Contenido

[Objetivo 1](#_Toc84163202)

[Fuentes de datos 1](#_Toc84163203)

[Datos de las distancias que se van a calcular 1](#_Toc84163204)

[Preparación del entorno de ejecución 2](#_Toc84163205)

[1) Importar las bibliotecas necesarias 2](#_Toc84163206)

[2) Importar los datos 2](#_Toc84163207)

[Matrices de distancia 2](#_Toc84163208)

[a) Distancia Euclidiana 2](#_Toc84163209)

[b) Distancia de Chebyshev 3](#_Toc84163210)

[c) Distancia de Manhattan 4](#_Toc84163211)

[d) Distancia de Minkowski 5](#_Toc84163212)

[Otras mediciones 5](#_Toc84163213)

[Distancia Euclidiana 5](#_Toc84163214)

[Distancia de Chebyshev 5](#_Toc84163215)

[Distancia de Manhattan 5](#_Toc84163216)

[Distancia de Minkowski 6](#_Toc84163217)

[Conclusiones 6](#_Toc84163218)

[Link de Google Colab 6](#_Toc84163219)

# Objetivo

* Obtener las matrices de distancia (Euclidiana, Chebyshev, Manhattan, Minkowski) en Google Colab a partir de una matriz de datos.

# Fuentes de datos

* Ingresos: son ingresos mensuales de 1 o 2 personas, si están casados.
* Gastos\_comunes: son gastos mensuales de 1 o 2 personas, si están casados.
* Pago\_coche
* Gastos\_otros
* Ahorros
* Vivienda: valor de la vivienda.
* Estado\_civil: 0-soltero, 1-casado, 2-divorciado
* Hijos: cantidad de hijos menores (no trabajan).
* Trabajo: 0-sin trabajo, 1-autonomo, 2-asalariado, 3-empresario, 4-autonomos, 5-asalariados, 6-autonomo y asalariado, 7-empresario y autonomo, 8-empresarios o empresario y autónomo
* Comprar: 0-alquilar, 1-comprar casa a través de crédito hipotecario con tasa fija a 30 años.

## Datos de las distancias que se van a calcular

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Ingresos mensuales | Gastos Comunes | Pago Coche | Otros gastos | Ahorros | Vivienda | Estado Civil | Hijos | Trabajo | Comprar |
| 0 | 6,000 | 1,000 | 0 | 600 | 50,000 | 400,000 | Soltero | 2 | Asalariado | Comprar casa a través de crédito hipotecario con tasa fija a 30 años. |
| 1 | 6,745 | 944 | 123 | 429 | 43,240 | 636,897 | Casado | 3 | Autónomo y asalariado | Alquilar |

# 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 generar gráficas a partir de los datos

from scipy.spatial.distance import cdist    # Para el cálculo de distancias

## Importar los datos

Fuente de datos: Hipoteca.csv

# Para importar los datos desde Drive

from google.colab import drive

drive.mount('/content/drive')

# Para importar los datos de manera local

#from google.colab import files

#files.upload()

Hipoteca = pd.read\_csv("/content/drive/MyDrive/Ingeniería en Computación/Hipoteca.csv") #Importamos y mostramos los datos

Hipoteca

Una pantalla de un video juego

Descripción generada automáticamente con confianza media

# Matrices de distancia

## Distancia Euclidiana

DstEuclidiana = cdist(Hipoteca, Hipoteca, metric='euclidean')

MEuclidiana = pd.DataFrame(DstEuclidiana)

print(MEuclidiana.round(2)) # round(n) Se redondea a solo n decimales

Un conjunto de letras blancas en un fondo blanco

Descripción generada automáticamente con confianza media

#Matriz de distancias de 10 objetos

DstEuclidiana = cdist(Hipoteca.iloc[0:10],Hipoteca.iloc[0:10],metric='euclidean')

MEuclidiana = pd.DataFrame(DstEuclidiana)

print(MEuclidiana)

Texto

Descripción generada automáticamente

#Matriz euclidiana entre dos objetos

from scipy.spatial import distance # Para el cálculo de distancias

#Calculando la distancia entre dos objetos

Objeto1 = Hipoteca.iloc[0]

Objeto2 = Hipoteca.iloc[1]

dstEuclidiana = distance.euclidean(Objeto1, Objeto2)

dstEuclidiana # DISTANCIA: 236994.70196398906

El espacio o distancia Euclidiana entre los primeros dos tipos de personas es de: 236,994.70196398906

Gráfico, Gráfico de líneas

Descripción generada automáticamente

## Distancia de Chebyshev

DstChebyshev = cdist(Hipoteca,Hipoteca,metric='chebyshev')

MChebyshev = pd.DataFrame(DstChebyshev)

print(MChebyshev)

Un conjunto de letras blancas en un fondo blanco

Descripción generada automáticamente con confianza media

#Matriz de distancias de 10 objetos

DstChebyshev = cdist(Hipoteca.iloc[0:10],Hipoteca.iloc[0:10],metric='chebyshev')

MChebyshev = pd.DataFrame(DstChebyshev)

print(MChebyshev)

Un conjunto de letras blancas en un fondo blanco

Descripción generada automáticamente con confianza media

Objeto1 = Hipoteca.iloc[0]

Objeto2 = Hipoteca.iloc[1]

#Calculando la distancia entre dos objetos

dstChebyshev = distance.chebyshev(Objeto1, Objeto2)

dstChebyshev # DISTANCIA: 236897

La distancia de Chebyshev entre los primeros dos tipos de personas es de: 236,897

Gráfico

Descripción generada automáticamente

## Distancia de Manhattan

DstManhattan = cdist(Hipoteca,Hipoteca,metric='cityblock')

MManhattan = pd.DataFrame(DstManhattan)

print(MManhattan)

Un conjunto de letras blancas en un fondo blanco

Descripción generada automáticamente con confianza media

#Matriz de distancias de 10 objetos

DstManhattan = cdist(Hipoteca.iloc[0:10],Hipoteca.iloc[0:10],metric='cityblock')

MManhattan = pd.DataFrame(DstManhattan)

print(MManhattan)

Texto

Descripción generada automáticamente con confianza media

Objeto1 = Hipoteca.iloc[0]

Objeto2 = Hipoteca.iloc[1]

#Calculando la distancia entre dos objetos

dstManhattan = distance.cityblock(Objeto1, Objeto2)

dstManhattan # DISTANCIA: 244759

La distancia de Manhattan entre los primeros dos tipos de personas es de: 244,759

Gráfico

Descripción generada automáticamente

## Distancia de Minkowski

DstMinkowski = cdist(Hipoteca,Hipoteca,metric='minkowski', p=1.5) # p es el orden para calcular la distancia

MMinkowski = pd.DataFrame(DstMinkowski)

print(MMinkowski)

Texto

Descripción generada automáticamente

#Matriz de distancias de 10 objetos

DstMinkowski = cdist(Hipoteca.iloc[0:10],Hipoteca.iloc[0:10],metric='minkowski', p=1.5)

MMinkowski = pd.DataFrame(DstMinkowski)

print(MMinkowski)

Texto

Descripción generada automáticamente

Objeto1 = Hipoteca.iloc[0]

Objeto2 = Hipoteca.iloc[1]

#Calculando la distancia entre dos objetos

dstMinkowski = distance.minkowski(Objeto1, Objeto2, p=1.5)

dstMinkowski # DISTANCIA: 237690.9959246789

La distancia de Minkowski entre los primeros dos tipos de personas es de: 237,690.9959246789

Imagen que contiene Diagrama

Descripción generada automáticamente

# Otras mediciones

## Distancia Euclidiana

from scipy.spatial import distance

E1 = (10000,1,0,0,0,0,7,15,1)

E2 = (20000,0,1,1,0,1,3,3,0)

dstEuclidiana = distance.euclidean(E1,E2)

dstEuclidiana #DISTANCIA EUCLIDIANA: 10000.008249996597

## Distancia de Chebyshev

from scipy.spatial import distance

E1 = (10000,1,0,0,0,0,7,15,1)

E2 = (20000,0,1,1,0,1,3,3,0)

dstChebyshev = distance.chebyshev(E1,E2)

dstChebyshev #DISTANCIA DE CHEBYSHEV: 10000

## Distancia de Manhattan

from scipy.spatial import distance

E1 = (10000,1,0,0,0,0,7,15,1)

E2 = (20000,0,1,1,0,1,3,3,0)

dstManhattan = distance.cityblock(E1,E2)

dstManhattan #DISTANCIA DE MANHATTAN: 10021

## Distancia de Minkowski

from scipy.spatial import distance

E1 = (10000,1,0,0,0,0,7,15,1)

E2 = (20000,0,1,1,0,1,3,3,0)

dstMinkowski = distance.minkowski(E1,E2, 1.5) # Orden 1.5

dstMinkowski #DISTANCIA DE MINKOWSKI: 10000.363791487287

# Conclusiones

Se puede concluir que dependiendo de la situación que se esté trabajando, es la métrica de distancia que usaremos. En algunas ocasiones usaremos la distancia Euclidiana y en otras la distancia de Minkowksi.

Se calcularon las matrices de distancias, desde la Euclidiana hasta la de Minkowski. Se observó que la distancia entre las primeras dos personas era algo grande, en comparación con las otras observadas en las matrices de distancias, por lo que se puede concluir que estas dos personas son bastante diferentes en cuanto a sus características, desde su estado civil, sus ingresos, sus objetivos, etc.

Hay que recordar que *“Una buena métrica de distancia ayuda a mejorar significativamente el rendimiento del proceso de clasificación, clusterización, recuperación de información y otras aplicaciones”.*

# Link de Google Colab

🔗 [OCG-Práctica3-MétricasDeDistancia.ipynb - Colaboratory (google.com)](https://colab.research.google.com/drive/1kv1eE8tiUHcGXu7lhe1eqqw3-11PVRkd#scrollTo=Fav1N6Pp_dW9)