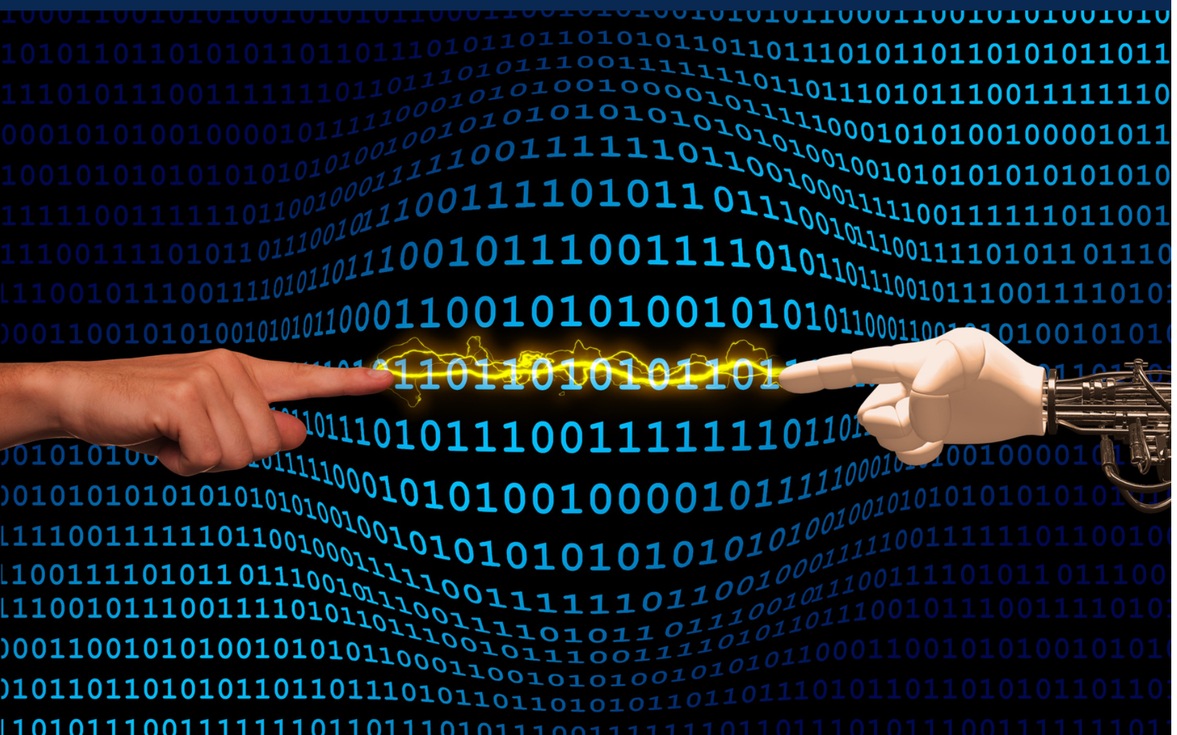
EL IMPACTO DE LA INFORMACIÓN RELACIONAL EN EL APRENDIZAJE AUTOMÁTICO: ARTISTAS DE SPOTIFY

JAVIER FERNÁNDEZ CASTILLO
MANUEL OTERO BARBASÁN

INTELIGENCIA ARTIFICIAL
JUNIO DE 2023

INDICE DE CONTENIDOS



1. INTRODUCCIÓN

Objetivo del proyecto -> estudiar el impacto de la información relacional en en aprendizaje automático.

Dominio del problema: colaboraciones entre artistas de Spotify.

Atributos relacionales empleado: coeficiente de clustering, grado de centralidad y comunidades.

Atributo objetivo: índice de popularidad.

Modelos empleados: k-NN, árboles de decisión y redes neuornales

2. CONJUNTO DE DATOS Y OBJETIVO A PREDECIR

2.1. DATOS BASE

	spotify_id	name	followers	popularity	genres	chart_hits
0	4lDiJcOJ2GLCK6p9q5BgfK	Kontra K	1999676.0	72	['christlicher rap', 'german hip hop']	['at (44)', 'de (111)', 'lu (22)', 'ch (31)', ...
1	652XIvIBNGg3C0KIGEJWit	Maxim	34596.0	36	[]	['de (1)']
2	3dXC1YPbnQPsfHPVkm1ipj	Christopher Martin	249233.0	52	['dancehall', 'lovers rock', 'modern reggae', ...	['at (1)', 'de (1)']
3	74terC9ol9zMo8rfzhSOiG	Jakob Hellman	21193.0	39	['classic swedish pop', 'norrbotten indie', 's...	['se (6)']
4	0FQMb3mVrAKlyU4H5mQOJh	Madh	26677.0	19	[]	['it (2)']

2.2. OBJETIVO DE PREDICCIÓN

Atributo objetivo: índice de popularidad.

2.3. ATRIBUTOS SELECCIONADOS PARA EL ENTRENAMIENTO

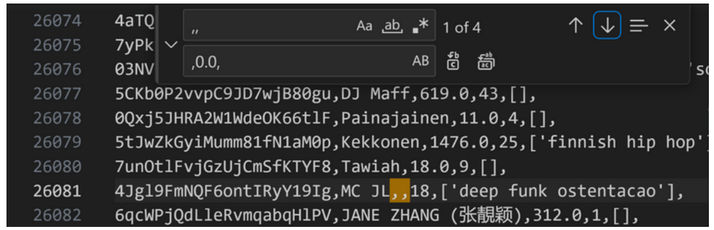
	spotify_id	name	followers	popularity	genres	chart_hits
0	4lDiJcOJ2GLCK6p9q5BgfK	Kontra K	1999676.0	72	['christlicher rap', 'german hip hop']	['at (44)', 'de (111)', 'lu (22)', 'ch (31)', ...
1	652XIvIBNGg3C0KIGEJWit	Maxim	34596.0	36	[]	['de (1)']
2	3dXC1YPbnQPsfHPVkm1ipj	Christopher Martin	249233.0	52	['dancehall', 'lovers rock', 'modern reggae', ...	['at (1)', 'de (1)']
3	74terC9ol9zMo8rfzhSOiG	Jakob Hellman	21193.0	39	['classic swedish pop', 'norrbotten indie', 's...	['se (6)']
4	0FQMb3mVrAKlyU4H5mQOJh	Madh	26677.0	19	[]	['it (2)']

2.4. PROCESAMIENTO DE DATOS

Creación del contador de países

	chart_hits	country_hits_count
1	['de (1)']	1
2	['at (1)', 'de (1)']	2
3	['se (6)']	1

Subsanación de datos nulos



Elminación de nodos duplicados

Normalización de datos

	followers	popularity	centrality	clustering	community	country_hits_count
0	1.957457	72.0	3.593487	6.795635	0.000000	7.042254
1	0.033866	36.0	0.449186	10.714286	0.000000	1.408451
2	0.243971	52.0	2.189781	3.778677	0.008501	2.816901

Nodos antes de la subsanación = 156422

Nodos después de la subsanación = 156319

Diferencia = 109

2.5. IMPORTANCIA DE LOS ATRIBUTOS A PRIORI

Followers

Centralidad

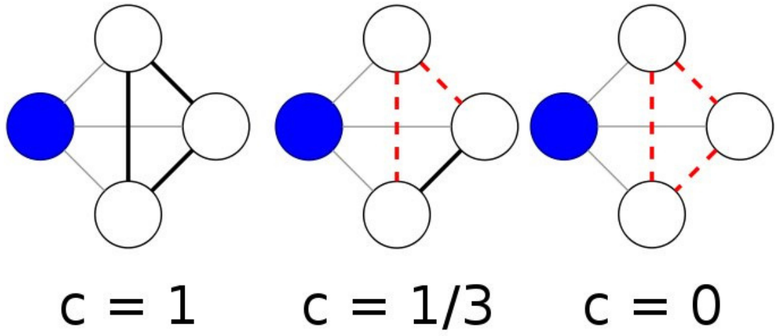
Lista de éxitos

Clustering y comunidades

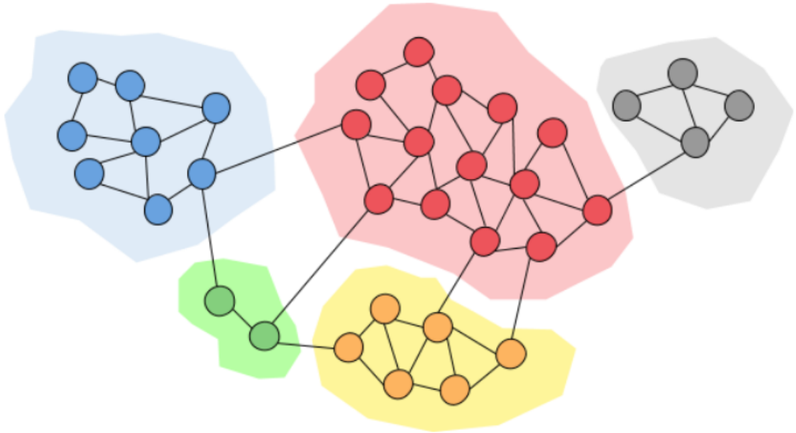
2.6. DIVISIÓN DE DATOS

5 pruebas. Máxima diferencia 0.4%

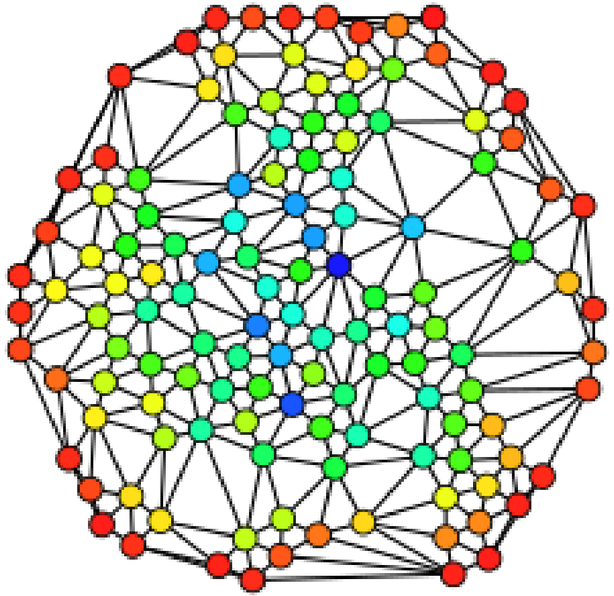
3. MÉTRICAS RELACIONALES



Coeficiente de clustering



Comunidades (Label Propagation)



Centralidad

4. VARIANTES DE DATOS DE ENTRENAMIENTO

	followers	centrality	clustering	community	country_hits_count
0	1.957457	3.593487	6.795635	0.000000	7.042254
1	0.033866	0.449186	10.714286	0.000000	1.408451
2	0.243971	2.189781	3.778677	0.008501	2.816901

Todos los atributos (drop([]))

	followers	country_hits_count
0	1.957457	7.042254
1	0.033866	1.408451
2	0.243971	2.816901

Atributos no relacionales
(drop([’centrality’,’clustering’,’community’]))

	centrality	clustering	community	country_hits_count
0	3.593487	6.795635	0.000000	7.042254
1	0.449186	10.714286	0.000000	1.408451
2	2.189781	3.778677	0.008501	2.816901

Todos los atributos menos followers
(drop([’followers’]))

	centrality	clustering	community
0	3.593487	6.795635	0.000000
1	0.449186	10.714286	0.000000
2	2.189781	3.778677	0.008501

Atributos relacionales
(drop([’followers’,’country-hits-count’]))

5. MÓDELOS EMPLEADOS

5.1. K-NN

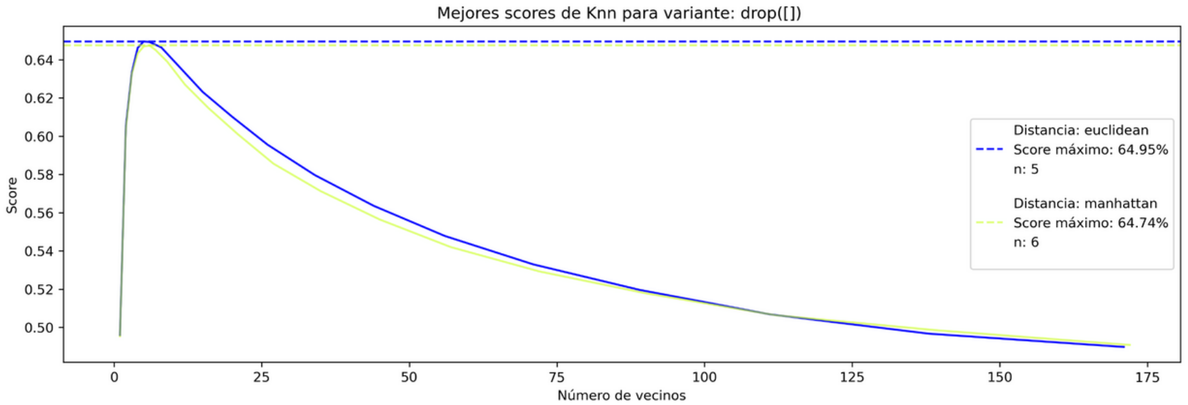
ALGORITMO DE CÁLCULO DE HIPERPARÁMETROS

Algorithm 1 allCombinationsKnnForDataSet

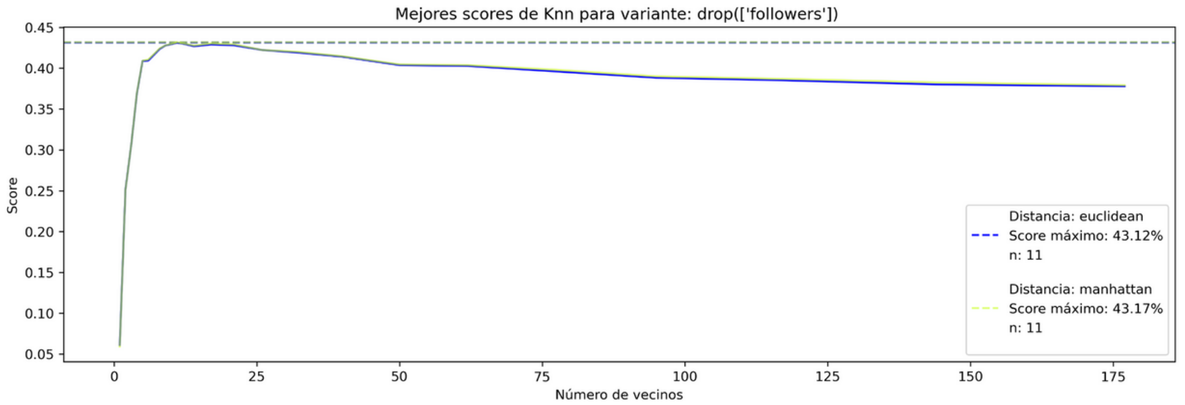
Require: *metrics* es una lista de distancias de knn, *jumpHit*: salto en acierto, *jumpFactorFail* es el factor por el que se multiplica el salto en caso de fallo, *variantDropping* es una variante de datos, $nMax$ es el número máximo de vecinos

$scores \leftarrow \{\}$
for $m = 0$ hasta tamaño($metrics$)-1 **do**
  $scores[m] \leftarrow []$; $best \leftarrow 0$; $jump \leftarrow 1$; $n \leftarrow 1$
  while $n < nMax$ **do**
    (pred, score) $\leftarrow$ knnRegressor(metrics[m], n, variantDropping)
    scores[m] $\leftarrow$ ((n, score, pred))
    if $score \geq best$ **then**
      $jump \leftarrow jumpHit$; $best \leftarrow score$
    else
      $jump \leftarrow \lceil jump \times jumpFactorFail \rceil$
    end if
    actualiza valores
    $n \leftarrow n + jump$
  end while
end for
return *scores*

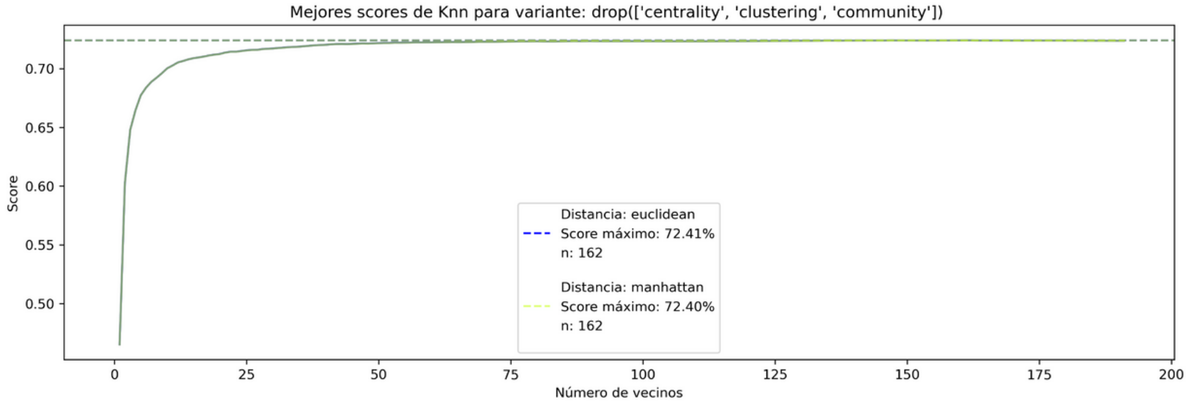
5.1. K-NN



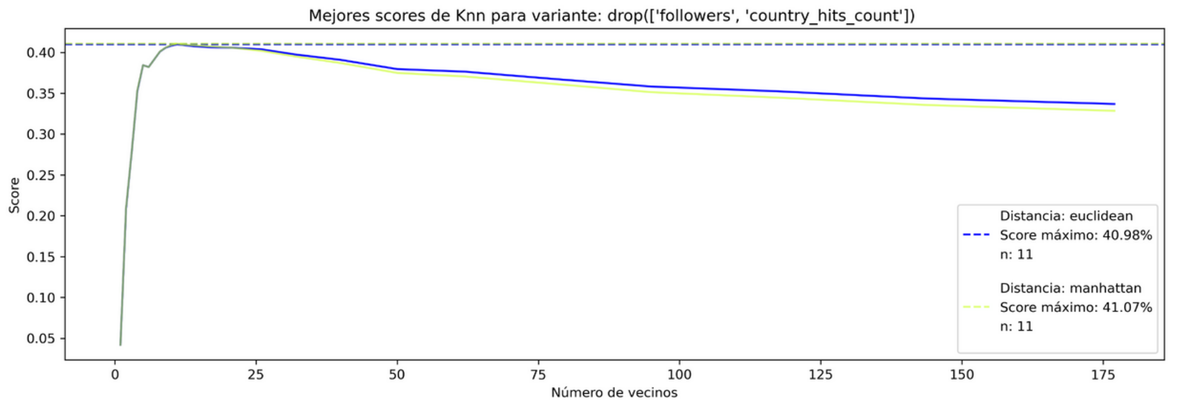
Todos los atributos (drop([]))



Todos los atributos menos followers
(drop([’followers’]))

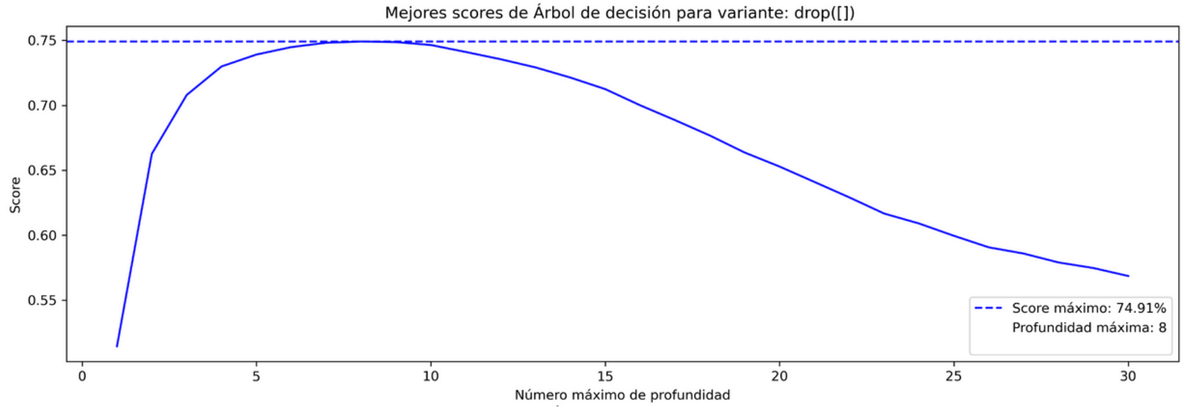


Atributos no relacionales
(drop([’centrality’,’clustering’,’community’]))

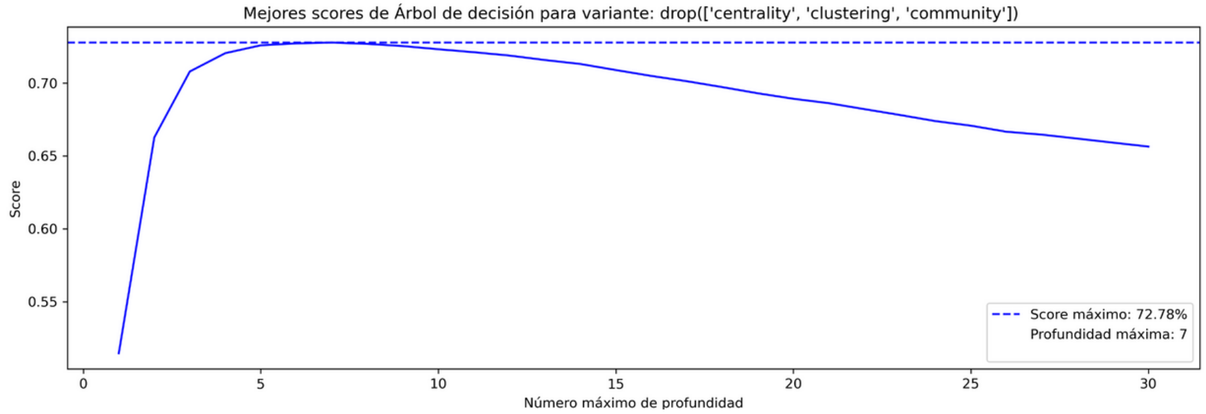


Atributos relacionales
(drop([’followers’,’country-hits-count’]))

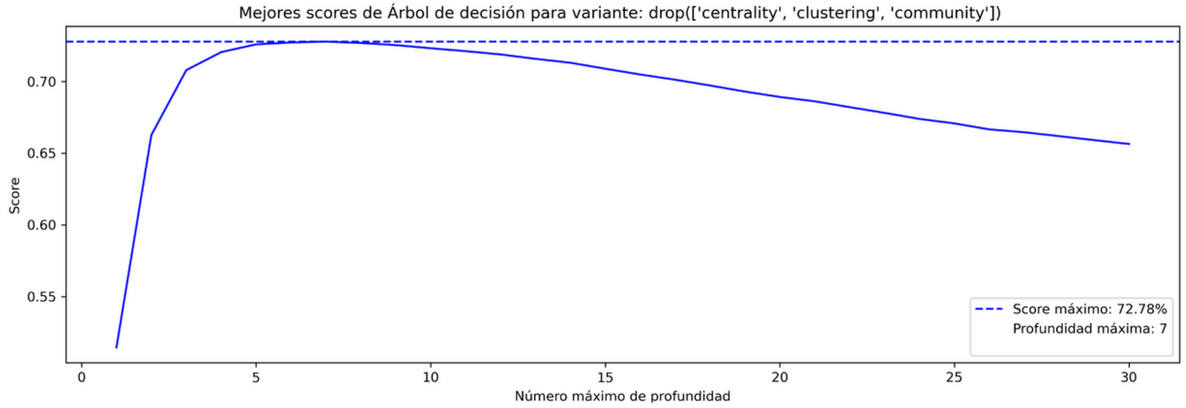
5.2. ÁRBOLES DE DECISIÓN



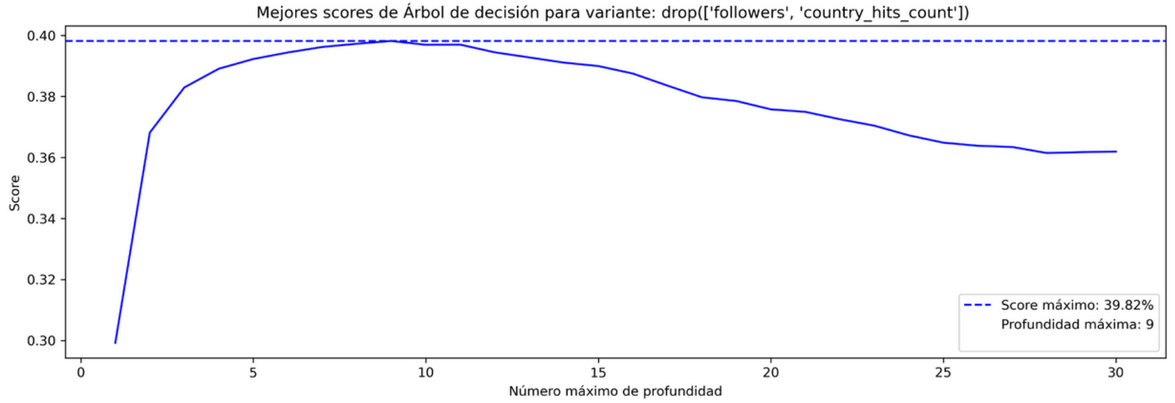
Todos los atributos (drop([]))



Todos los atributos menos followers
(drop([’followers’]))



Atributos no relacionales
(drop([’centrality’,’clustering’,’community’]))



Atributos relacionales
(drop([’followers’,’country-hits-count’]))

5.3. REDES NEURONALES

Números de capas ocultas: 7

Número de neuronas por capa oculta: 2, 50, 2, 30, 7, 3, 2

Función de activación de las capas ocultas: tanh, identidad, tanh, identidad, identidad, identidad, identidad

Función de activación de la capa de entrada y de salida: identidad, identidad

Factor de aprendizaje: 0,00008

Número de lotes: 100

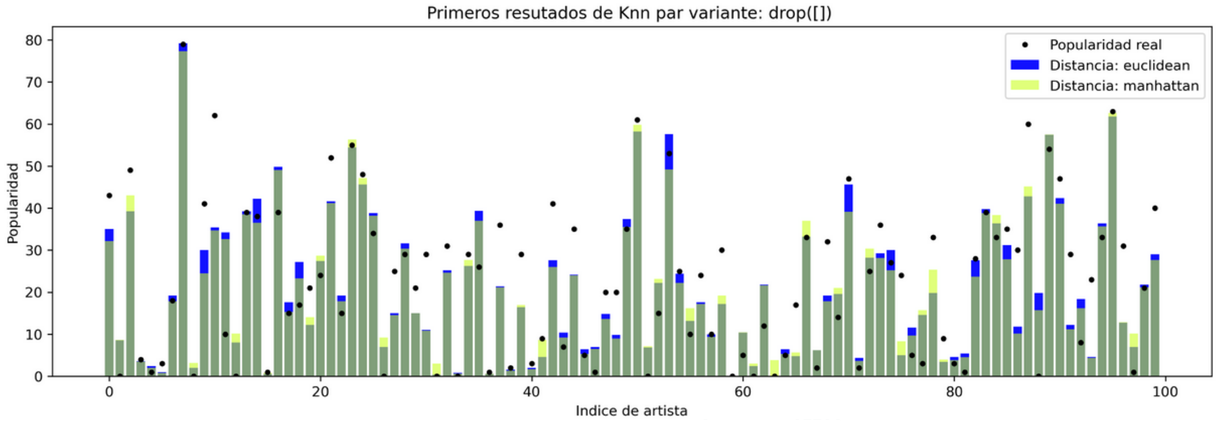
Número de épocas de entrenamiento: 20

Optimizador: SGD

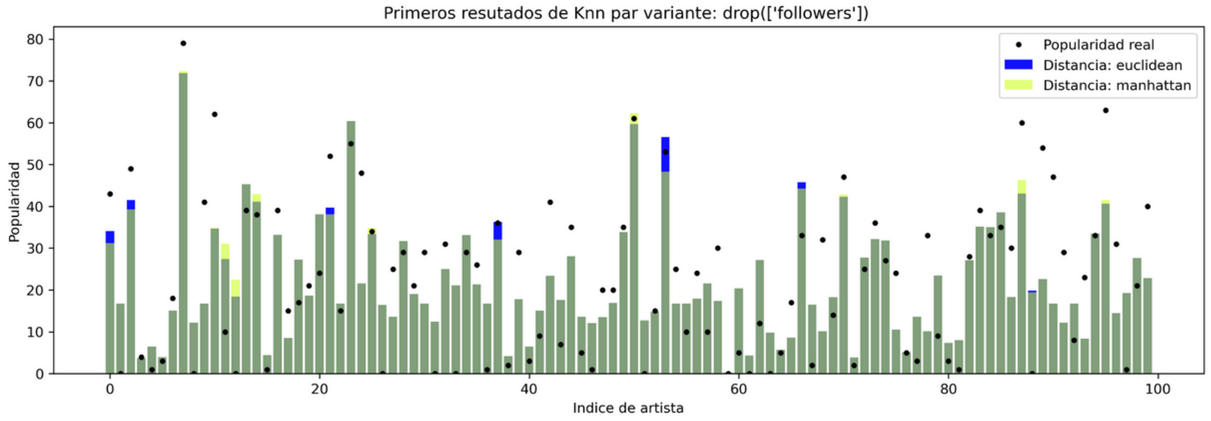
Pérdidas: mean-square-error

6. ÁNALISIS DE RESULTADOS

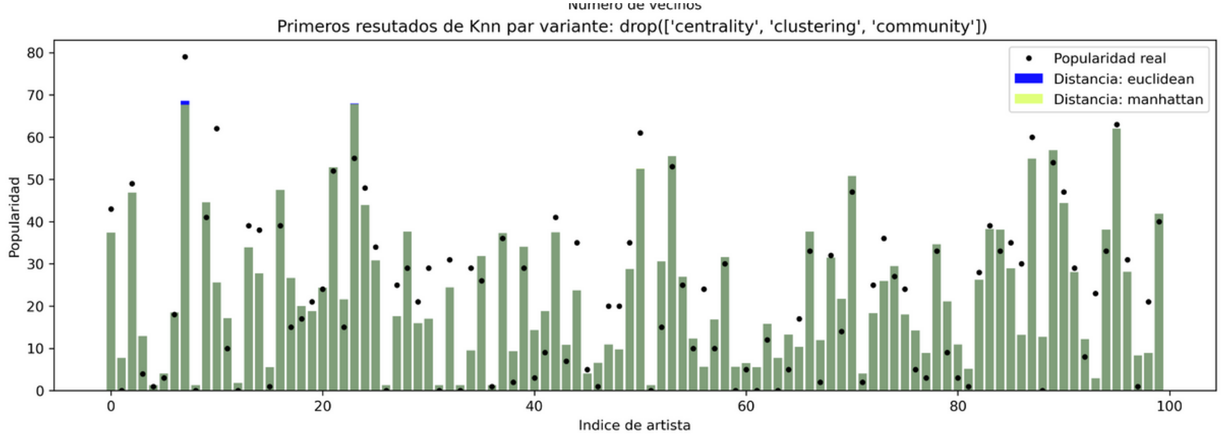
6.1. K-NN



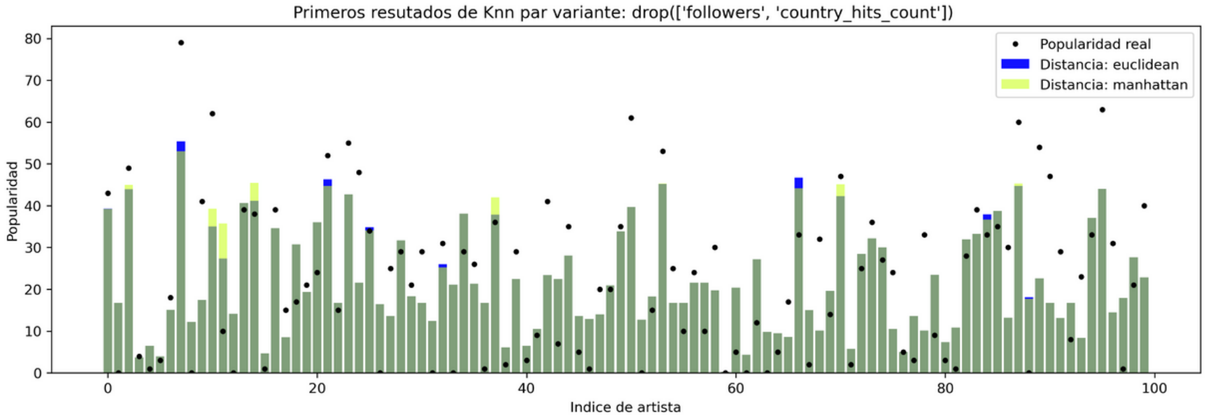
Todos los atributos (drop([]))



Todos los atributos menos followers (drop([’followers’]))



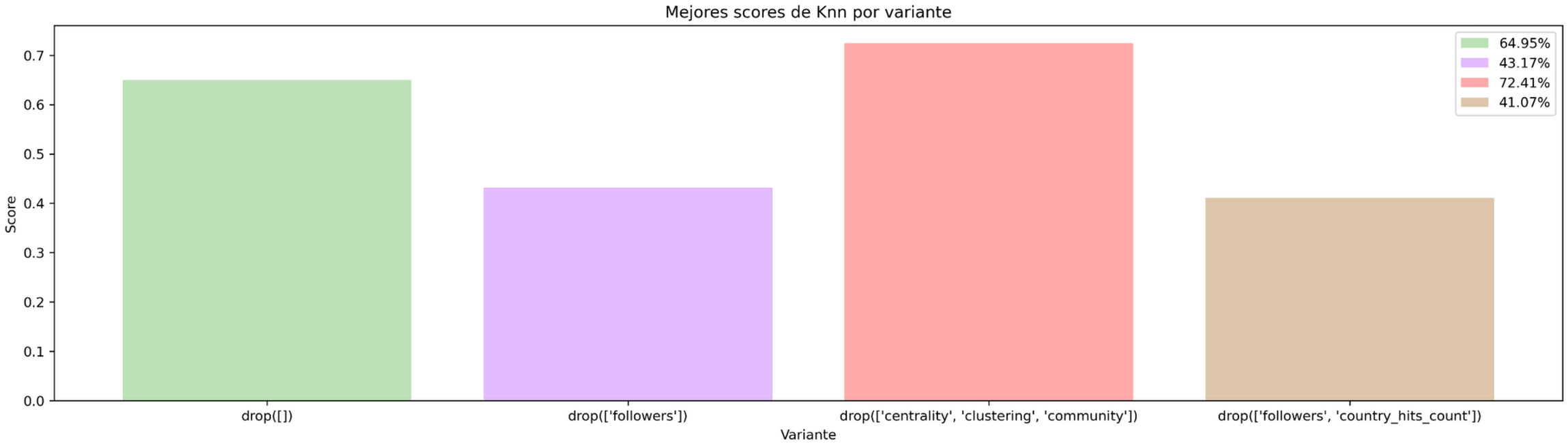
Atributos no relacionales (drop([’centrality’,’clustering’,’community’]))



Atributos relacionales (drop([’followers’,’country-hits-count’]))

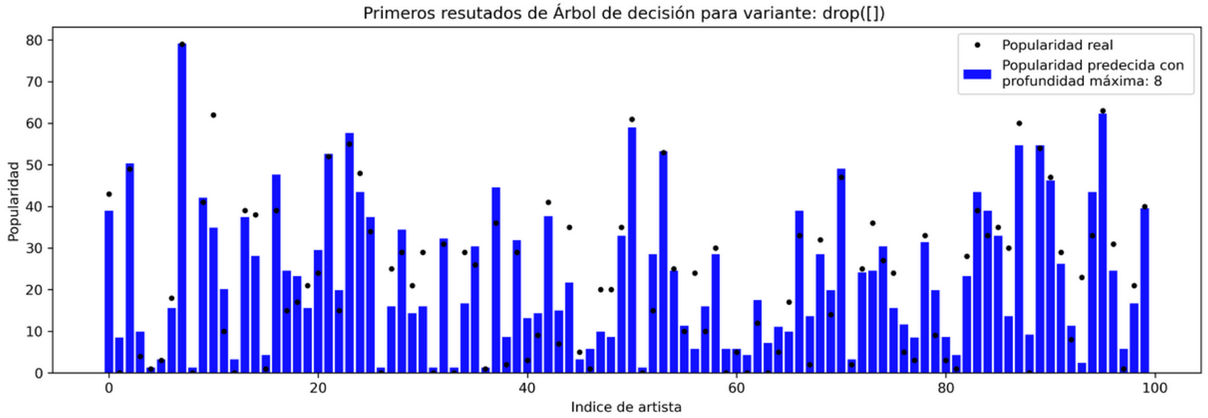
6. ÁNALISIS DE RESULTADOS

6.1. K-NN

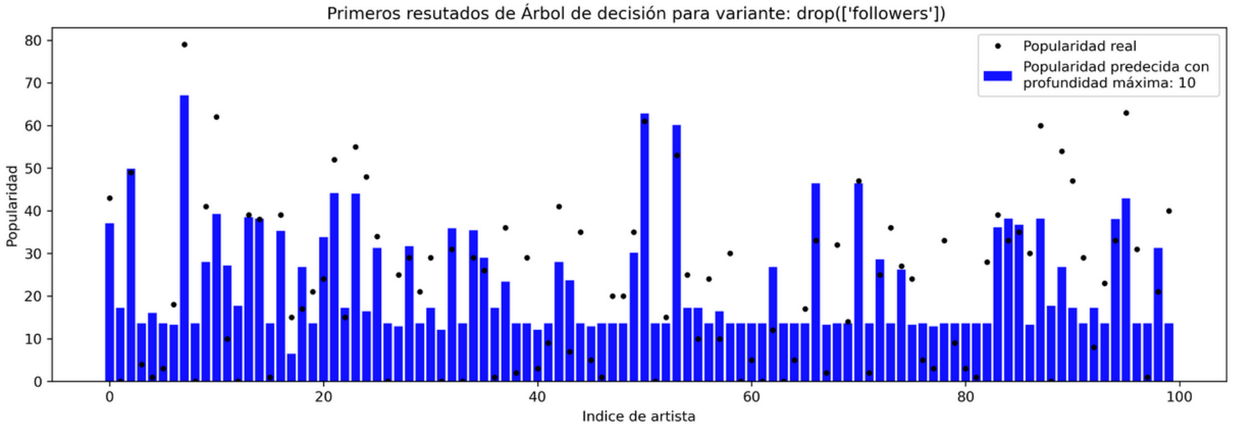


6. ÁNALISIS DE RESULTADOS

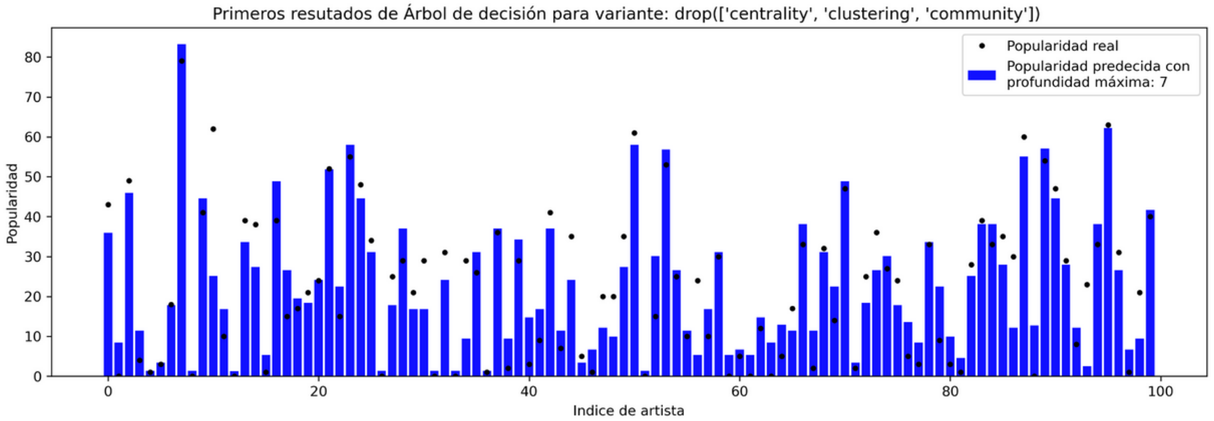
6.2. ÁRBOLES DE DECISIONES



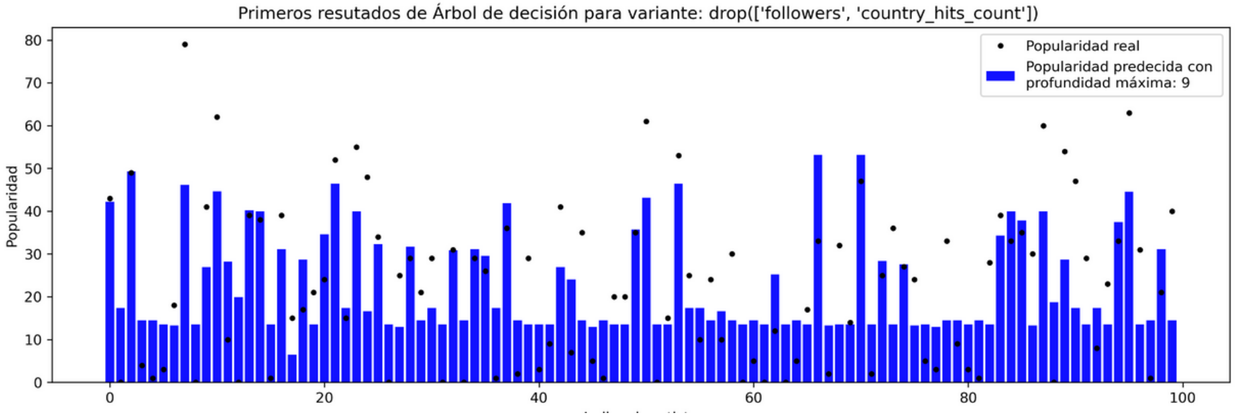
Todos los atributos (drop([]))



Todos los atributos menos followers (drop([’followers’]))



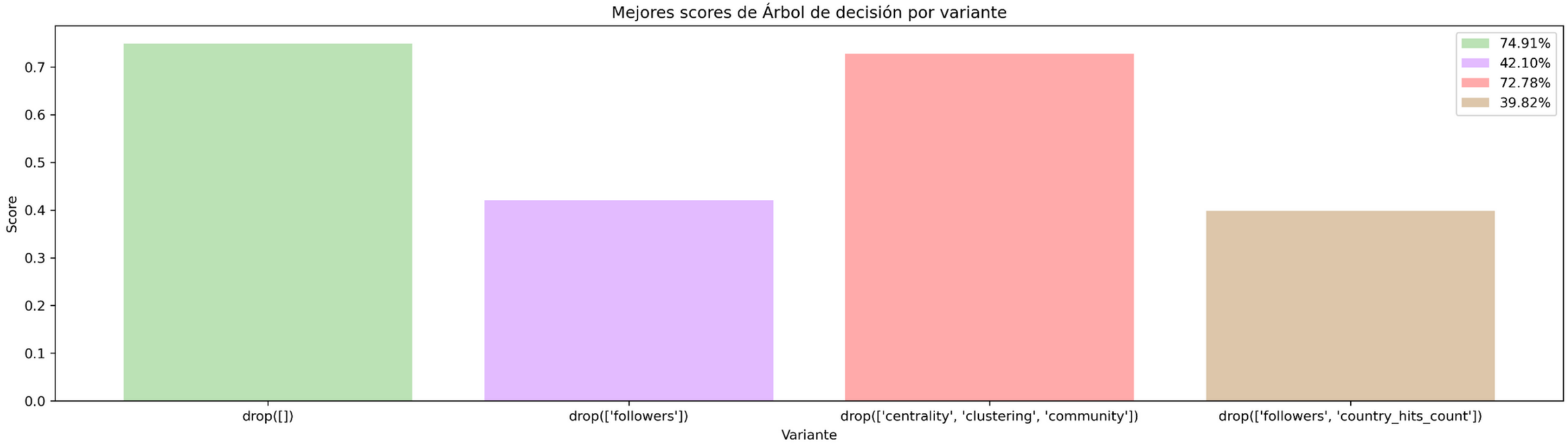
Atributos no relacionales (drop([’centrality’,’clustering’,’community’]))



Atributos relacionales (drop([’followers’,’country-hits-count’]))

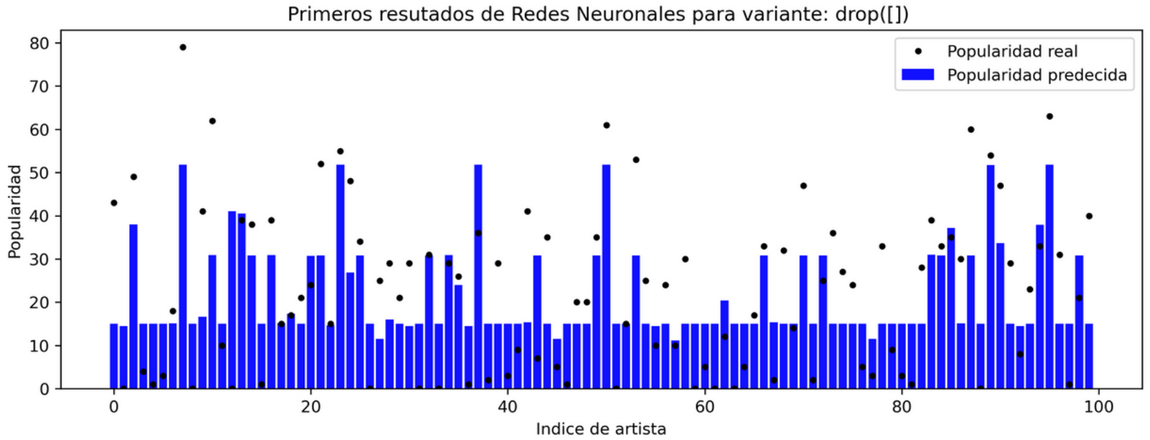
6. ÁNALISIS DE RESULTADOS

6.2. ÁRBOLES DE DECISIONES

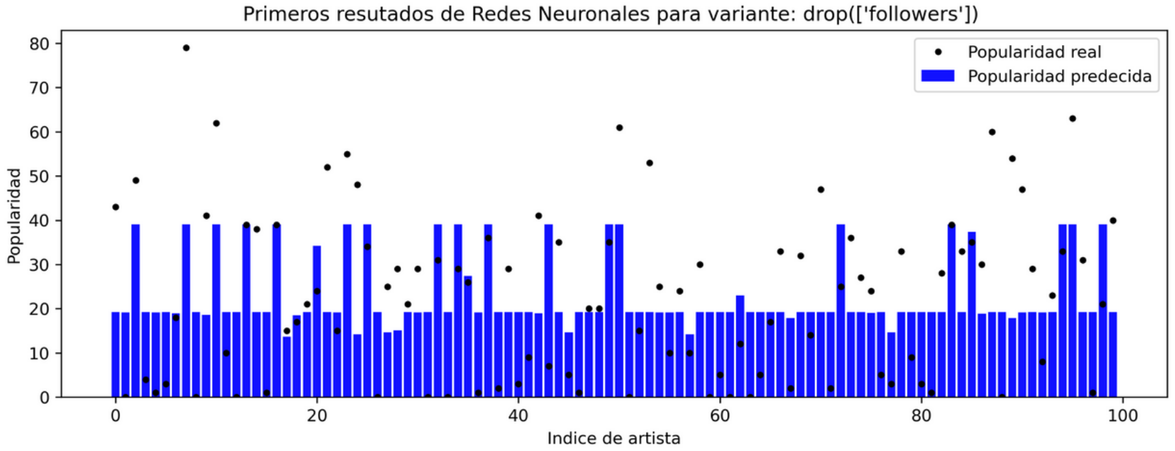


6. ÁNALISIS DE RESULTADOS

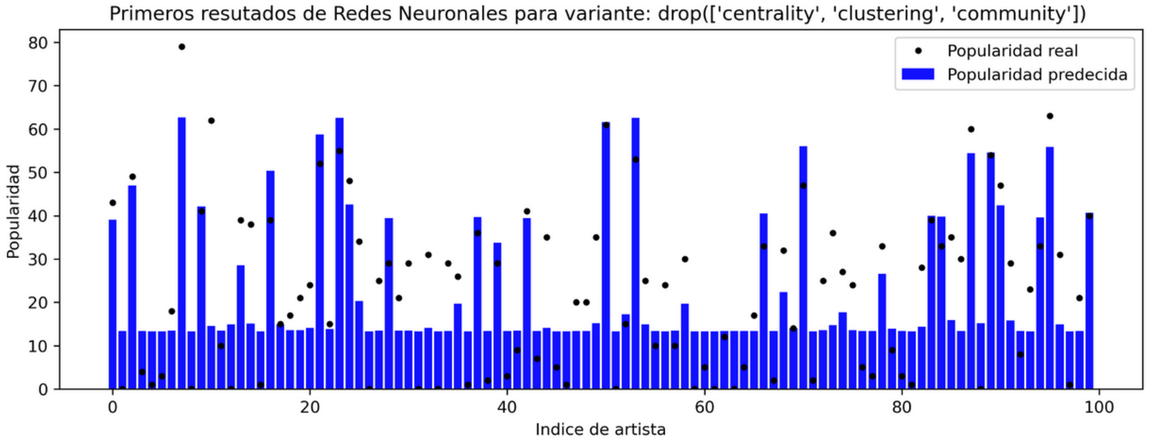
6.3. REDES NEURONALES



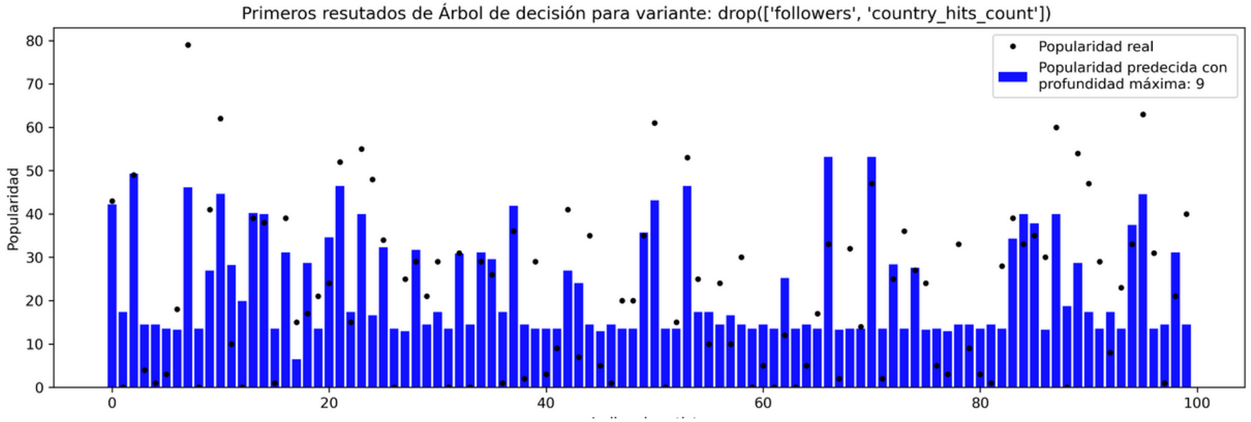
Todos los atributos (drop([]))



Todos los atributos menos followers (drop([’followers’]))



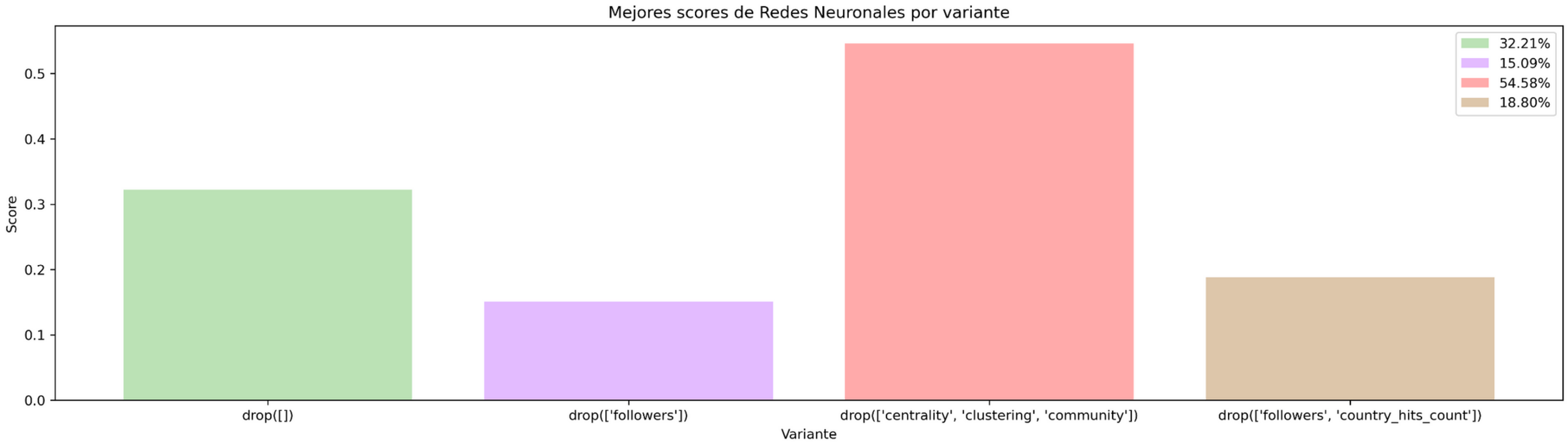
Atributos no relacionales (drop([’centrality’,’clustering’,’community’]))



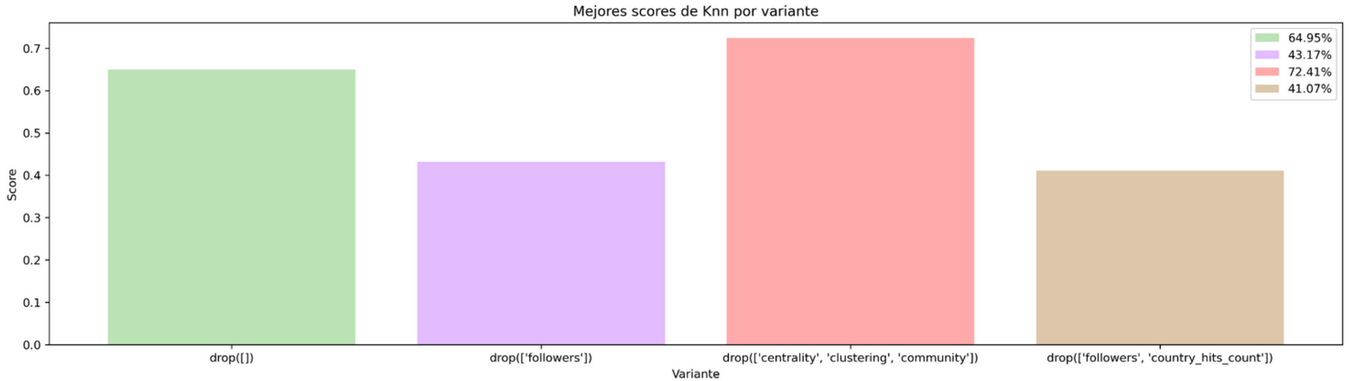
Atributos relacionales (drop([’followers’,’country-hits-count’]))

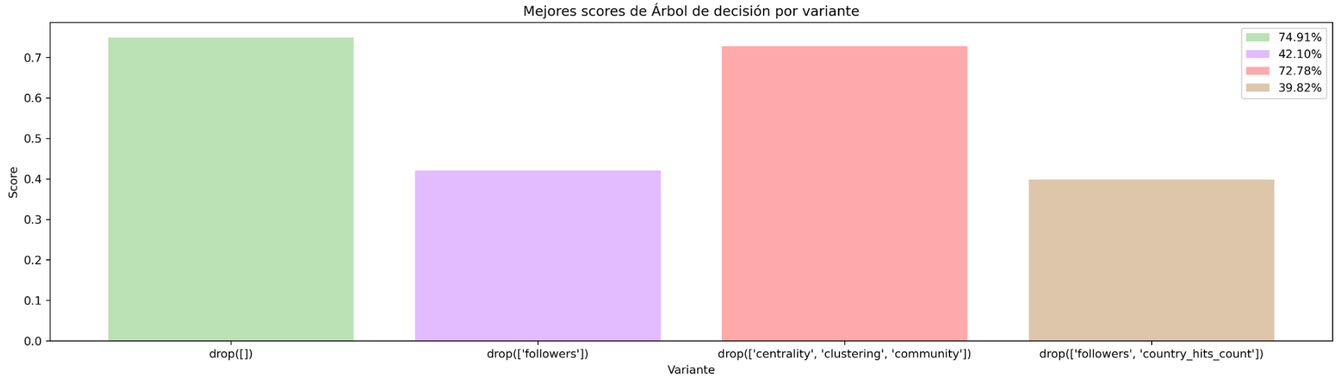
6. ÁNALISIS DE RESULTADOS

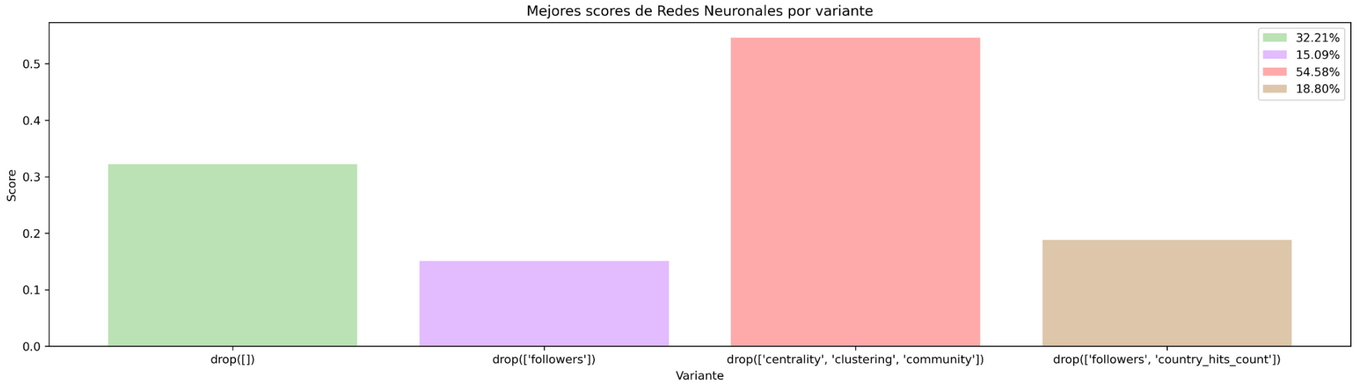
6.3. REDES NEURONALES



7. CONLUSIONES

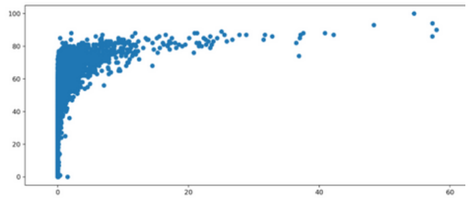






Followers es determinante:

- Correlación con popularidad



CountryHitCount mejora parcialmente

Influencia de atributos relacionales depende del modelo

- Arboles: mejora
- Knn: empeora
- Redes neuronales: empeora mas

Mejor resultado:

- Arboles con relacionales y no relacionales (74.91%)

Conclusión final:
De forma general los atributos relacionales mejoran la precisión. Sin embargo dependen de:

- Dominio del problema
- Correlación entre atributos
- Modelos de aprendizaje automatico empleados

En este caso los atributos relacionales proporcionan una mejoria en la precisión.

MUCHAS GRACIAS

JAVIER FERNÁNDEZ CASTILLO
MANUEL OTERO BARBASÁN

INTELIGENCIA ARTIFICIAL
JUNIO DE 2023