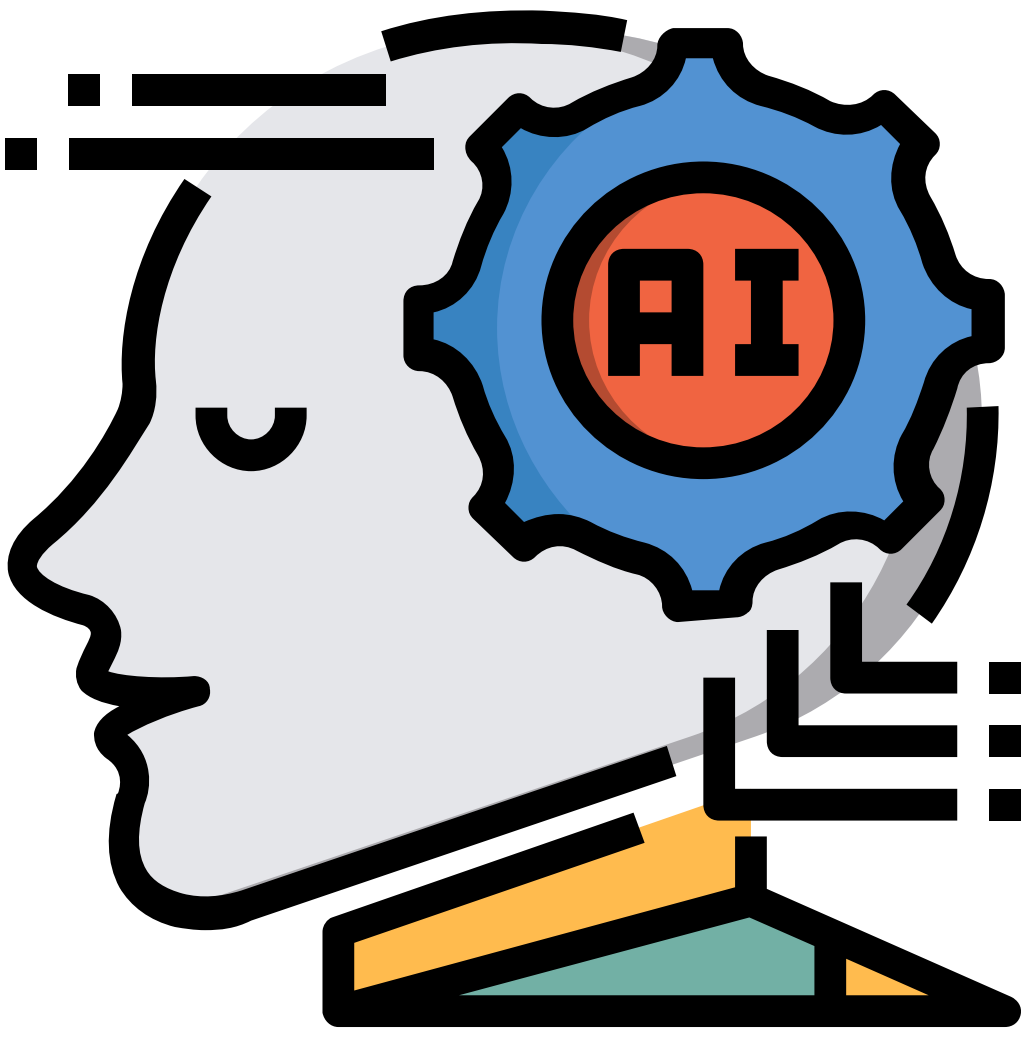
Universidad Nacional Autónoma de México
Facultad de Ingeniería

Inteligencia Artificial

PRÁCTICA 4. CLUSTERING JERÁRQUICO

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



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

Tabla de 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 autónomo, 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())
```

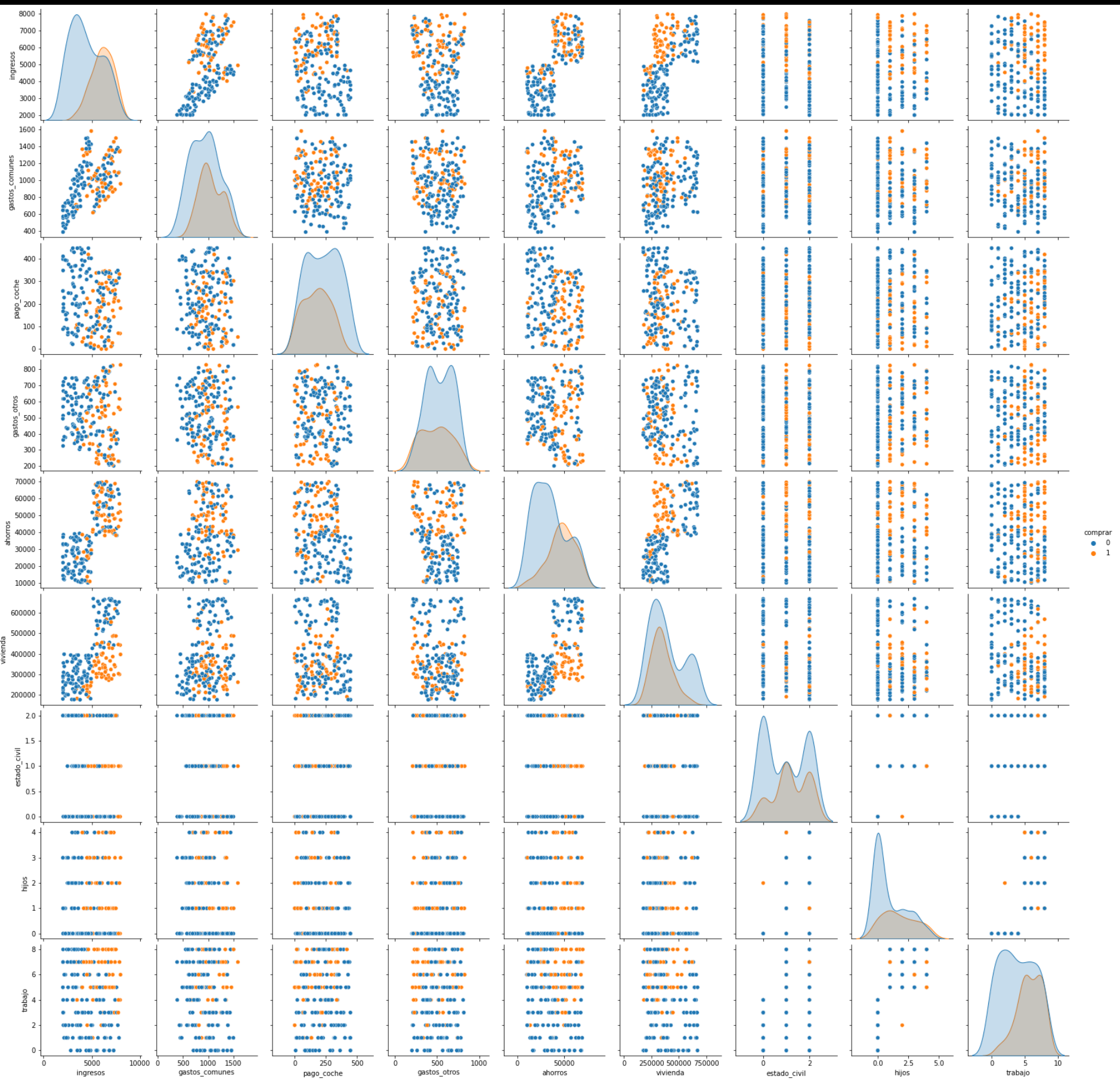
```
comprar
0    135
1     67
dtype: int64
```

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.

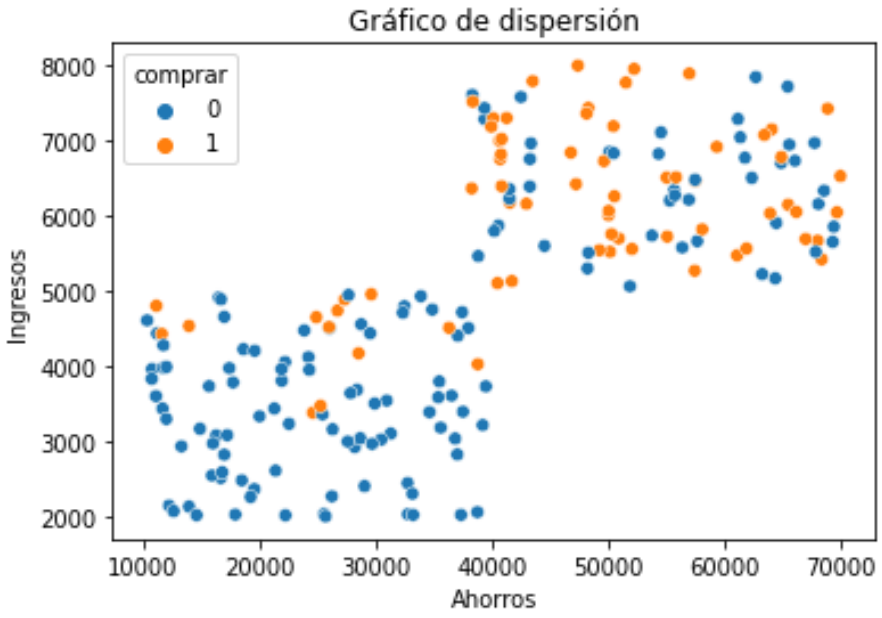
Selección de características

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
```

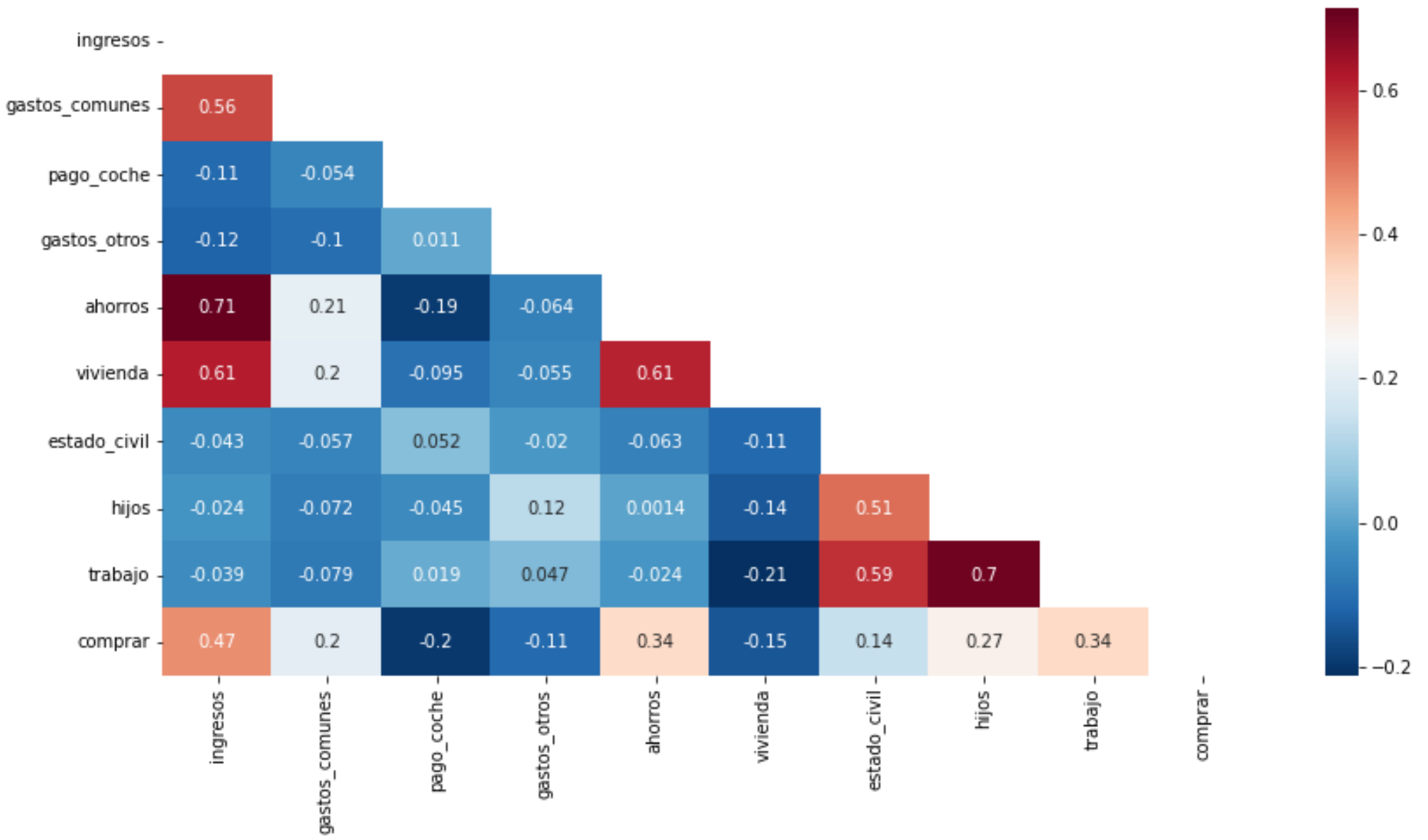
	ingresos	gastos_comunes	pago_coche	gastos_otros	ahorros	vivienda	estado_civil	hijos	trabajo	comprar
ingresos	1.000000	0.560211	-0.109780	-0.124105	0.712889	0.614721	-0.042556	-0.024483	-0.038852	0.467123
gastos_comunes	0.560211	1.000000	-0.054400	-0.099881	0.209414	0.204781	-0.057152	-0.072321	-0.079095	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: Jerárquico Ascendente

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)
```

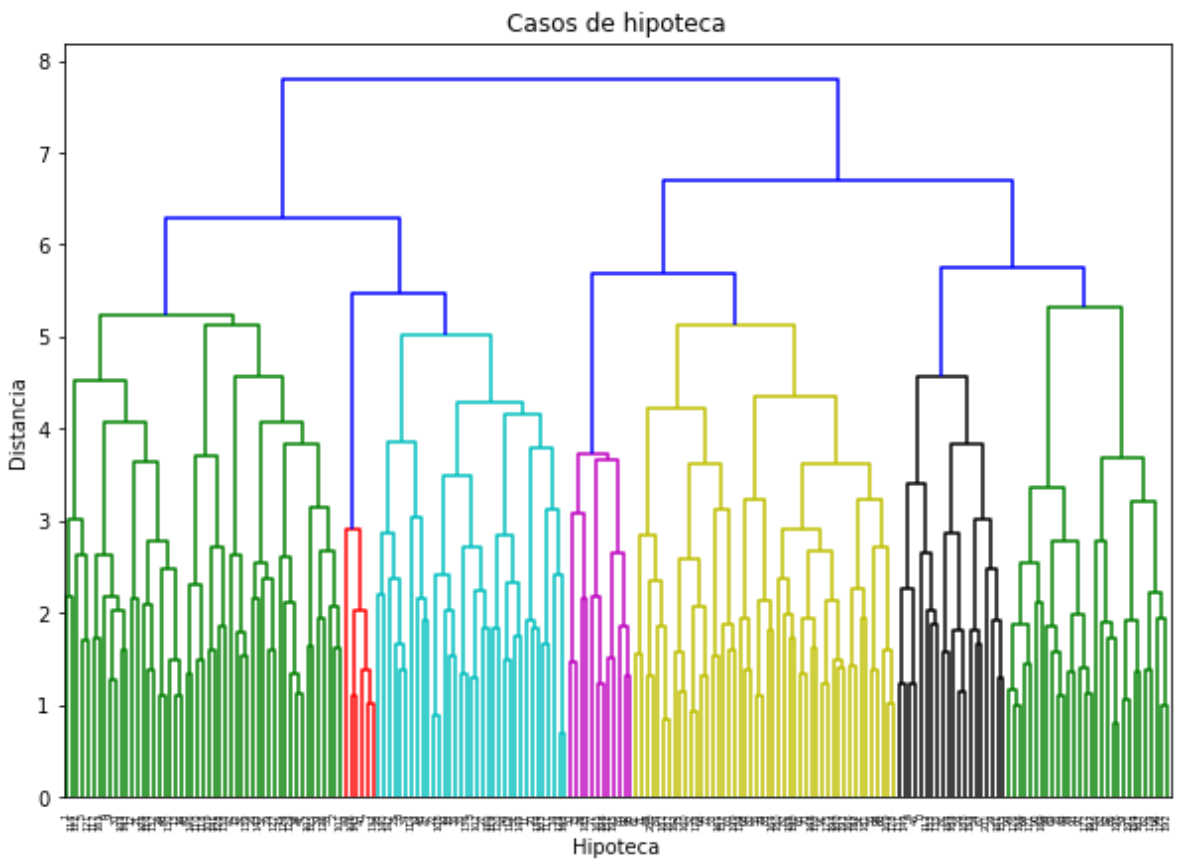
MinMaxScaler se usa cuando no hay datos fuera de rango. Estos datos atípicos se pueden remover de nuestro análisis.

Cuando hay datos atípicos se usa la función StandardScaler.

	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 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("Casos de hipoteca")
plt.xlabel('Hipoteca')
plt.ylabel('Distancia')
Arbol = shc.dendrogram(shc.linkage(MEstandarizada, method='complete', metric='euclidean')) #Utilizamos la matriz estandarizada
#plt.axhline(y=6, color='orange', linestyle='--') # Hace un corte en las ramas
#Probar con otras mediciones de distancia (chebyshev, cityblock, etc.)
```



Se puede observar que hay 7 ramas en nuestro árbol, cada una con su respectivo color.

```
#Se crean las etiquetas de los elementos en los clústeres
MJerarquico = AgglomerativeClustering(n_clusters=7, linkage='complete', affinity='euclidean')
MJerarquico.fit_predict(MEstandarizada)
MJerarquico.labels_
```

```
array([4, 1, 1, 2, 4, 1, 1, 6, 2, 1, 2, 2, 1, 1, 2, 1, 1, 1, 2, 2, 2, 1,
       2, 1, 4, 2, 1, 2, 2, 1, 1, 2, 1, 2, 2, 1, 2, 1, 1, 1, 6, 1, 1, 1,
       2, 2, 4, 2, 1, 6, 5, 3, 3, 3, 4, 3, 3, 0, 4, 0, 3, 3, 0, 3, 0, 3,
       3, 4, 3, 0, 3, 3, 3, 5, 0, 3, 0, 5, 5, 3, 3, 4, 0, 3, 3, 5, 0, 3,
       3, 0, 0, 3, 5, 0, 0, 5, 0, 0, 3, 0, 3, 1, 2, 1, 1, 2, 6, 1, 2, 1,
       1, 2, 4, 2, 2, 1, 1, 1, 1, 2, 2, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 4,
       6, 4, 2, 4, 2, 1, 1, 1, 2, 1, 2, 1, 2, 6, 1, 1, 2, 4, 2, 4, 5, 4,
       4, 4, 0, 3, 3, 0, 3, 3, 3, 1, 3, 5, 3, 0, 3, 3, 3, 0, 0, 3, 0, 3,
       0, 0, 3, 3, 3, 3, 3, 3, 4, 5, 0, 3, 4, 0, 3, 0, 0, 3, 3, 5, 0, 0,
       5, 3, 3, 4])
```

```
Hipoteca = Hipoteca.drop(columns=['comprar'])
Hipoteca['clusterH'] = MJerarquico.labels_
Hipoteca
```

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

202 rows × 10 columns

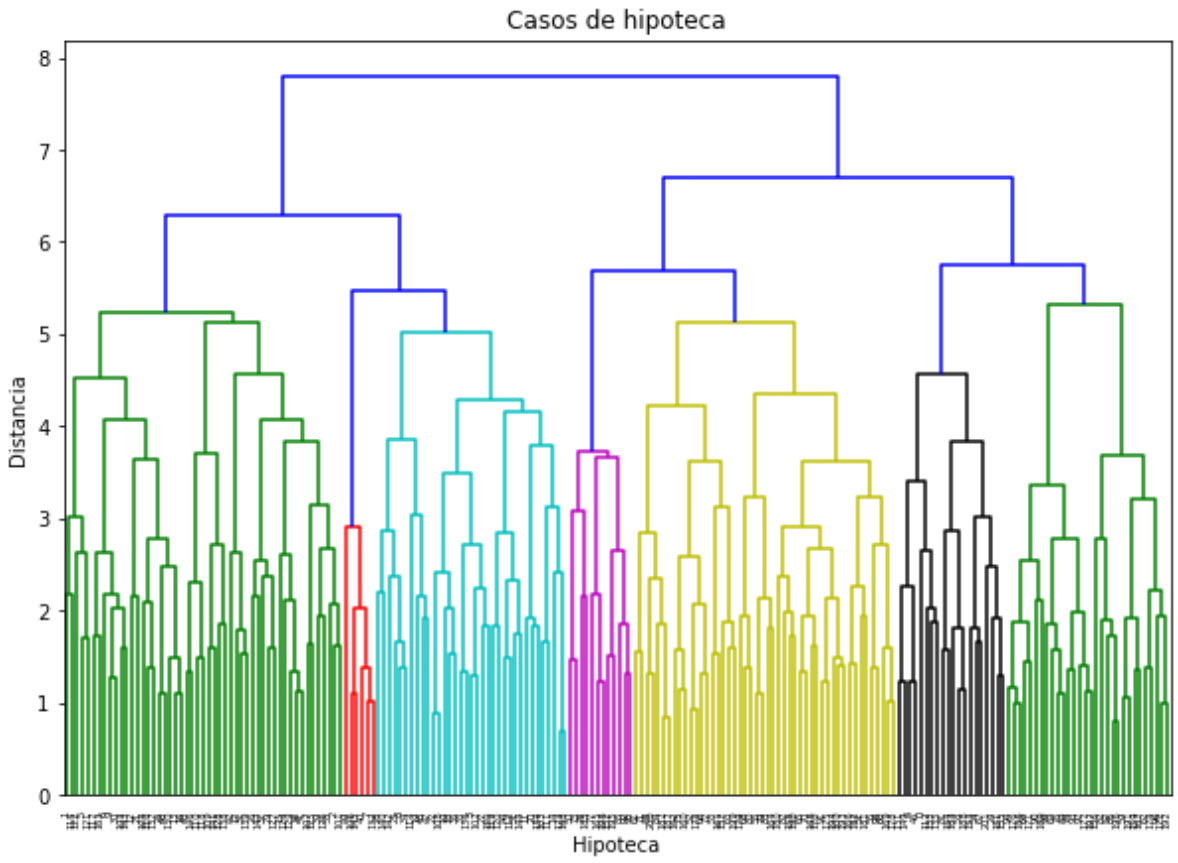
```
#Cantidad de elementos en los clusters
Hipoteca.groupby(['clusterH'])['clusterH'].count()
```

```
clusterH
0    30
1    51
2    35
3    48
4    20
5    12
6     6
Name: clusterH, dtype: int64
```

```
Hipoteca[Hipoteca.clusterH == 6] # Filtramos los clusters etiquetados como 6, que son 6 elementos
```

	ingresos	gastos_comunes	pago_coche	gastos_otros	ahorros	vivienda	estado_civil	hijos	trabajo	clusterH
7	6470	1035	39	782	57439	606291	0	0	1	6
40	6822	1296	81	786	50433	669054	0	0	0	6
49	6959	1392	333	818	67714	571076	0	0	3	6
106	6205	1179	240	729	56904	661009	0	0	2	6
132	6325	1139	102	754	68527	588004	0	0	0	6
145	5646	1016	215	747	69276	655399	0	0	1	6

Los datos etiquetados con el 6, son los que están en color rojo:




```
CentroidesH = Hipoteca.groupby('clusterH').mean()
CentroidesH
```

Los centroides es el promedio de los valores contenidos en los clústeres.

clusterH	ingresos	gastos_comunes	pago_coche	gastos_otros	ahorros	vivienda	estado_civil	hijos	trabajo
0	3421.133333	846.466667	309.933333	527.233333	24289.633333	295590.700000	0.233333	0.000000	2.000000
1	6394.019608	1021.627451	192.274510	533.039216	54382.529412	421178.764706	1.490196	2.254902	6.313725
2	6599.542857	1087.428571	204.771429	362.600000	51863.028571	515494.257143	0.685714	0.228571	2.885714
3	3189.687500	785.020833	243.208333	548.270833	23616.854167	277066.687500	1.645833	1.979167	6.208333
4	4843.750000	1009.200000	122.200000	572.850000	36340.650000	337164.850000	0.050000	0.100000	1.900000
5	4466.416667	1315.083333	114.416667	502.750000	23276.166667	269429.916667	1.666667	2.416667	6.750000
6	6404.500000	1176.166667	168.333333	769.333333	61715.500000	625138.833333	0.000000	0.000000	1.166667

Conclusiones de los clústeres obtenidos

Clúster 0:

a) Este clúster está conformado por **30 casos de una evaluación hipotecaria.**
b) Con un **ingreso promedio mensual de 3421.13 USD**
c) Con **gastos comunes promedios de 846.47 USD**
d) Tienen un pago **promedio mensual de su coche de 309.93 USD**
e) **Otros gastos en promedio de 527.23 USD**.

Estos gastos en promedio representan el 49.21 % del ingreso promedio mensual, es decir, casi la mitad del salario mensual.

f) Por otro lado, este grupo de usuarios tienen un **ahorro promedio de 24289.63 USD**
g) y un **valor promedio de vivienda (a comprar o hipotecar) de 295590.7 USD**
h) Además, su **estado civil en promedio es: 0.23** [0-soltero, 1-casado, 2-divorciado]

En este punto se puede observar que en su mayoría son solteros.

```
0    25
1     3
2     2
Name: estado_civil, dtype: int64
```

i) Tienen en promedio, 0.0 hijos menores, es decir, **no tienen hijos menores.**
j) y tienen en promedio un tipo de trabajo 2.0 [0-sin trabajo, 1-autónomo, 2-asalariado, 3-empresario, 4-autónomos, 5-asalariados, 6-autónomo y asalariado, 7-empresario y autónomo, 8-empresarios o empresario y autónomo]

Se puede observar que la **mayoría son Autónomos:**

```
1    9
2    8
4    5
3    5
0    3
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 (gastan la mitad de lo que ganan) y además en su mayoría son trabajadores autónomos, por lo que se corre el riesgo de que no paguen a tiempo porque sus ingresos son algo volátiles, es decir, no tienen un ingreso fijo. Por otra parte, su ahorro promedio no es muy considerable.

Clúster 1:

a) Este clúster está conformado por **51 casos de una evaluación hipotecaria.**
b) Con un **ingreso promedio mensual de 6394.02 USD**
c) Con **gastos comunes promedios de 1021.63 USD**
d) Tienen un **pago promedio mensual de su coche de 192.27 USD**
e) **Otros gastos en promedio de 533.04 USD**

Estos gastos en promedio representan el 27.32 % del ingreso promedio mensual.

f) Por otro lado, este grupo de usuarios tienen un **ahorro promedio de 54382.53 USD**
g) y un **valor promedio de vivienda (a comprar o hipotecar) de 421178.76 USD**
h) Además, su **estado civil en promedio es: 1.49** [0-soltero, 1-casado, 2-divorciado]

Se puede observar que en su mayoría son casados.

```
1    26
2    25
Name: estado_civil, dtype: int64
```

i) Tienen en **promedio, 2.25 hijos menores**

La mayoría tiene un hijo:

```
1    16
2    13
3    11
4    10
0     1
Name: hijos, dtype: int64
```

a) y tienen en **promedio un tipo de trabajo 6.31** [0-sin trabajo, 1-autónomo, 2-asalariado, 3-empresario, 4-autónomos, 5-asalariados, 6-autónomo y asalariado, 7-empresario y autónomo, 8-empresarios o empresario y autónomo]

Se puede observar que hay más asalariados:

```
5    18
8    13
7    10
6     9
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 son muy bajos en comparación a sus ingresos, la mayoría son casados, tienen hijos y son asalariados y empresarios, por lo que tienen un ingreso constante y el riesgo de que incumplan con los pagos es mínimo. En sumo a lo anterior, su ahorro promedio es una cantidad considerable.

Clúster 2:

a) Este clúster está conformado por **35 casos de una evaluación hipotecaria.**
b) Con un **ingreso promedio mensual de 6599.54 USD**
c) Con **gastos comunes promedios de 1087.43 USD**
d) Tienen un **pago promedio mensual de su coche de 204.77 USD**
e) **Otros gastos en promedio de 362.6 USD**

Estos gastos en promedio representan el 25.07 % del ingreso promedio mensual.

f) Por otro lado, este grupo de usuarios tienen un **ahorro promedio de 51863.03 USD**
g) y un **valor promedio de vivienda (a comprar o hipotecar) de 515494.26 USD**
h) Además, su **estado civil en promedio es: 0.69** [0-soltero, 1-casado, 2-divorciado]

En su mayoría son solteros:

```
0    18
1    10
2     7
Name: estado_civil, dtype: int64
```

i) Tienen en **promedio, 0.23 hijos menores**

La mayoría no tiene hijos:

```
0    30
2     3
1     2
Name: hijos, dtype: int64
```

b) y tienen en **promedio un tipo de trabajo 2.89** [0-sin trabajo, 1-autónomo, 2-asalariado, 3-empresario, 4-autónomos, 5-asalariados, 6-autónomo y asalariado, 7-empresario y autónomo, 8-empresarios o empresario y autónomo]

Es decir, en su mayoría son empresarios:

```
3    9
4    8
0    5
2    5
1    3
5    2
6    2
7    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 en comparación a sus ingresos, la mayoría son solteros, no tienen hijos y son empresarios, por lo que tienen un ingreso constante y considerable, por lo que el riesgo de que incumplan con los pagos es mínimo. En sumo a lo anterior, su ahorro promedio es considerable.

Clúster 3:

a) Este clúster está conformado por **48 casos de una evaluación hipotecaria.**
b) Con un **ingreso promedio mensual de 3189.69 USD**
c) Con **gastos comunes promedios de 785.02 USD**
d) Tienen un **pago promedio mensual de su coche de 243.21 USD**
e) **Otros gastos en promedio de 548.27 USD**

Estos gastos en promedio representan el 49.42 % del ingreso promedio mensual.

f) Por otro lado, este grupo de usuarios tienen un **ahorro promedio de 23616.85 USD**
g) y un **valor promedio de vivienda (a comprar o hipotecar) de 277066.69 USD**
h) Además, su **estado civil en promedio es: 1.65** [0-soltero, 1-casado, 2-divorciado]

En su mayoría son divorciados:

```
2    31
1    17
Name: estado_civil, dtype: int64
```

i) Tienen **en promedio, 1.98 hijos menores**

La mayoría tiene 3 hijos:

```
3    14
1    13
2    12
0     5
4     4
Name: hijos, dtype: int64
```

c) y tienen en **promedio un tipo de trabajo 6.21** [0-sin trabajo, 1-autónomo, 2-asalariado, 3-empresario, 4-autónomos, 5-asalariados, 6-autónomo y asalariado, 7-empresario y autónomo, 8-empresarios o empresario y autónomo]

En su mayoría son empresarios y autónomos:

```
7    13
8    12
5     9
6     9
2     3
3     2
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 bastante elevados en comparación a sus ingresos (gastan casi la mitad de sus ingresos), la mayoría son divorciados, con varios hijos y son empresarios. Considero que aunque puedan tener un ingreso considerable, sus obligaciones y gastos son elevados (mantenimiento de hijos + divorcio) por lo que el riesgo de que incumplan con los pagos es algo elevado. En sumo a lo anterior, su ahorro promedio es bastante bajo para sus responsabilidades que tienen.

Clúster 4:

a) Este clúster está conformado por **20 casos de una evaluación hipotecaria**.
b) Con un **ingreso promedio mensual de 4843.75 USD**
c) Con **gastos comunes promedios de 1009.2 USD**
d) Tienen un **pago promedio mensual de su coche de 122.2 USD**
e) **Otros gastos en promedio de 572.85 USD**

Estos gastos en promedio representan el 35.18 % del ingreso promedio mensual.

d) Por otro lado, este grupo de usuarios tienen un **ahorro promedio de 36340.65 USD**
e) y un **valor promedio de vivienda (a comprar o hipotecar) de 337164.85 USD**
f) Además, su **estado civil en promedio es: 0.05** [0-soltero, 1-casado, 2-divorciado]

En su mayoría son solteros:

```
0    19
1     1
Name: estado_civil, dtype: int64
```

g) Tienen en **promedio, 0.1 hijos menores**

La mayoría no tiene hijos:

```
0    19
2     1
Name: hijos, dtype: int64
```

h) y tienen en **promedio un tipo de trabajo 1.9** [0-sin trabajo, 1-autónomo, 2-asalariado, 3-empresario, 4-autónomos, 5-asalariados, 6-autónomo y asalariado, 7-empresario y autónomo, 8-empresarios o empresario y autónomo]

La mayoría de ellos no tiene trabajo.

```
0    6
4    5
3    4
1    4
2    1
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 algo elevados en comparación con sus ingresos. Aunque la mayoría sea soltera y sin hijos, se debe observar que o no tienen trabajo, o son autónomos (la mayoría), por lo que tienen un ingreso volátil o no lo tienen, por lo que el riesgo de que incumplan con los pagos es elevado.

Clúster 5:

a) Este clúster está conformado por **12 casos de una evaluación hipotecaria**.
b) Con un **ingreso promedio mensual de 4466.42 USD**
c) Con **gastos comunes promedios de 1315.08 USD**
d) Tienen un **pago promedio mensual de su coche de 114.42 USD**
e) **Otros gastos en promedio de 502.75 USD**

Estos gastos en promedio representan el 43.26 % del ingreso promedio mensual.

f) Por otro lado, este grupo de usuarios tienen un **ahorro promedio de 23276.17 USD**
g) y un **valor promedio de vivienda (a comprar o hipotecar) de 269429.92 USD**
h) Además, su **estado civil en promedio es: 1.67** [0-soltero, 1-casado, 2-divorciado]

La mayoría está divorciada:

```
2    8
1    4
Name: estado_civil, dtype: int64
```

i) Tienen en **promedio, 2.42 hijos menores**

La mayoría tiene un hijo:

```
1    4
4    3
3    3
2    2
Name: hijos, dtype: int64
```

i) y tienen en promedio un tipo de trabajo 6.75 [0-sin trabajo, 1-autónomo, 2-asalariado, 3-empresario, 4-autónomos, 5-asalariados, 6-autonomo y asalariado, 7-empresario y autónomo, 8-empresarios o empresario y autónomo]

La mayoría son empresarios o empresarios y autónomos:

```
8    4
7    3
6    3
5    2
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 bastante elevados en comparación a sus ingresos (gastan casi la mitad de sus ingresos), la mayoría son divorciados, con uno o varios hijos y son empresarios. Considero que aunque puedan tener un ingreso considerable, sus obligaciones y gastos son elevados (mantenimiento de hijos + divorcio) por lo que el riesgo de

que incumplan con los pagos es algo elevado. En sumo a lo anterior, su ahorro promedio es bastante bajo para sus responsabilidades que tienen.

Clúster 6:

a) Este clúster está conformado por **6 casos de una evaluación hipotecaria**.
b) Con un **ingreso promedio mensual de 6404.5 USD**
c) Con **gastos comunes promedios de 1176.17 USD**
d) Tienen un **pago promedio mensual de su coche de 168.33 USD**
e) **Otros gastos en promedio de 769.33 USD**

Estos gastos en promedio representan el 33.01 % del ingreso promedio mensual.

f) Por otro lado, este grupo de usuarios tienen un **ahorro promedio de 61715.5 USD**
g) y un **valor promedio de vivienda (a comprar o hipotecar) de 625138.83 USD**
h) Además, su **estado civil en promedio es: 0.0** [0-soltero, 1-casado, 2-divorciado]

Es decir, **todos son solteros.**

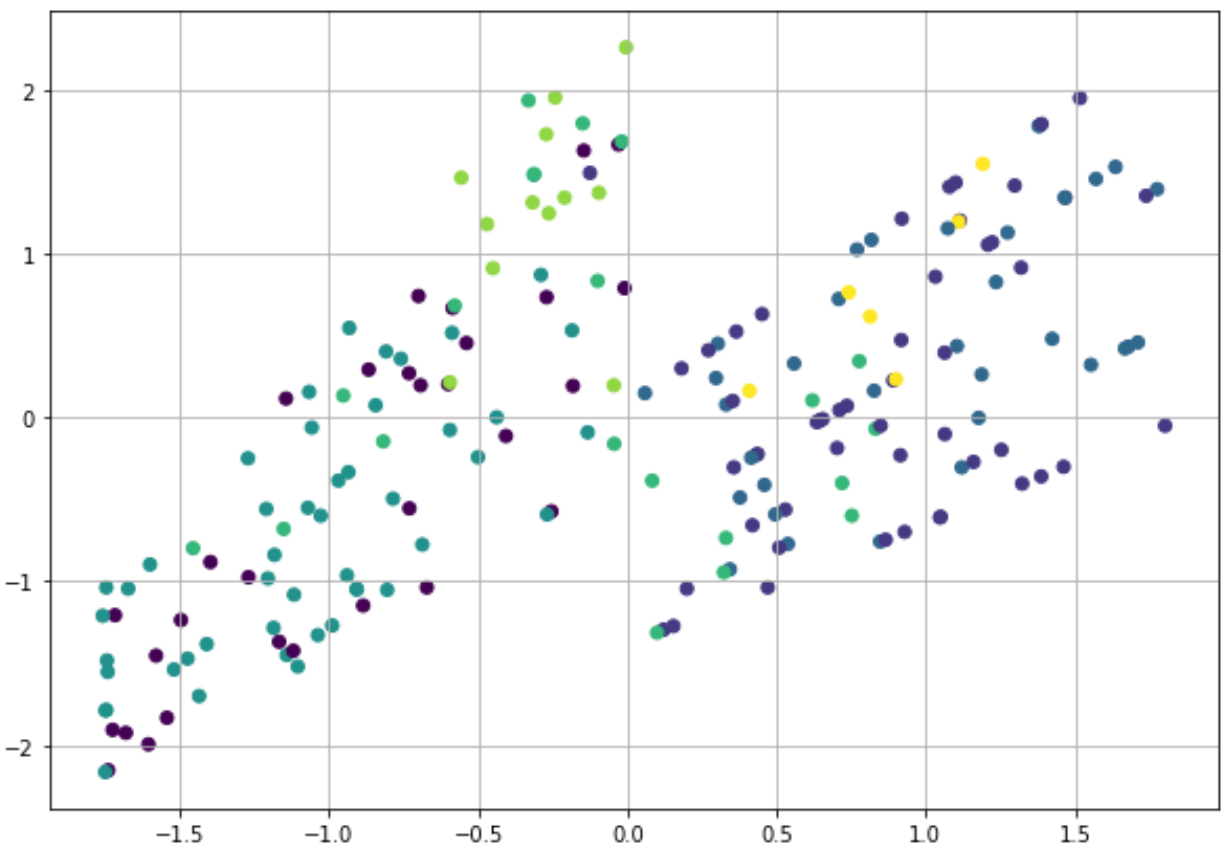
i) Tienen en promedio, 0.0 hijos menores, es decir, **no tienen hijos.**
j) y tienen en promedio un tipo de trabajo 1.17 [0-sin trabajo, 1-autónomo, 2-asalariado, 3-empresario, 4-autónomos, 5-asalariados, 6-autonomo y asalariado, 7-empresario y autónomo, 8-empresarios o empresario y autónomo]

La mayoría no tienen trabajo o son autónomos:

```
1    2
0    2
3    1
2    1
Name: trabajo, dtype: int64
```

Basándome en mi propio criterio, yo consideraría no darle el crédito a este grupo de usuarios, ya que siento que los datos son engañosos, puesto que se menciona que 2 de estos usuarios no tienen empleo, pero ganan lo mismo que el usuario que es empresario. Sin tomar el tipo de trabajo, si daría el crédito, pero considerando el tipo de trabajo y que son desempleados, no. En este caso tal vez se necesite más la opinión de un experto (aunque claro, en todos se necesitaría de un Actuario, por ejemplo)

```
plt.figure(figsize=(10, 7))
plt.scatter(MEstandarizada[:,0], MEstandarizada[:,1], c=MJerarquico.labels_)
plt.grid()
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 7 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 ascendente jerárquico para este análisis, pero, como se está trabajando con clústering (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 7 clústeres o 7 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-Práctica4-Clústering.ipynb - Colaboratory (google.com)