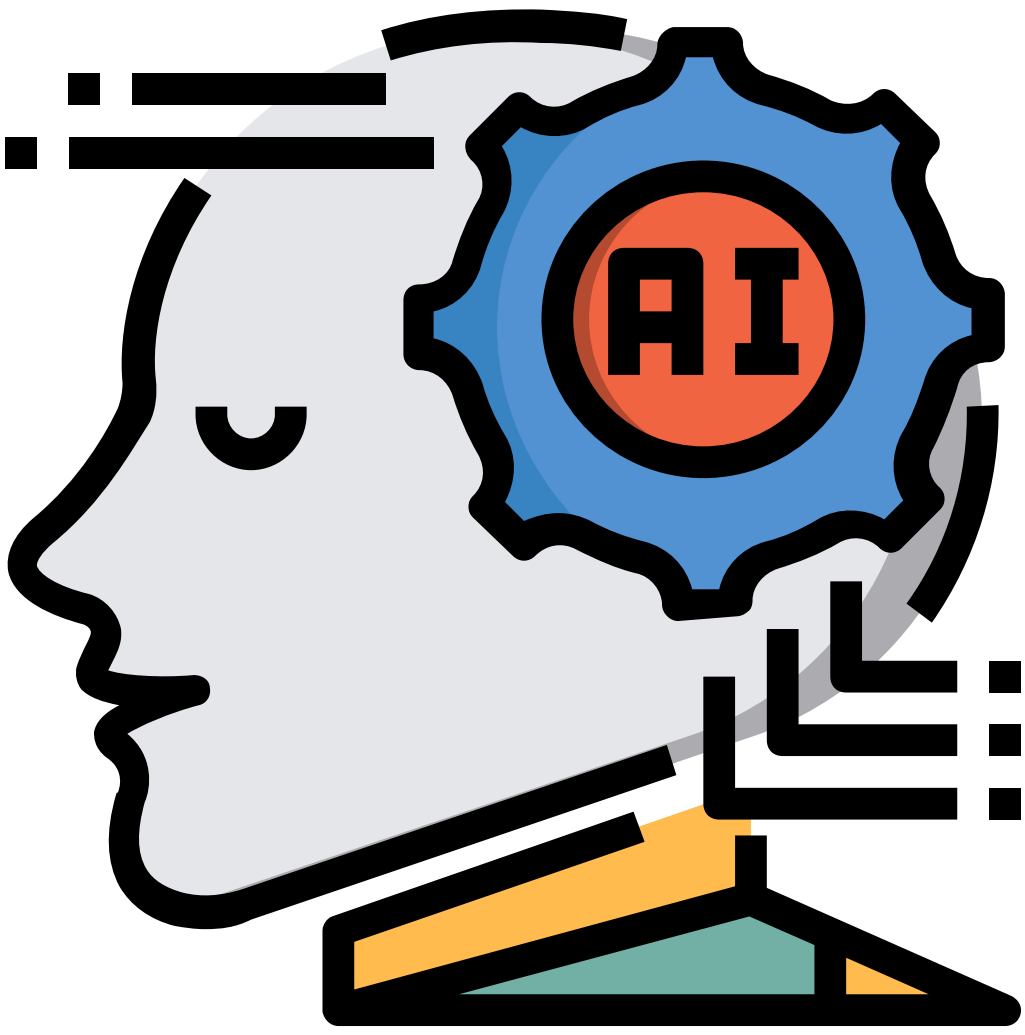
Universidad Nacional Autónoma de México
Facultad de Ingeniería

Inteligencia Artificial

# PRÁCTICA 5. CLUSTERING PARTICIONAL

Casasola García Oscar
316123747
oscar.casasola.g7@gmail.com
Grupo 03




Profesor: Dr. Guillermo Gilberto Molero Castillo
Semestre 2022-1

Contenido

## Contexto

**Objetivo**: Obtener clústeres de casos de usuarios, con características similares, evaluados para la adquisición de una casa a través de un crédito hipotecario con tasa fija a 30 años.

## Fuente 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.

## Preparación del entorno de ejecución

### 1) 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
```

### 2) Importar los datos

Fuente de datos: Hipoteca.csv

```
from google.colab import files
files.upload()

# Para importar los datos desde Drive
#from google.colab import drive
#drive.mount('/content/drive')
```

```
Hipoteca = pd.read_csv("Hipoteca.csv")
Hipoteca
```

| | ingresos | gastos_comunes | pago_coche | gastos_otros | ahorros | vivienda | estado_civil | hijos | trabajo | comprar |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 6000 | 1000 | 0 | 600 | 50000 | 400000 | 0 | 2 | 2 | 1 |
| **1** | 6745 | 944 | 123 | 429 | 43240 | 636897 | 1 | 3 | 6 | 0 |
| **2** | 6455 | 1033 | 98 | 795 | 57463 | 321779 | 2 | 1 | 8 | 1 |
| **3** | 7098 | 1278 | 15 | 254 | 54506 | 660933 | 0 | 0 | 3 | 0 |
| **4** | 6167 | 863 | 223 | 520 | 41512 | 348932 | 0 | 0 | 3 | 1 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **197** | 3831 | 690 | 352 | 488 | 10723 | 363120 | 0 | 0 | 2 | 0 |
| **198** | 3961 | 1030 | 270 | 475 | 21880 | 280421 | 2 | 3 | 8 | 0 |
| **199** | 3184 | 955 | 276 | 684 | 35565 | 388025 | 1 | 3 | 8 | 0 |
| **200** | 3334 | 867 | 369 | 652 | 19985 | 376892 | 1 | 2 | 5 | 0 |
| **201** | 3988 | 1157 | 105 | 382 | 11980 | 257580 | 0 | 0 | 4 | 0 |

202 rows × 10 columns

```
Hipoteca.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 202 entries, 0 to 201
Data columns (total 10 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   ingresos        202 non-null    int64
 1   gastos_comunes  202 non-null    int64
 2   pago_coche      202 non-null    int64
 3   gastos_otros    202 non-null    int64
 4   ahorros         202 non-null    int64
 5   vivienda        202 non-null    int64
 6   estado_civil    202 non-null    int64
 7   hijos           202 non-null    int64
 8   trabajo         202 non-null    int64
 9   comprar         202 non-null    int64
dtypes: int64(10)
memory usage: 15.9 KB
```

```
print(Hipoteca.groupby('comprar').size())
```

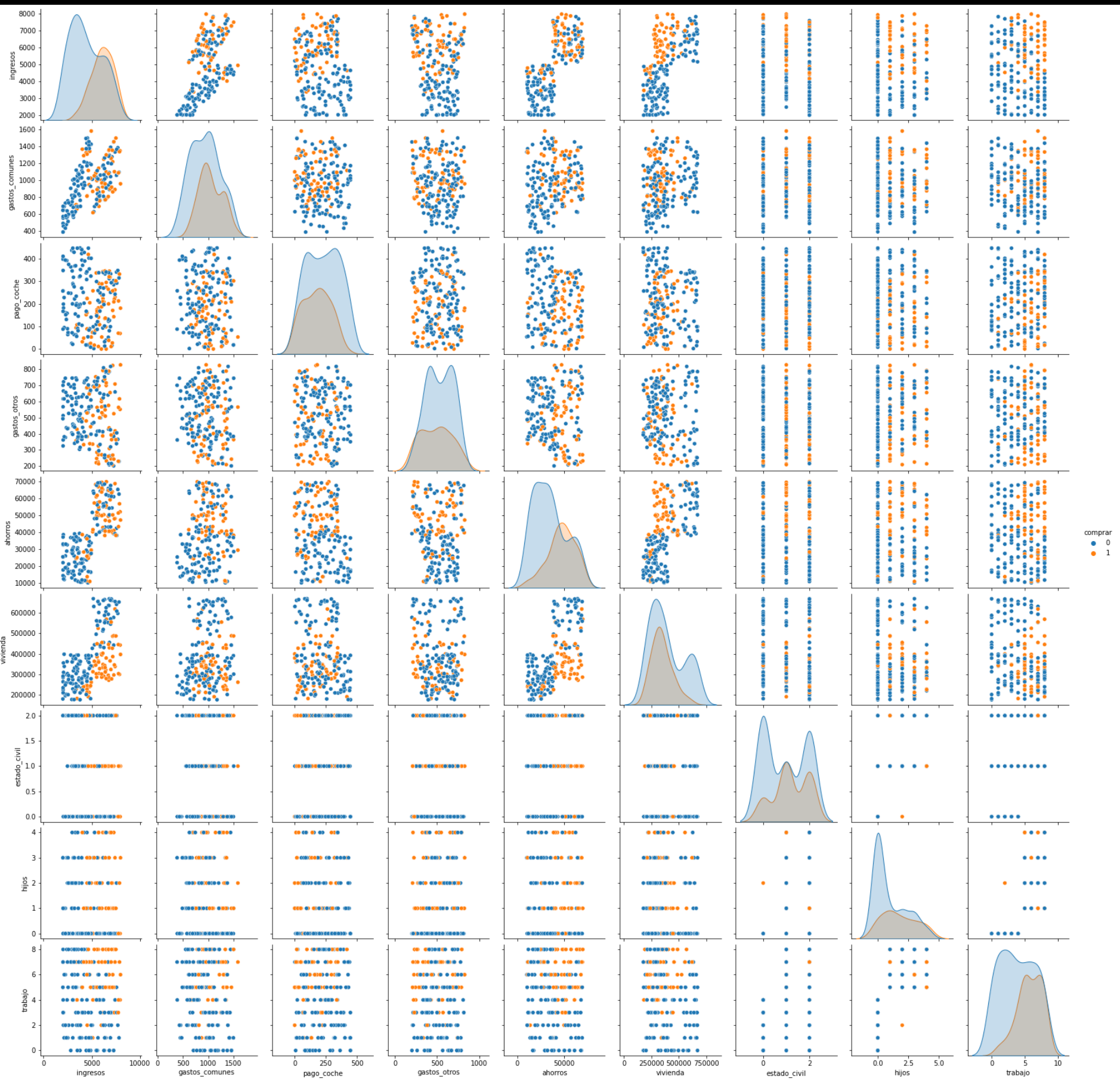
```
comprar
0    135
1     67
dtype: int64
```

Se puede observar que **135 usuarios desean alquilar**, mientras que **67 usuarios desean comprar** una casa a través de un crédito hipotecario con tasa fija a 30 años.

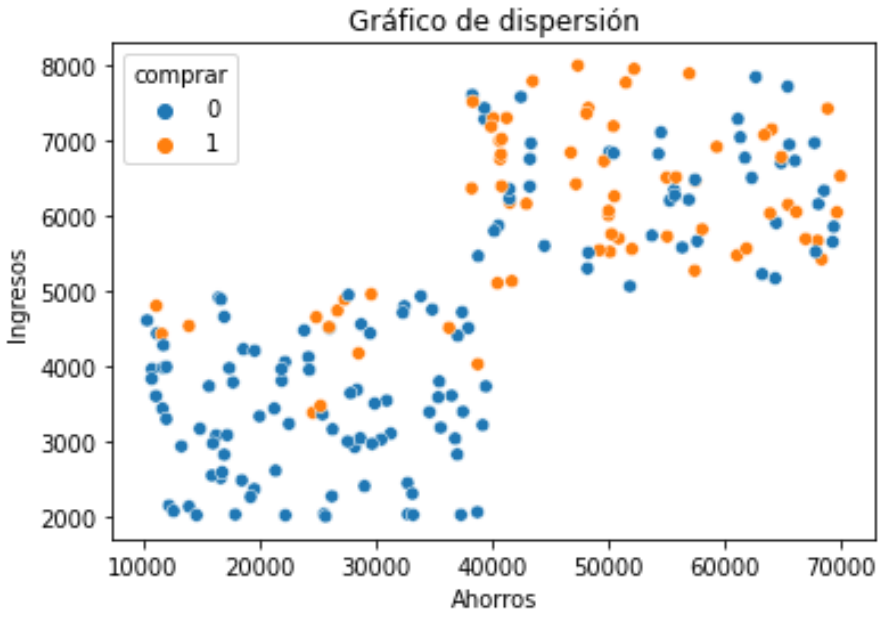
# Selección de características

## Evaluación visual

```
sns.pairplot(Hipoteca, hue='comprar')
plt.show()
```



```
sns.scatterplot(x='ahorros', y ='ingresos', data=Hipoteca, hue='comprar')
plt.title('Gráfico de dispersión')
plt.xlabel('Ahorros')
plt.ylabel('Ingresos')
plt.show()
```




## Matriz de correlaciones

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

```
CorrHipoteca = Hipoteca.corr(method='pearson')
CorrHipoteca
```

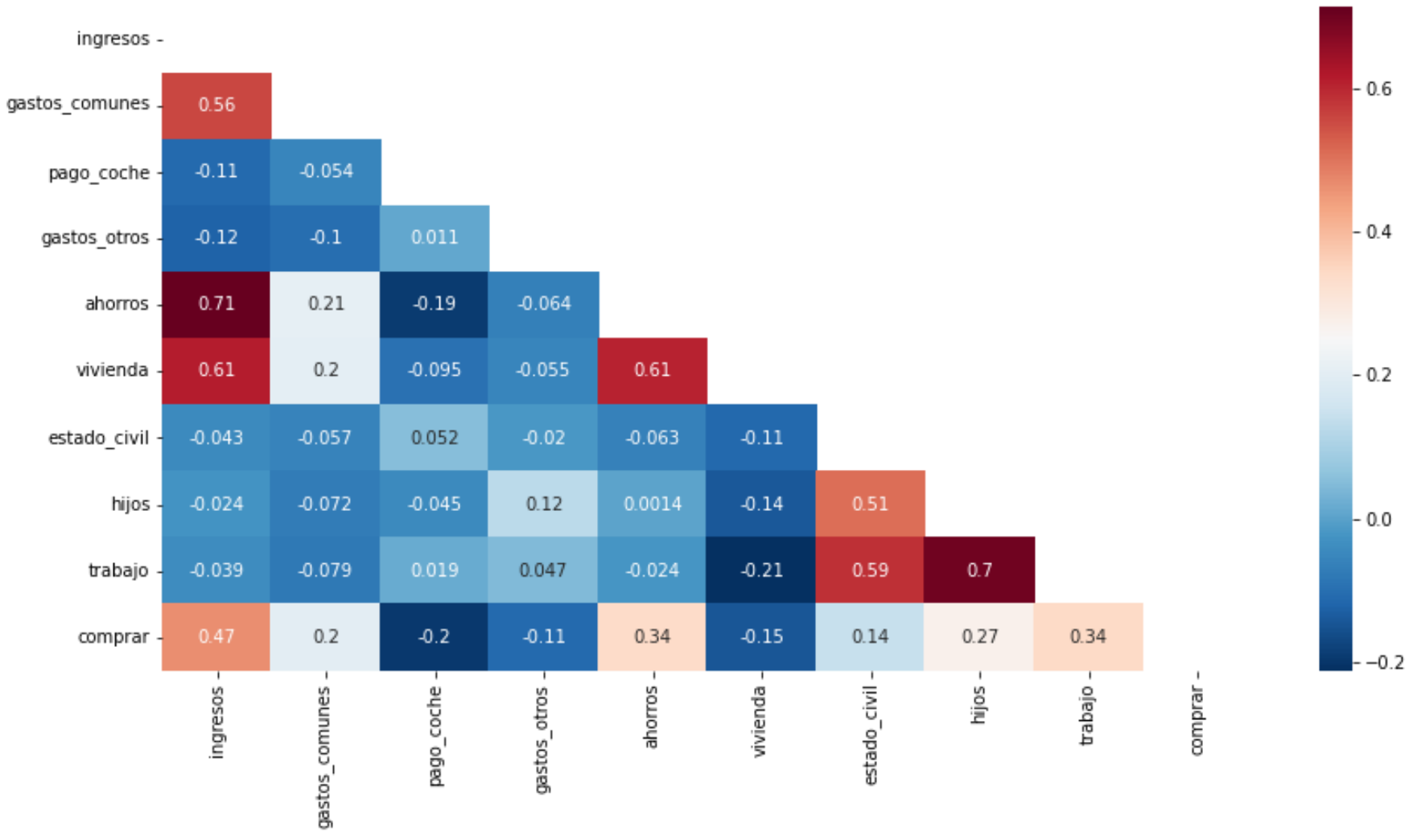
| | ingresos | gastos_comunes | pago_coche | gastos_otros | ahorros | vivienda | estado_civil | hijos | trabajo | comprar |
|---|---|---|---|---|---|---|---|---|---|---|
| **ingresos** | 1.000000 | 0.560211 | -0.109780 | -0.124105 | 0.712889 | 0.614721 | -0.042556 | -0.024483 | -0.038852 | 0.467123 |
| **gastos_comunes** | 0.560211 | 1.000000 | -0.054400 | -0.099881 | 0.209414 | 0.204781 | -0.057152 | -0.072321 | -0.079095 | 0.200191 |
| **pago_coche** | -0.109780 | -0.054400 | 1.000000 | 0.010602 | -0.193299 | -0.094631 | 0.052239 | -0.044858 | 0.018946 | -0.196468 |
| **gastos_otros** | -0.124105 | -0.099881 | 0.010602 | 1.000000 | -0.064384 | -0.054577 | -0.020226 | 0.124845 | 0.047313 | -0.110330 |
| **ahorros** | 0.712889 | 0.209414 | -0.193299 | -0.064384 | 1.000000 | 0.605836 | -0.063039 | 0.001445 | -0.023829 | 0.340778 |
| **vivienda** | 0.614721 | 0.204781 | -0.094631 | -0.054577 | 0.605836 | 1.000000 | -0.113420 | -0.141924 | -0.211790 | -0.146092 |
| **estado_civil** | -0.042556 | -0.057152 | 0.052239 | -0.020226 | -0.063039 | -0.113420 | 1.000000 | 0.507609 | 0.589512 | 0.142799 |
| **hijos** | -0.024483 | -0.072321 | -0.044858 | 0.124845 | 0.001445 | -0.141924 | 0.507609 | 1.000000 | 0.699916 | 0.272883 |
| **trabajo** | -0.038852 | -0.079095 | 0.018946 | 0.047313 | -0.023829 | -0.211790 | 0.589512 | 0.699916 | 1.000000 | 0.341537 |
| **comprar** | 0.467123 | 0.200191 | -0.196468 | -0.110330 | 0.340778 | -0.146092 | 0.142799 | 0.272883 | 0.341537 | 1.000000 |

```
print(CorrHipoteca['ingresos'].sort_values(ascending=False)[:10], '\n')   #Top 10 valores
```

```
ingresos          1.000000
ahorros           0.712889
vivienda          0.614721
gastos_comunes    0.560211
comprar           0.467123
hijos            -0.024483
trabajo          -0.038852
estado_civil     -0.042556
pago_coche       -0.109780
gastos_otros     -0.124105
Name: ingresos, dtype: float64
```

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

```
# Mapa de calor de la relación que existe entre variables
plt.figure(figsize=(14,7))
MatrizInf = np.triu(CorrHipoteca)
sns.heatmap(CorrHipoteca, cmap='RdBu_r', annot=True, mask=MatrizInf)
plt.show()
```




## Selección de variables

a) A pesar de existir 2 correlaciones altas, entre 'ingresos' y 'ahorros' (0.71) y 'trabajo' e 'hijos' (0.69); éstas se tomarán en cuenta para obtener una segmentación que combine las variables mediante la similitud de los elementos.

b) Se suprimirá la variable 'comprar' debido a que representa inherentemente un agrupamiento, y fue un campo calculado con base a un análisis hipotecario preliminar.

```
MatrizHipoteca = np.array(Hipoteca[['ingresos', 'gastos_comunes', 'pago_coche', 'gastos_otros', 'ahorros', 'vivienda', 'estado_civil', 'hijos', 'trabajo']])
pd.DataFrame(MatrizHipoteca)
#MatrizHipoteca = Hipoteca.iloc[: , 0:9].values     #iloc para seleccionar filas y columnas según su posición
```

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 6000 | 1000 | 0 | 600 | 50000 | 400000 | 0 | 2 | 2 |
| **1** | 6745 | 944 | 123 | 429 | 43240 | 636897 | 1 | 3 | 6 |
| **2** | 6455 | 1033 | 98 | 795 | 57463 | 321779 | 2 | 1 | 8 |
| **3** | 7098 | 1278 | 15 | 254 | 54506 | 660933 | 0 | 0 | 3 |
| **4** | 6167 | 863 | 223 | 520 | 41512 | 348932 | 0 | 0 | 3 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **197** | 3831 | 690 | 352 | 488 | 10723 | 363120 | 0 | 0 | 2 |
| **198** | 3961 | 1030 | 270 | 475 | 21880 | 280421 | 2 | 3 | 8 |
| **199** | 3184 | 955 | 276 | 684 | 35565 | 388025 | 1 | 3 | 8 |
| **200** | 3334 | 867 | 369 | 652 | 19985 | 376892 | 1 | 2 | 5 |
| **201** | 3988 | 1157 | 105 | 382 | 11980 | 257580 | 0 | 0 | 4 |

202 rows × 9 columns

# Aplicación del algoritmo

## Algoritmo: K-means

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(MatrizHipoteca)   # Se calculan la media y desviación y se escalan los datos

pd.DataFrame(MEstandarizada)  # Matriz estandarizada
```

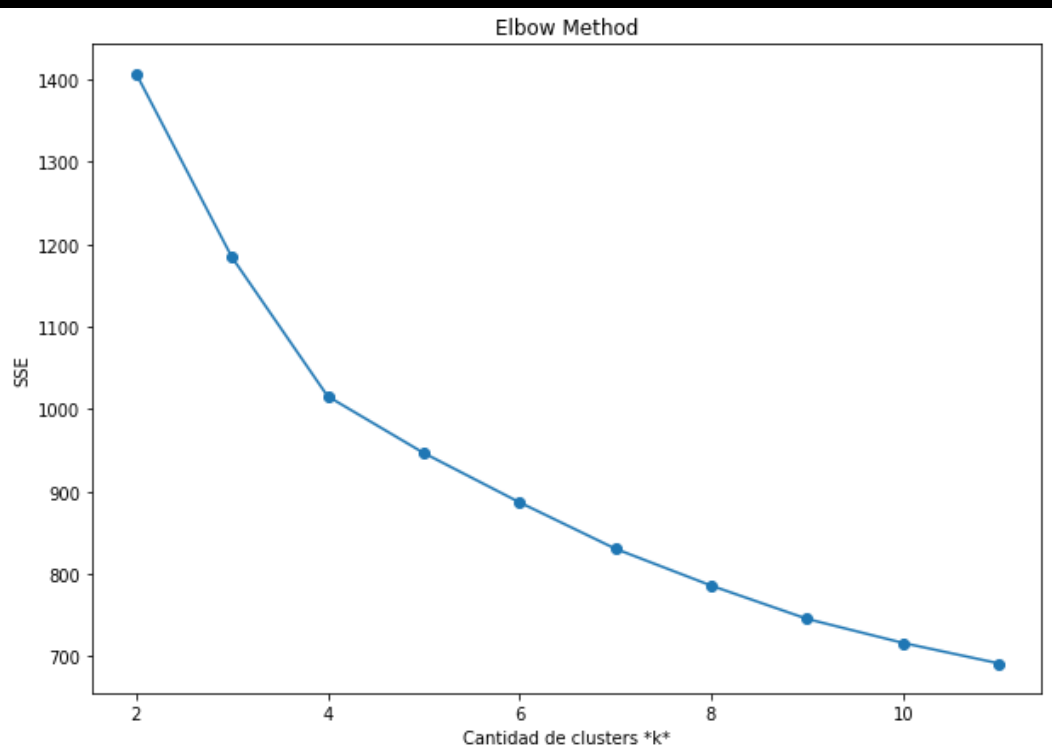
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 0.620129 | 0.104689 | -1.698954 | 0.504359 | 0.649475 | 0.195910 | -1.227088 | 0.562374 | -0.984420 |
| **1** | 1.063927 | -0.101625 | -0.712042 | -0.515401 | 0.259224 | 1.937370 | -0.029640 | 1.295273 | 0.596915 |
| **2** | 0.891173 | 0.226266 | -0.912634 | 1.667244 | 1.080309 | -0.379102 | 1.167809 | -0.170526 | 1.387582 |
| **3** | 1.274209 | 1.128886 | -1.578599 | -1.559015 | 0.909604 | 2.114062 | -1.227088 | -0.903426 | -0.589086 |
| **4** | 0.719611 | -0.400042 | 0.090326 | 0.027279 | 0.159468 | -0.179497 | -1.227088 | -0.903426 | -0.589086 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **197** | -0.671949 | -1.037402 | 1.125381 | -0.163554 | -1.617963 | -0.075199 | -1.227088 | -0.903426 | -0.984420 |
| **198** | -0.594508 | 0.215214 | 0.467439 | -0.241079 | -0.973876 | -0.683130 | 1.167809 | 1.295273 | 1.387582 |
| **199** | -1.057368 | -0.061099 | 0.515581 | 1.005294 | -0.183849 | 0.107880 | -0.029640 | 1.295273 | 1.387582 |
| **200** | -0.968013 | -0.385305 | 1.261783 | 0.814462 | -1.083273 | 0.026040 | -0.029640 | 0.562374 | 0.201581 |
| **201** | -0.578424 | 0.683102 | -0.856468 | -0.795686 | -1.545397 | -0.851037 | -1.227088 | -0.903426 | -0.193753 |

202 rows × 9 columns

```
#Se importan las bibliotecas
from sklearn.cluster import KMeans
from sklearn.metrics import pairwise_distances_argmin_min
```

```
#Definición de k clusters 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 clusters *k*')
plt.ylabel('SSE')
plt.title('Elbow Method')
plt.show()
```

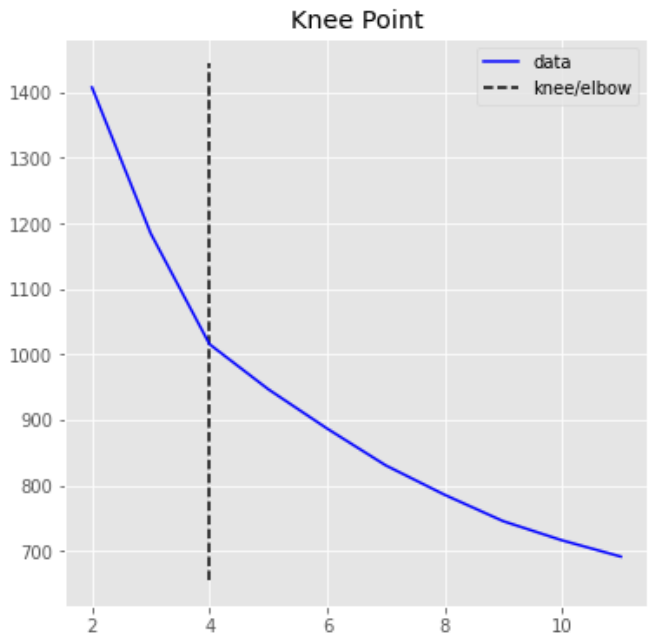



**Observación.** En la práctica, puede que no exista un codo afilado (codo agudo) y, como método heurístico, ese "codo" no siempre puede identificarse sin ambigüedades.

```
!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 #4
```

```
plt.style.use('ggplot')
kl.plot_knee()
```




```
#Se crean las etiquetas de los elementos en los clústeres
MParticional = KMeans(n_clusters=4, random_state=0).fit(MEstandarizada)
MParticional.predict(MEstandarizada)
MParticional.labels_
```

```
array([0, 2, 2, 0, 0, 2, 0, 0, 0, 2, 0, 0, 2, 2, 2, 2, 2, 2, 0, 2, 0, 2,
       0, 2, 0, 0, 2, 0, 0, 2, 2, 0, 2, 0, 0, 2, 0, 2, 2, 2, 0, 2, 2, 2,
       0, 0, 3, 2, 2, 0, 1, 1, 1, 1, 3, 1, 3, 3, 3, 3, 1, 1, 3, 1, 3, 1,
       1, 3, 1, 3, 1, 1, 1, 1, 3, 1, 3, 1, 1, 3, 1, 0, 3, 3, 1, 1, 3, 1,
       1, 3, 3, 1, 1, 3, 3, 1, 3, 3, 1, 3, 1, 2, 0, 2, 2, 0, 0, 2, 0, 2,
       2, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 0,
       0, 0, 0, 0, 0, 2, 2, 2, 0, 2, 0, 2, 0, 0, 2, 2, 0, 0, 0, 3, 1, 3,
       0, 3, 0, 1, 1, 3, 1, 1, 1, 1, 3, 1, 1, 3, 1, 1, 1, 3, 3, 1, 3, 1,
       3, 3, 1, 3, 1, 1, 1, 1, 3, 1, 3, 1, 0, 3, 1, 3, 3, 1, 1, 1, 3, 3,
       1, 1, 1, 3], dtype=int32)
```

```
Hipoteca = Hipoteca.drop(columns=['comprar'])
Hipoteca['clusterP'] = MParticional.labels_
Hipoteca
```

| | ingresos | gastos_comunes | pago_coche | gastos_otros | ahorros | vivienda | estado_civil | hijos | trabajo | clusterP |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 6000 | 1000 | 0 | 600 | 50000 | 400000 | 0 | 2 | 2 | 0 |
| **1** | 6745 | 944 | 123 | 429 | 43240 | 636897 | 1 | 3 | 6 | 2 |
| **2** | 6455 | 1033 | 98 | 795 | 57463 | 321779 | 2 | 1 | 8 | 2 |
| **3** | 7098 | 1278 | 15 | 254 | 54506 | 660933 | 0 | 0 | 3 | 0 |
| **4** | 6167 | 863 | 223 | 520 | 41512 | 348932 | 0 | 0 | 3 | 0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **197** | 3831 | 690 | 352 | 488 | 10723 | 363120 | 0 | 0 | 2 | 3 |
| **198** | 3961 | 1030 | 270 | 475 | 21880 | 280421 | 2 | 3 | 8 | 1 |
| **199** | 3184 | 955 | 276 | 684 | 35565 | 388025 | 1 | 3 | 8 | 1 |
| **200** | 3334 | 867 | 369 | 652 | 19985 | 376892 | 1 | 2 | 5 | 1 |
| **201** | 3988 | 1157 | 105 | 382 | 11980 | 257580 | 0 | 0 | 4 | 3 |

202 rows × 10 columns

```
#Cantidad de elementos en los clusters
numClusters = Hipoteca.groupby(['clusterP'])['clusterP'].count()
print(numClusters)
```

```
clusterP
0    49
1    56
2    54
3    43
Name: clusterP, dtype: int64
```

```
Hipoteca[Hipoteca.clusterP == 0] # Datos del primer clúster
```

| | ingresos | gastos_comunes | pago_coche | gastos_otros | ahorros | vivienda | estado_civil | hijos | trabajo | clusterP |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 6000 | 1000 | 0 | 600 | 50000 | 400000 | 0 | 2 | 2 | 0 |
| 3 | 7098 | 1278 | 15 | 254 | 54506 | 660933 | 0 | 0 | 3 | 0 |
| 4 | 6167 | 863 | 223 | 520 | 41512 | 348932 | 0 | 0 | 3 | 0 |
| 6 | 6830 | 1298 | 345 | 309 | 46761 | 429812 | 1 | 1 | 5 | 0 |
| 7 | 6470 | 1035 | 39 | 782 | 57439 | 606291 | 0 | 0 | 1 | 0 |
| 8 | 6251 | 1250 | 209 | 571 | 50503 | 291010 | 0 | 0 | 3 | 0 |
| 10 | 7273 | 1455 | 303 | 201 | 39340 | 577972 | 0 | 0 | 0 | 0 |
| 11 | 5058 | 1012 | 74 | 463 | 51836 | 427334 | 2 | 0 | 2 | 0 |
| 18 | 7705 | 1387 | 348 | 366 | 65410 | 597411 | 0 | 0 | 2 | 0 |
| 20 | 6840 | 889 | 127 | 263 | 50080 | 455906 | 2 | 0 | 0 | 0 |
| 22 | 7831 | 1096 | 315 | 229 | 62661 | 653266 | 0 | 0 | 1 | 0 |

## Obtención de los centroides

```
CentroidesP = Hipoteca.groupby('clusterP').mean()
CentroidesP
```

| clusterP | ingresos | gastos_comunes | pago_coche | gastos_otros | ahorros | vivienda | estado_civil | hijos | trabajo |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 6358.959184 | 1117.306122 | 190.755102 | 465.653061 | 50687.081633 | 497262.265306 | 0.448980 | 0.061224 | 2.122449 |
| 1 | 3472.482143 | 905.607143 | 224.732143 | 536.589286 | 23957.642857 | 272010.535714 | 1.625000 | 2.250000 | 6.660714 |
| 2 | 6389.685185 | 998.851852 | 190.203704 | 524.148148 | 54899.722222 | 430860.092593 | 1.462963 | 2.222222 | 6.296296 |
| 3 | 3502.930233 | 857.209302 | 245.790698 | 533.627907 | 24129.139535 | 291900.953488 | 0.348837 | 0.000000 | 2.093023 |

## Conclusiones de los clústeres

### Clúster número: 0

a) Este clúster está conformado por 49 casos de una evaluación hipotecaria.
b) Con un ingreso promedio mensual de 6358.96 USD
c) Con gastos comunes promedios de 1117.31 USD
d) Tienen un pago promedio mensual de su coche de 190.76 USD
e) Otros gastos en promedio de 465.65 USD

Estos gastos en promedio representan el 27.89 % del ingreso total.

f) Por otro lado, este grupo de usuarios tienen un ahorro promedio de 50687.08 USD
g) y un valor promedio de vivienda (a comprar o hipotecar) de 497262.27 USD
h) Además, su estado civil en promedio es: 0.45 [ 0-soltero, 1-casado, 2-divorciado ]

Se puede observar que en su mayoría son solteros.

```
0    34
1     8
2     7
Name: estado_civil, dtype: int64
```

i) Tienen en promedio, 0.06 hijos menores. Por lo que la mayoría no tiene hijos menores.

```
0    47
2     1
1     1
Name: hijos, dtype: int64
```

j) y tienen en promedio un tipo de trabajo 2.12 [ 0-sin trabajo, 1-autónomo, 2-asalariado, 3-empresario, 4-autonomos, 5-asalariados, 6-autonomo y asalariado, 7-empresario y autónomo, 8-empresarios o empresario y autónomo ]

Se puede observar que en su mayoría son empresarios, autónomos y desempleados.

```
3    12
4    10
0    10
1     9
2     7
5     1
Name: trabajo, dtype: int64
```

Basándome en mi propio criterio, yo consideraría sí darle el crédito a este grupo de usuarios, ya que sus gastos son bajos (gastan poco) y además en su mayoría son trabajadores empresarios, por lo que tienen buen sustento económico. Por otra parte, su ahorro promedio es muy considerable.

## Clúster número 1

a) Este clúster está conformado por 56 casos de una evaluación hipotecaria.
b) Con un ingreso promedio mensual de 3472.48 USD
c) Con gastos comunes promedios de 905.61 USD
d) Tienen un pago promedio mensual de su coche de 224.73 USD
e) Otros gastos en promedio de 536.59 USD

Estos gastos en promedio representan el 48.0 % del ingreso total.

f) Por otro lado, este grupo de usuarios tienen un ahorro promedio de 23957.64 USD
g) y un valor promedio de vivienda (a comprar o hipotecar) de 272010.54 USD
h) Además, su estado civil en promedio es: 1.62 [ 0-soltero, 1-casado, 2-divorciado ]

Se observa que la mayoría son divorciados.

```
2    35
1    21
Name: estado_civil, dtype: int64
```

i) Tienen en promedio, 2.25 hijos menores.

```
3    17
1    17
2    15
4     7
Name: hijos, dtype: int64
```

a) y tienen en promedio un tipo de trabajo 6.66 [ 0-sin trabajo, 1-autónomo, 2-asalariado, 3-empresario, 4-autonomos, 5-asalariados, 6-autonomo y asalariado, 7-empresario y autónomo, 8-empresarios o empresario y autónomo ]

Se observa que la mayoría son empresario y autónomo y empresarios o empresario y autónomo

```
8    16
7    16
6    13
5    11
Name: trabajo, dtype: int64
```

Basándome en mi propio criterio, yo consideraría no darle el crédito a este grupo de usuarios, ya que sus gastos son muy elevados (casi el 50% de sus ingresos), en su mayoría son empresarios o trabajadores autónomos, por lo que tienen buen sustento económico. Por otra parte, su ahorro promedio no es muy bueno ya que como se ve, tienen muchas responsabilidades (la mayoría tiene un hijo o más) por lo que es muy probable que se atrasen con los pagos debido a su exceso de responsabilidades y su ahorro tan pobre.

## Clúster número: 2

a) Este clúster está conformado por 54 casos de una evaluación hipotecaria.
b) Con un ingreso promedio mensual de 6389.69 USD
c) Con gastos comunes promedios de 998.85 USD
d) Tienen un pago promedio mensual de su coche de 190.2 USD
e) Otros gastos en promedio de 524.15 USD

Estos gastos en promedio representan el 26.81 % del ingreso total.

f) Por otro lado, este grupo de usuarios tienen un ahorro promedio de 54899.72 USD
g) y un valor promedio de vivienda (a comprar o hipotecar) de 430860.09 USD
h) Además, su estado civil en promedio es: 1.46 [ 0-soltero, 1-casado, 2-divorciado ]

Se observa que la mayoría son casados, pero también hay un grupo de divorciados.

```
1    29
2    25
Name: estado_civil, dtype: int64
```

i) Tienen en promedio, 2.22 hijos menores

```
1    17
2    15
3    11
4    10
0     1
Name: hijos, dtype: int64
```

b) y tienen en promedio un tipo de trabajo 6.3 [ 0-sin trabajo, 1-autónomo, 2-asalariado, 3-empresario, 4-autonomos, 5-asalariados, 6-autonomo y asalariado, 7-empresario y autonomo, 8-empresarios o empresario y autónomo ]

Se observa que la mayoría es asalariada.

```
5    19
8    13
7    11
6    10
4     1
Name: trabajo, dtype: int64
```

Basándome en mi propio criterio, yo consideraría sí darle el crédito a este grupo de usuarios, ya que sus gastos no son muy elevados (son poco más del 25% de sus ingresos), la mayoría de este grupo tiene un empleo sólido, por lo que tienen buen sustento económico. Por otra parte, su ahorro promedio es muy bueno, entonces es poco probable que se atrasen con los pagos.

**Clúster número: 3**

a) Este clúster está conformado por 43 casos de una evaluación hipotecaria.
b) Con un ingreso promedio mensual de 3502.93 USD
c) Con gastos comunes promedios de 857.21 USD
d) Tienen un pago promedio mensual de su coche de 245.79 USD
e) Otros gastos en promedio de 533.63 USD

Estos gastos en promedio representan el 46.72 % del ingreso total.

c) Por otro lado, este grupo de usuarios tienen un ahorro promedio de 24129.14 USD
d) y un valor promedio de vivienda (a comprar o hipotecar) de 291900.95 USD
e) Además, su estado civil en promedio es:  0.35 [ 0-soltero, 1-casado, 2-divorciado ]

Se observa que en su mayoría son solteros.

```
0    34
2     6
1     3
Name: estado_civil, dtype: int64
```

f) Tienen en promedio, 0.0 hijos menores, es decir, ninguno tiene hijos.
g) y tienen en promedio un tipo de trabajo 2.09 [ 0-sin trabajo, 1-autónomo, 2-asalariado, 3-empresario, 4-autonomos, 5-asalariados, 6-autonomo y asalariado, 7-empresario y autónomo, 8-empresarios o empresario y autónomo ]

Se observa que en su mayoría son asalariados.

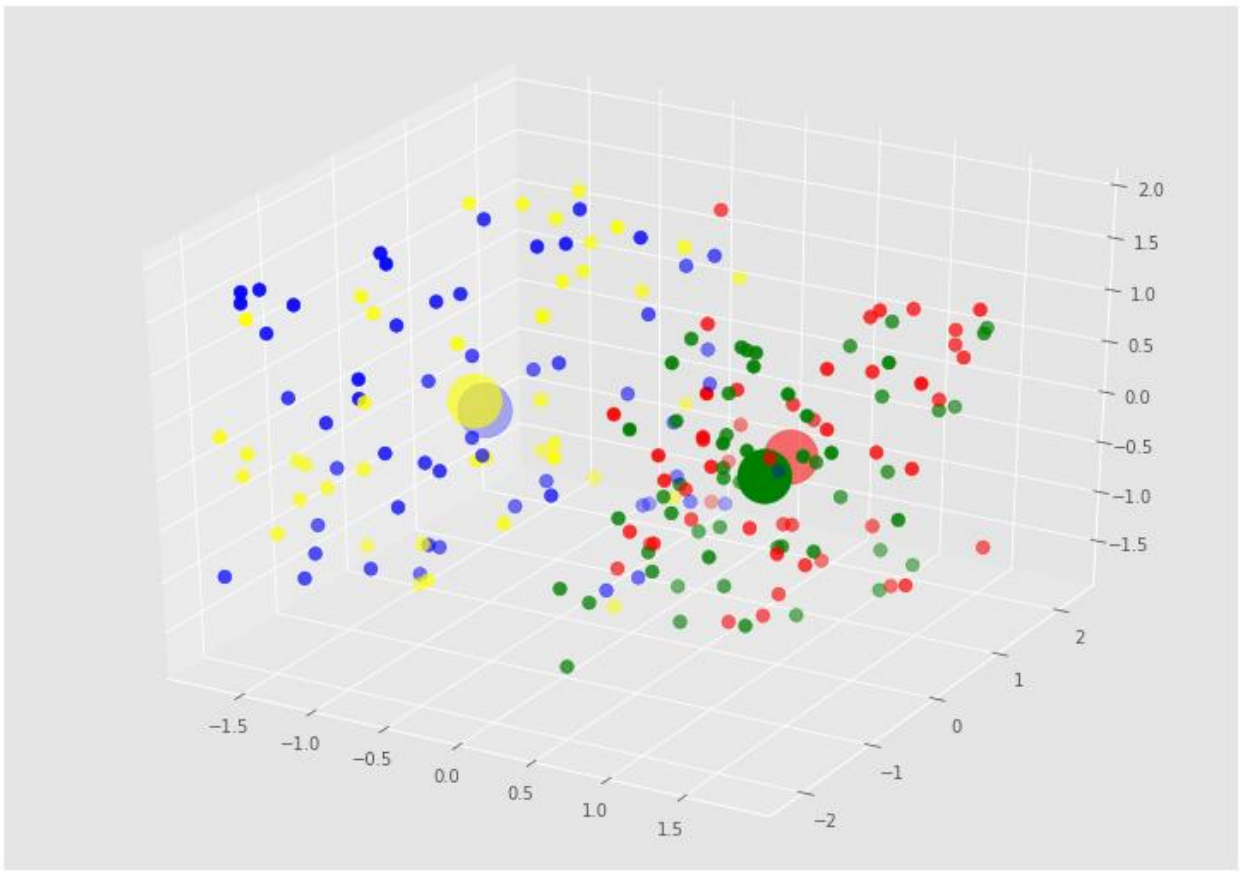
```
2    11
3     9
1     9
4     8
0     6
Name: trabajo, dtype: int64
```

Basándome en mi propio criterio, yo consideraría sí darle el crédito a este grupo de usuarios, ya que a pesar de que sus gastos son muy elevados (casi el 50% de sus ingresos), su empleo lo compensa ya que en su mayoría son asalariados, empresarios o trabajadores autónomos, por lo que tienen buen sustento económico. Por otra parte, su ahorro promedio no es muy bueno pero como no tienen muchas responsabilidades (la mayoría son solteros y sin hijos), entonces es poco probable que se atrasen con los pagos.

```
# Gráfica de los elementos y los centros de los clústeres
from mpl_toolkits.mplot3d import Axes3D
plt.rcParams['figure.figsize'] = (10, 7)
plt.style.use('ggplot')
colores=['red', 'blue', 'green', 'yellow']
```

```
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()
```




## Conclusiones

En esta práctica pude aprender cómo generar clústeres a partir de una fuente de datos, con el objetivo de analizar si cada uno de estos grupos (en total fueron 4 grupos obtenidos) es apto para acceder a la adquisición de una casa a través de un crédito hipotecario con tasa fija a 30 años.

Se implementó el algoritmo de K-means para este análisis, pero, como se está trabajando con clustering (que entra dentro de la categoría de aprendizaje no supervisado), los cuales son algoritmos basado en distancias, se tuvo 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, se obtuvieron 4 clústeres o 4 grupos de usuarios, los cuales pude analizar y dar mis conclusiones sobre si es óptimo otorgarles el crédito o no, basándome en mis propios criterios.

En esta práctica 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 clústering.

## Link de Google Colab

OCG-Práctica5-Clústering.ipynb - Colaboratory (google.com)