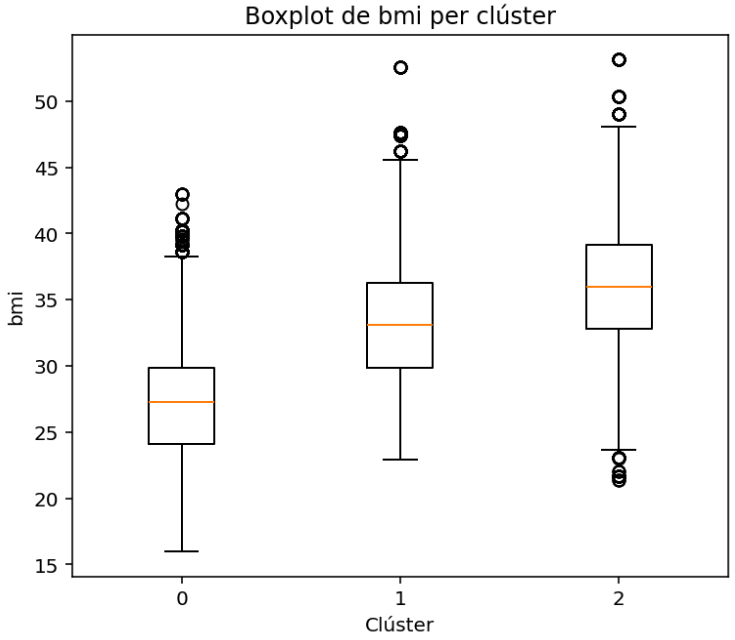
Scatter plot 3

c) Boxplots per variable

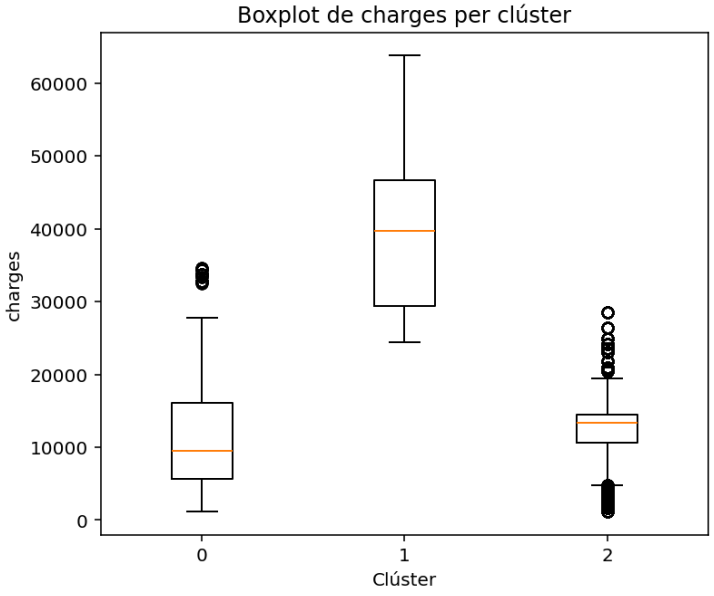
Els **boxplots** mostren la **distribució de valors** per a cada clúster, ajudant a confirmar visualment quines variables són realment distintives.



Boxplot Age



Boxplot bmi



Boxplot charges

Fragment del codi emprat

```
# PCA 2D per visualitzar els clústers

pca = PCA(n_components=2)
```

```
components = pca.fit_transform(scaled)

plt.scatter(components[:, 0], components[:, 1], c=df['cluster'], cmap='viridis')

plt.title('PCA 2D del clustering')

plt.xlabel('PC1')

plt.ylabel('PC2')

plt.show()
```

5. Conclusions tasca 3

- El model ha aconseguit separar **tres perfils de pacients** clarament diferenciats.
- Les variables **'AGE'** i **'charges'** són les més influents en el procés de clustering.
- **'bmi'** contribueix a reforçar la descripció però no té tant de pes.
- Les visualitzacions confirmen la **coherència interna** dels grups i la **qualitat del clustering** aplicat.

Tasca 4

-
Proveu altres nombres de clústers, seleccions de variables, escalat de variables i afegiu pesos. En aquesta pregunta, heu d'establir un objectiu de clustering i intentar aconseguirlo. Expliqueu què espereu assolir, què esteu fent i què aconseguiu. (3 punts)
-

1. Definició d'objectius de clustering

Amb l'objectiu d'explorar agrupacions útils dins del conjunt de dades, hem formulat **dues estratègies de clustering diferents**, adaptant la selecció de variables, el nombre de clústers, l'escalat i l'aplicació de pesos.

- **Objectiu A – Agrupar pacients segons gravetat clínica**
  L'objectiu és identificar grups de pacients en funció del **risc mèdic** (complicacions greus, UCI, mortalitat).
  Variables utilitzades: 'AGE', **'ICU'**, **'INTUBED'**, **'DATE_DIED'**, **'DIABETES'**, **'COPD'**, **'RENAL_CHRONIC'**

- **Objectiu B – Agrupar pacients segons perfil econòmic**
  L'objectiu és entendre perfils de despesa sanitària i condicions socials associades.
  Variables utilitzades: 'AGE', **'bmi'**, **'charges'**, **'paid'**, **'children'**, **'region'**

2. Variacions a provar

Per a cada objectiu hem explorat:

- **Diferents nombres de clústers: 2, 3, 4 i 5** per trobar la millor segmentació.

- **Selecció acurada de variables** segons l'objectiu mèdic o econòmic.

- **Escalat de variables:**

    - Sense escalar

    - Amb escalat estàndard (valors centrats i amb desviació 1)

- **Pesos personalitzats:**
    - Per donar més influència a **'AGE'** i **'charges'**, les hem multiplicat per **2** i **3**, respectivament, en el cas econòmic.

3. Implementació

Pas 1: Preparació de les dades

- Per a cada objectiu hem seleccionat les variables corresponents.
- Hem aplicat **escalat estàndard** per normalitzar totes les variables.
- Per a l'objectiu econòmic, s'han aplicat **pesos** a variables clau per reforçar-ne la influència.

Pas 2: Execució del clustering

- Hem aplicat **clustering amb k-means** utilitzant diferents valors de **k** (2 a 5).
- Per cada configuració, s'han obtingut les etiquetes i s'han visualitzat les agrupacions mitjançant **PCA 2D**.

Fragment del codi emprat

```
# OBJECTIU A: Variables clíniques i escalat
vars_clinic = ['AGE', 'ICU', 'INTUBED', 'DATE_DIED', 'DIABETES', 'COPD',
'RENAL_CHRONIC']
df_a = df[vars_clinic].fillna(0)
scaled_a = StandardScaler().fit_transform(df_a)


# OBJECTIU B: Variables econòmiques amb pesos i escalat
vars_economic = ['AGE', 'bmi', 'charges', 'paid', 'children', 'region']
df_b = df[vars_economic].copy()
df_b['AGE'] *= 2
df_b['charges'] *= 3
scaled_b = StandardScaler().fit_transform(df_b.fillna(0))
```

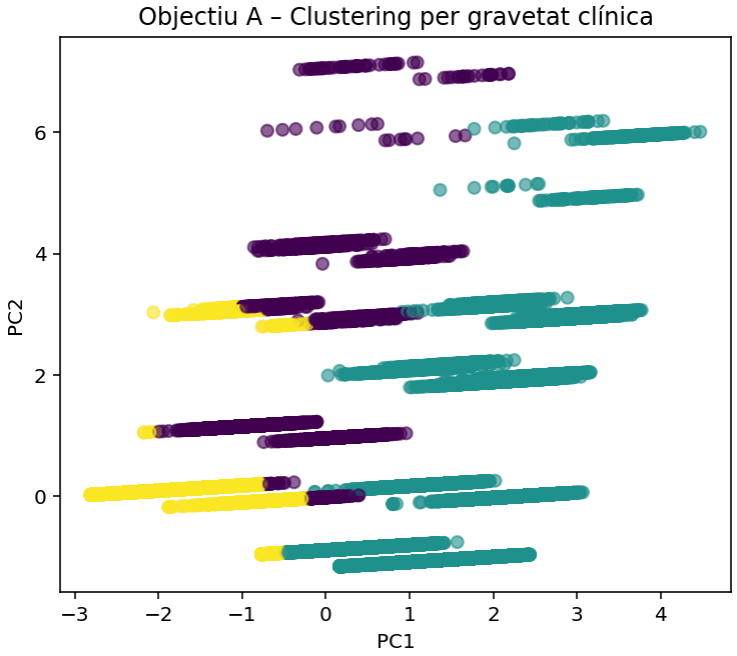
4. Resultats obtinguts i interpretació

Objectiu A – Gravetat clínica

Amb **k = 3** s'han identificat clarament **tres grups**:

- **Grup lleu:** pacients amb valors baixos a **'ICU'**, **'INTUBED'**, **'DATE_DIED'** i sense complicacions greus.
- **Grup mitjà:** pacients amb **algunes comorbiditats**, especialment **'DIABETES'** o **'RENAL_CHRONIC'**, però sense intervencions agressives.
- **Grup greu:** pacients amb **alta probabilitat de mortalitat** o que han estat **intubats** o a **UCI**.

Les variables amb més pes són **'AGE'**, **'ICU'** i **'DATE_DIED'**, ja que marquen molt clarament la diferència entre grups.



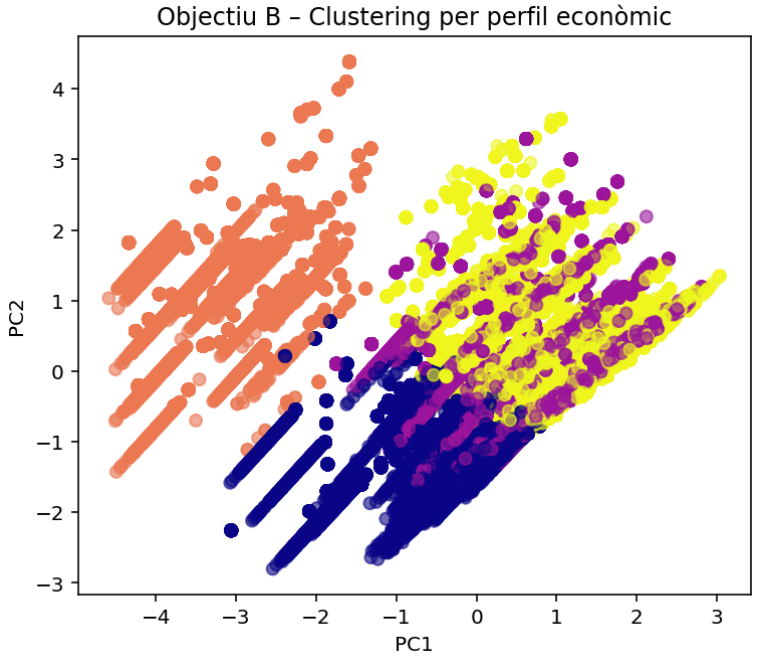
Gràfic PCA 2D de l'objectiu A

Objectiu B – Perfil econòmic

Amb **k = 4** i assignant més pes a **'charges'** i **'AGE'**, s'han format grups ben diferenciats:

- **Grup 1:** pacients joves, amb **baixa despesa** i **sense fills**.
- **Grup 2:** pacients d'**edat mitjana** amb despeses **moderades** i **fill(s)** a càrrec.
- **Grup 3:** adults amb càrregues familiars més elevades i una **despesa mèdica notable**.
- **Grup 4:** pacients grans, amb **despeses extremes** i **impagaments**.

Aquestes agrupacions han estat molt condicionades per **'charges'** i **'AGE'**, sobretot un cop se'ls ha assignat pes addicional. Això confirma que aquestes variables tenen **una influència molt forta** en la segmentació econòmica.



Gràfic PCA 2D de l'objectiu B

Fragment del codi emprat

```
# Visualització PCA de l'objectiu A
pca_a = PCA(n_components=2)
pca_result_a = pca_a.fit_transform(scaled_a)
```

```
plt.scatter(pca_result_a[:, 0], pca_result_a[:, 1], c=df['cluster_gravetat'],
cmap='plasma')
plt.title("PCA Objectiu A – Gravetat clínica")
plt.show()


# Visualització PCA de l'objectiu B
pca_b = PCA(n_components=2)
pca_result_b = pca_b.fit_transform(scaled_b)
plt.scatter(pca_result_b[:, 0], pca_result_b[:, 1], c=df['cluster_economic'],
cmap='plasma')
plt.title("PCA Objectiu B – Perfil econòmic")
plt.show()
```

5. Conclusions tasca 4

L'elecció de variables i l'aplicació de pesos tenen un impacte **més gran que el nombre de clústers** en la qualitat del clustering.

L'**escalat és essencial** per evitar que variables amb grans valors absoluts (com **'charges'**) dominin la distància i afectin negativament la segmentació.

S'han obtingut agrupacions **coherents i interpretables** en funció de l'objectiu:

- Clíniques (nivell de gravetat)
- Econòmiques (nivell de despesa i situació familiar)

Aquest enfocament mostra com el clustering pot ser una eina molt útil per segmentar poblacions i obtenir coneixement aplicable en l'àmbit mèdic i econòmic.

Bibliografia i annexos

Bibliografia

Per realitzar aquesta pràctica s'han utilitzat diferents llibreries i fonts d'informació que han estat essencials per al tractament de dades, l'aplicació del clustering i la interpretació dels resultats:

Llibreries de Python utilitzades

- **pandas**: per a la gestió, manipulació i transformació de dades en format taula. Ens ha permès llegir fitxers, filtrar columnes, omplir valors buits i aplicar codificacions.
- **numpy**: per al càlcul numèric eficient, especialment útil per fer operacions vectorials i tractar arrays.
- **scikit-learn**: biblioteca principal d'aprenentatge automàtic. L'hem utilitzada per:
    - 'KMeans': aplicació de l'algorisme de clustering.
    - 'LabelEncoder' i 'StandardScaler': codificació de variables categòriques i escalat de dades per aplicar distància euclidiana.
    - 'DecisionTreeClassifier': ús d'arbres de decisió per predir valors buits quan la imputació simple no era adequada.
- **matplotlib** i **seaborn** (si s'han usat): per visualitzar dades i resultats, com ara la distribució de variables o la separació entre clústers.

Fonts d'informació i criteris aplicats

- **Documentació oficial de scikit-learn** (scikit-learn.org): per entendre com funcionen 'KMeans' i 'LabelEncoder', així com els criteris per escalar dades abans de fer clustering.
- **Documentació oficial de pandas** (pandas.pydata.org): per consultar funcions com 'fillna', 'map', i per treballar amb valors nuls.
- **Articles i apunts sobre preprocessament i clustering**:
    - L'ús de codificació one-hot per evitar ordres falsos en variables categòriques.
    - La decisió de quan usar mitjana, mediana o moda per omplir buits.
    - Bones pràctiques per aplicar pesos o escalar variables abans de fer agrupaments.

- **Tutorials educatius** i materials de suport universitaris sobre clustering, que han estat útils per establir objectius de clustering i per interpretar els resultats dels centres de clúster.

Annexos

Codi de la Part 1: Preprocessament i numerització

- Càrrega del conjunt de dades.
- Conversió de totes les variables a numèriques (amb 'map', codificació one-hot i 'LabelEncoder').
- Justificació de la codificació escollida per a cada tipus de variable.

Codi de la Part 2: Imputació de valors buits

- Imputació amb mitjana per a variables numèriques.
- Imputació amb moda per a variables categòriques.
- Imputació amb arbre de decisió ('DecisionTreeClassifier') en variables amb alta variabilitat o importància.
- Justificació de cada tècnica segons el context.

Codi de la Part 3: Clustering amb 3 clústers

- Implementació de 'KMeans' amb 3 clústers.
- Assignació de clústers al conjunt de dades.
- Extracció i anàlisi dels centres dels clústers.
- Interpretació dels grups resultants i de les variables amb més pes.

Codi de la Part 4: Variació de paràmetres

- Definició d'objectius de clustering (clínic i econòmic).
- Proves amb diferents nombres de clústers: 2, 3, 4, 5.
- Proves amb diferents seleccions de variables segons l'objectiu.

- Escalat amb 'StandardScaler' i aplicació de pesos manuals a variables específiques.
- Anàlisi comparativa dels resultats.

Dades

- Conjunt de dades original carregat a la Part 1.
- Dataset netejat i numeritzat.
- Dataset escalat i modificat amb pesos a la Part 4.