Contenido

[Contexto 2](#_Toc86059254)

[Objetivo 2](#_Toc86059255)

[Fuente de datos 2](#_Toc86059256)

[Preparación del entorno de ejecución 2](#_Toc86059257)

[1) Importar las bibliotecas necesarias 2](#_Toc86059258)

[2) Importar los datos 2](#_Toc86059259)

[Selección de características 3](#_Toc86059260)

[Evaluación visual 3](#_Toc86059261)

[Matriz de correlaciones 5](#_Toc86059262)

[Selección de variables 6](#_Toc86059263)

[Aplicación de los algoritmos 6](#_Toc86059264)

[Estandarización 6](#_Toc86059265)

[Algoritmo: Ascendente Jerárquico 7](#_Toc86059266)

[Se crean etiquetas en los clústeres 7](#_Toc86059267)

[Obtención de los centroides 8](#_Toc86059268)

[Interpretación de los datos obtenidos 8](#_Toc86059269)

[Visualización gráfica 9](#_Toc86059270)

[Algoritmo K-Means 9](#_Toc86059271)

[Método del codo 9](#_Toc86059272)

[Creando las etiquetas en los clústeres 10](#_Toc86059273)

[Obtención de los centroides 11](#_Toc86059274)

[Interpretación de los clústeres obtenidos 11](#_Toc86059275)

[Visualización gráfica 12](#_Toc86059276)

[Conclusiones 13](#_Toc86059277)

[Link de Google Colab 13](#_Toc86059278)

# Contexto

Objetivo: Obtener grupos de pacientes con características similares, diagnosticadas con un tumor de mama, a través de clustering jerárquico y particional.

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

## Fuente de datos

Estudios clínicos a partir de imágenes digitalizadas de pacientes con cáncer de mama de Wisconsin (WDBC, Wisconsin Diagnostic Breast Cancer).

Texto

Descripción generada automáticamente

Fuente: <https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)>

# Preparación del entorno de ejecución

## Importar las bibliotecas necesarias

import pandas as pd               # Para la manipulación y análisis de datos

import numpy as np                # Para crear vectores y matrices n dimensionales

import matplotlib.pyplot as plt   # Para la generación de gráficas a partir de los datos

import seaborn as sns             # Para la visualización de datos basado en matplotlib

%matplotlib inline

## Importar los datos

Fuente de datos: WDBCOriginal.csv

from google.colab import files

files.upload()

# Para importar los datos desde Drive

#from google.colab import drive

#drive.mount('/content/drive')

BCancer = pd.read\_csv("WDBCOriginal.csv")

BCancer

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

BCancer.info()

Un texto con letras negras

Descripción generada automáticamente

print(BCancer.groupby('Diagnosis').size())

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente con confianza media

Se puede observar que **357 usuarios tienen un tumor benigno**, mientras que **212 usuarios tienen un tumor maligno**.

# Selección de características

## Evaluación visual

sns.pairplot(BCancer, hue='Diagnosis')

plt.show()

Texto, Tabla, Pizarra

Descripción generada automáticamente

sns.scatterplot(x='Radius', y ='Perimeter', data=BCancer, hue='Diagnosis')

plt.title('Gráfico de dispersión')

plt.xlabel('Radius')

plt.ylabel('Perimeter')

plt.show()

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

sns.scatterplot(x='Concavity', y ='ConcavePoints', data=BCancer, hue='Diagnosis')

plt.title('Gráfico de dispersión')

plt.xlabel('Concavity')

plt.ylabel('ConcavePoints')

plt.show()

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

## Matriz de correlaciones

Una matriz de correlaciones es útil para analizar la relación entre las variables numéricas. Se emplea la función corr().

CorrBCancer = BCancer.corr(method='pearson')

CorrBCancer

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

print(CorrBCancer['Radius'].sort\_values(ascending=False)[:10], '\n')   #Top 10 valores

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

Se muestra la correlación que tiene la variable **Radius** con las demás variables.

# Mapa de calor de la relación que existe entre variables

plt.figure(figsize=(14,7))

MatrizInf = np.triu(CorrBCancer)

sns.heatmap(CorrBCancer, cmap='RdBu\_r', annot=True, mask=MatrizInf)

plt.show()

Gráfico

Descripción generada automáticamente

## Selección de variables

1. Textura [Posición 3]
2. Área [Posición 5]
3. Smoothness [Pos. 6]
4. Compactness [Pos. 7]
5. Symmetry [Posición 10]
6. FractalDimension [Pos. 11]

MatrizVariables = np.array(BCancer[['Texture', 'Area', 'Smoothness', 'Compactness', 'Symmetry', 'FractalDimension']])

pd.DataFrame(MatrizVariables)

#MatrizHipoteca = BCancer.iloc[:, 0:9].values     #iloc para seleccionar filas y columnas según su posición

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

# Aplicación de los algoritmos

## Estandarización

Cuando se trabaja con clustering, dado que son algoritmos basados en distancias, es fundamental escalar los datos para que cada una de las variables contribuyan por igual en el análisis.

from sklearn.preprocessing import StandardScaler, MinMaxScaler

estandarizar = StandardScaler()                               # Se instancia el objeto StandardScaler o MinMaxScaler

MEstandarizada = estandarizar.fit\_transform(MatrizVariables)   # Se calculan la media y desviación y se escalan los datos

pd.DataFrame(MEstandarizada)  # Matriz estandarizada

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

## Algoritmo: Ascendente Jerárquico

#Se importan las bibliotecas de clustering jerárquico para crear el árbol

import scipy.cluster.hierarchy as shc

from sklearn.cluster import AgglomerativeClustering

plt.figure(figsize=(10,7))

plt.title("Pacientes con cáncer de mama")

plt.xlabel('Observaciones')

plt.ylabel('Distancia')

Arbol = shc.dendrogram(shc.linkage(MEstandarizada, method='complete', metric='euclidean'))

#plt.axhline(y=7, color='orange', linestyle='--')

#Probar con otras mediciones de distancia (chebyshev, cityblock)

Gráfico

Descripción generada automáticamente

### Se crean etiquetas en los clústeres

#Se crean las etiquetas de los elementos en los clústeres

MJerarquico = AgglomerativeClustering(n\_clusters=4, linkage='complete', affinity='euclidean')

MJerarquico.fit\_predict(MEstandarizada)

MJerarquico.labels\_

Patrón de fondo

Descripción generada automáticamente

BCancer['clusterH'] = MJerarquico.labels\_

BCancer

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

# Cantidad de elementos en los clústeres

BCancer.groupby(['clusterH'])['clusterH'].count()

Texto

Descripción generada automáticamente con confianza baja

### Obtención de los centroides

CentroidesH = BCancer.groupby(['clusterH'])['Texture', 'Area', 'Smoothness', 'Compactness', 'Symmetry', 'FractalDimension'].mean()

CentroidesH

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

### Interpretación de los datos obtenidos

**NOTA IMPORTANTE:** Para determinar si el tumor es maligno o benigno, nos basamos principalmente en el área, es decir, el tamaño del tumor, pero habría que presentarle estos datos a un especialista en el campo de la Medicina, en este caso, a un Oncólogo.

n = 0

colum = 0

while n < numClustersH.size:

    print("- Clúster número:", n)

    print("a) Conformado por un grupo de", numClustersH[n], "pacientes.")

    print("b) Con una desviación estándar de textura de", CentroidesH.iloc[n][colum].round(4),"píxeles")

    print("c) Con un área promedio de tumor de", CentroidesH.iloc[n][colum+1].round(4),"píxeles")

    if (CentroidesH.iloc[n][colum+1].round(4) > 600):

        print("d) El tamaño del tumor es muy grande, por ende, posiblemente este grupo de usuarios tengan un tumor maligno")

    else:

        print("d) El tamaño del tumor es relativamente pequeño, por ende, posiblemente este grupo de usuarios tengan un tumor benigno")

    print("e) Con una suavidad promedio de", CentroidesH.iloc[n][colum+2].round(4),"píxeles")

    print("f) Con una relación compacta promedio de", CentroidesH.iloc[n][colum+3].round(4),"píxeles")

    print("g) Con una simetría promedio de", CentroidesH.iloc[n][colum+4].round(4),"píxeles")

    print("h) Y con una aproximación de frontera, dimensión fractal promedio de", CentroidesH.iloc[n][colum+5].round(4),"píxeles")

    n = n + 1

    print("--------------------------------------------------------------")

#### Clúster número: 0

1. Conformado por un grupo de 23 pacientes.
2. Con una desviación estándar de textura de 20.1335 píxeles
3. Con un área promedio de tumor de 775.5435 píxeles
4. **El tamaño del tumor es muy grande, por ende, posiblemente este grupo de usuarios tengan un tumor maligno**
5. Con una suavidad promedio de 0.1243 píxeles
6. Con una relación compacta promedio de 0.2422 píxeles
7. Con una simetría promedio de 0.2408 píxeles
8. Y con una aproximación de frontera, dimensión fractal promedio de 0.0778 píxeles

#### Clúster número: 1

1. Conformado por un grupo de 88 pacientes.
2. Con una desviación estándar de textura de 22.5406 píxeles
3. Con un área promedio de tumor de 1243.7284 píxeles
4. **El tamaño del tumor es muy grande, por ende, posiblemente este grupo de usuarios tengan un tumor maligno**
5. Con una suavidad promedio de 0.0984 píxeles
6. Con una relación compacta promedio de 0.1371 píxeles
7. Con una simetría promedio de 0.1826 píxeles
8. Y con una aproximación de frontera, dimensión fractal promedio de 0.0589 píxeles

#### Clúster número: 2

1. Conformado por un grupo de 248 pacientes.
2. Con una desviación estándar de textura de 18.1675 píxeles
3. Con un área promedio de tumor de 561.3367 píxeles
4. **El tamaño del tumor es relativamente pequeño, por ende, posiblemente este grupo de usuarios tengan un tumor benigno**
5. Con una suavidad promedio de 0.1033 píxeles
6. Con una relación compacta promedio de 0.1142 píxeles
7. Con una simetría promedio de 0.1905 píxeles
8. Y con una aproximación de frontera, dimensión fractal promedio de 0.0657 píxeles

#### Clúster número: 3

1. Conformado por un grupo de 210 pacientes.
2. Con una desviación estándar de textura de 19.1601 píxeles
3. Con un área promedio de tumor de 505.4038 píxeles
4. **El tamaño del tumor es relativamente pequeño, por ende, posiblemente este grupo de usuarios tengan un tumor benigno**
5. Con una suavidad promedio de 0.0842 píxeles
6. Con una relación compacta promedio de 0.0638 píxeles
7. Con una simetría promedio de 0.163 píxeles
8. Y con una aproximación de frontera, dimensión fractal promedio de 0.0593 píxeles

### Visualización gráfica

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

## Algoritmo K-Means

### Método del codo

# Se importan las bibliotecas

from sklearn.cluster import KMeans

from sklearn.metrics import pairwise\_distances\_argmin\_min

#Definición de k clústeres para K-Means

# Se utiliza random\_state para inicializar el generador interno de números aleatorios

SSE = []

for i in range (2,12):

  km = KMeans(n\_clusters=i, random\_state=0)

  km.fit(MEstandarizada)

  SSE.append(km.inertia\_)

#Se grafica SSE en función de k

plt.figure(figsize=(10,7))

plt.plot(range(2,12), SSE, marker='o')

plt.xlabel('Cantidad de clústeres \*k\*')

plt.ylabel('SSE')

plt.title('Elbow Method')

plt.show()

Gráfico, Gráfico de líneas

Descripción generada automáticamente

**Observación.** En la práctica, puede que no exista un codo afilado (codo agudo) y, como método heurístico, ese "codo" no siempre puede identificarse sin ambigüedades.

!pip install kneed #Función que nos permite calcular el número estimado de clústers

from kneed import KneeLocator

kl = KneeLocator(range(2,12), SSE, curve="convex", direction="decreasing")

kl.elbow #El punto de inflexión se encuentra en el cluster 5

plt.style.use('ggplot')

kl.plot\_knee()

Gráfico, Gráfico de líneas

Descripción generada automáticamente

### Creando las etiquetas en los clústeres

MParticional = KMeans(n\_clusters=5, random\_state=0).fit(MEstandarizada)

MParticional.predict(MEstandarizada)

MParticional.labels\_

Patrón de fondo

Descripción generada automáticamente

BCancer['clusterP'] = MParticional.labels\_

BCancer

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

#Cantidad de elementos en los clusters

BCancer.groupby(['clusterP'])['clusterP'].count()

Imagen que contiene Texto

Descripción generada automáticamente

### Obtención de los centroides

CentroidesP = BCancer.groupby(['clusterP'])['Texture', 'Area', 'Smoothness', 'Compactness', 'Symmetry', 'FractalDimension'].mean()

CentroidesP

Captura de pantalla de computadora

Descripción generada automáticamente con confianza media

### Interpretación de los clústeres obtenidos

**NOTA IMPORTANTE:** Para determinar si el tumor es maligno o benigno, nos basamos principalmente en el área, es decir, el tamaño del tumor, pero habría que presentarle estos datos a un especialista en el campo de la Medicina, en este caso, a un Oncólogo.

n = 0

colum = 0

while n < numClustersP.size:

    print("- Clúster número:", n)

    print("a) Conformado por un grupo de", numClustersP[n], "pacientes.")

    print("b) Con una desviación estándar de textura de", CentroidesP.iloc[n][colum].round(4),"píxeles")

    print("c) Con un área promedio de tumor de", CentroidesP.iloc[n][colum+1].round(4),"píxeles")

    if (CentroidesP.iloc[n][colum+1].round(4) > 600):

        print("d) El tamaño del tumor es muy grande, por ende, posiblemente este grupo de usuarios tengan un tumor maligno")

    else:

        print("d) El tamaño del tumor es relativamente pequeño, por ende, posiblemente este grupo de usuarios tengan un tumor benigno")

    print("e) Con una suavidad promedio de", CentroidesP.iloc[n][colum+2].round(4),"píxeles")

    print("f) Con una relación compacta promedio de", CentroidesP.iloc[n][colum+3].round(4),"píxeles")

    print("g) Con una simetría promedio de", CentroidesP.iloc[n][colum+4].round(4),"píxeles")

    print("h) Y con una aproximación de frontera, dimensión fractal promedio de", CentroidesP.iloc[n][colum+5].round(4),"píxeles")

    n = n + 1

    print("--------------------------------------------------------------")

#### Clúster número: 0

1. Conformado por un grupo de 172 pacientes.
2. Con una desviación estándar de textura de 16.2974 píxeles
3. Con un área promedio de tumor de 514.2866 píxeles
4. **El tamaño del tumor es relativamente pequeño, por ende, posiblemente este grupo de usuarios tengan un tumor benigno**
5. Con una suavidad promedio de 0.0859 píxeles
6. Con una relación compacta promedio de 0.0627 píxeles
7. Con una simetría promedio de 0.1649 píxeles
8. Y con una aproximación de frontera, dimensión fractal promedio de 0.0591 píxeles

#### Clúster número: 1

1. Conformado por un grupo de 100 pacientes.
2. Con una desviación estándar de textura de 21.8375 píxeles
3. Con un área promedio de tumor de 1228.067 píxeles
4. **El tamaño del tumor es muy grande, por ende, posiblemente este grupo de usuarios tengan un tumor maligno**
5. Con una suavidad promedio de 0.1 píxeles
6. Con una relación compacta promedio de 0.1407 píxeles
7. Con una simetría promedio de 0.1874 píxeles
8. Y con una aproximación de frontera, dimensión fractal promedio de 0.0592 píxeles

#### Clúster número: 2

1. Conformado por un grupo de 56 pacientes.
2. Con una desviación estándar de textura de 20.3646 píxeles
3. Con un área promedio de tumor de 705.2839 píxeles
4. **El tamaño del tumor es muy grande, por ende, posiblemente este grupo de usuarios tengan un tumor maligno**
5. Con una suavidad promedio de 0.1156 píxeles
6. Con una relación compacta promedio de 0.2047 píxeles
7. Con una simetría promedio de 0.2261 píxeles
8. Y con una aproximación de frontera, dimensión fractal promedio de 0.0759 píxeles

#### Clúster número: 3

1. Conformado por un grupo de 156 pacientes.
2. Con una desviación estándar de textura de 17.7346 píxeles
3. Con un área promedio de tumor de 476.3372 píxeles
4. **El tamaño del tumor es relativamente pequeño, por ende, posiblemente este grupo de usuarios tengan un tumor benigno**
5. Con una suavidad promedio de 0.1047 píxeles
6. Con una relación compacta promedio de 0.1071 píxeles
7. Con una simetría promedio de 0.188 píxeles
8. Y con una aproximación de frontera, dimensión fractal promedio de 0.0664 píxeles

#### Clúster número: 4

1. Conformado por un grupo de 85 pacientes.
2. Con una desviación estándar de textura de 24.4927 píxeles
3. Con un área promedio de tumor de 559.5694 píxeles
4. **El tamaño del tumor es relativamente pequeño, por ende, posiblemente este grupo de usuarios tengan un tumor benigno**
5. Con una suavidad promedio de 0.085 píxeles
6. Con una relación compacta promedio de 0.0746 píxeles
7. Con una simetría promedio de 0.1645 píxeles
8. Y con una aproximación de frontera, dimensión fractal promedio de 0.0594 píxeles

### Visualización gráfica

# Gráfica de los elementos y los centros de los clusters

from mpl\_toolkits.mplot3d import Axes3D

plt.rcParams['figure.figsize'] = (10, 7)

plt.style.use('ggplot')

colores=['red', 'blue', 'green', 'yellow','cyan']

asignar=[]

for row in MParticional.labels\_:

    asignar.append(colores[row])

fig = plt.figure()

ax = Axes3D(fig)

ax.scatter(MEstandarizada[:, 0],

           MEstandarizada[:, 1],

           MEstandarizada[:, 2], marker='o', c=asignar, s=60)

ax.scatter(MParticional.cluster\_centers\_[:, 0],

           MParticional.cluster\_centers\_[:, 1],

           MParticional.cluster\_centers\_[:, 2], marker='o', c=colores, s=1000)

plt.show()

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

# Conclusiones

En esta práctica pude aprender cómo obtener grupos de pacientes con características similares, diagnosticadas con un tumor de mama, a través de clustering jerárquico y particional. Generé los clústeres a partir de una fuente de datos y en total obtuve cuatro grupos con el algoritmo de ascendente jerárquico y cinco con el algoritmo K-Means.

Como ambos algoritmos están basados en distancias, tuve que escalar los datos, de tal manera que cada variable contribuyera de igual manera en el análisis, es decir, que ninguna variable pesara más que la otra.

Finalmente, implementé ambos algoritmos, los cuales pude analizar y dar mis conclusiones sobre si cada grupo tenía potencialmente un tumor maligno o benigno, pero basándome en los criterios establecidos en clase, los cuales eran determinar si el tumor era potencialmente benigno o maligno basándonos en el tamaño de este, es decir, en el área del tumor.

En conclusión, pude aprender y visualizar de mejor manera la aplicación que tienen las distancias en el análisis de datos, de forma específica en el tema de clustering, y pude poner en práctica los conocimientos adquiridos en las prácticas pasadas.

# Link de Google Colab

🔗 [OCG-Práctica6-JerárquicoParticional.ipynb - Colaboratory (google.com)](https://colab.research.google.com/drive/1P9Q8J2ZsKLEN8LpWq1hJT0d8Y9FChV5l#scrollTo=Jk9dzAizg4a1)