

ESTUDIO DE TÉCNICAS DE APRENDIZAJE AUTOMÁTICO PARA RECOMENDAR LOCALIZACIONES SEGÚN EL TIPO DE NEGOCIO

Daniel González Pascual

Escuela Politécnica Superior

Ingeniería Informática

Trabajo Fin de Grado



1.

SISTEMAS DE RECOMENDACIÓN

Introducción

- ▷ Sugieren *items* que podrían ser interés del usuario.
- ▷ Se utilizan, por ejemplo, puntuaciones, compras o búsquedas.
- ▷ Los *items* son ropa, vídeos, personas, noticias...
Cualquier cosa que tenga sentido ser buscada.

The screenshot shows the Netflix homepage with the following sections:

- Because you watched Star Trek: Discovery**: Star Trek: Enterprise, MARS, DOCTOR WHO, PINE GAP, GODZILLA, TRAVELERS.
- NETFLIX ORIGINALS**: SEX EDUCATION, UNBREAKABLE KIMMY SCHMIDT, KINGDOM, POLAR, IO, TED BUNDY TAPES.
- Watch It Again**: The Good Place, STAR TREK DISCOVERIES, BLACK MIRROR BANDERSNATCH, ROMA, THE NIGHT COMES FOR US.

At the bottom, there is an Amazon.es navigation bar with a search bar and account options.

Los clientes que compraron este producto también compraron



ZEFAL CO2 16g Blister 2
Cartuchos, Deportes, Plata
★★★★☆ 39
3,70 €



Buff R-Flash Logo Gorra,
Unisex Adulto
★★★★☆ 89
18,95 € - 44,95 €



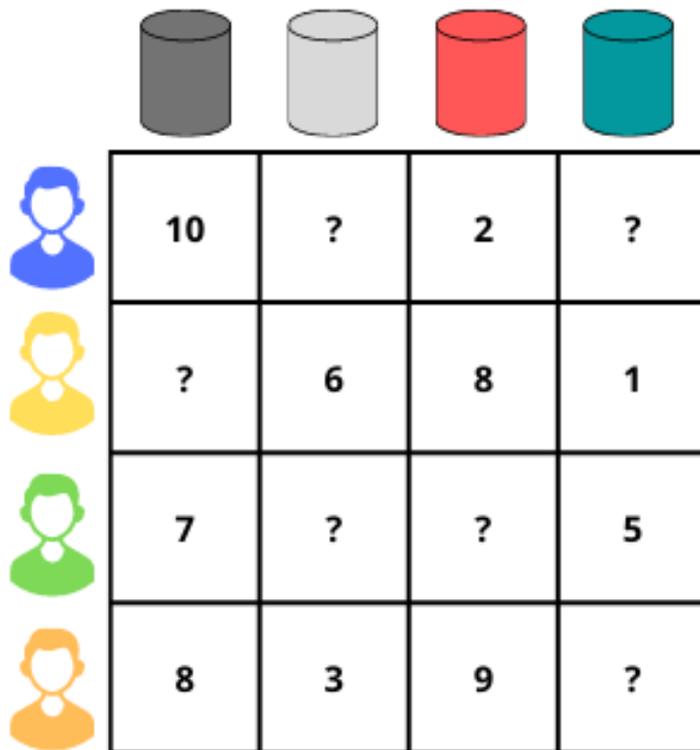
Tufo Extreme Líquido
Antipinchazos, Unisex
Adulto, Negro, Talla Única
★★★★☆ 45
7,50 € - 18,47 €

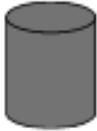


Garmin Fénix 3 - Reloj
★★★★☆ 552
422,61 € - 799,99 €

Estructura general

- ▷ Se necesitan datos de los usuarios y los *items*.
- ▷ Se requiere de una escala (*rating*) de cuánto le gusta el item a un usuario.
- ▷ Los *ratings* suele tener valores entre los intervalos $[1, 10]$, $[1, 5]$ o $[0.1, 1.0]$.



				
	10	?	2	?
	?	6	8	1
	7	?	?	5
	8	3	9	?

Tipos de sistemas

Filtrado colaborativo



Filtrado basado en contenido

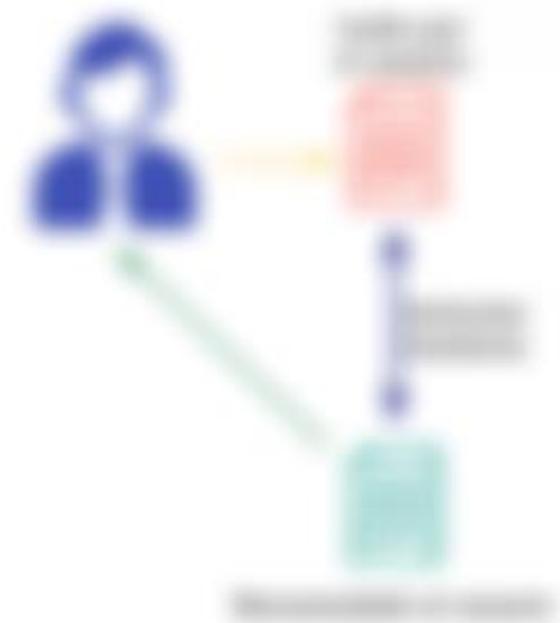


Tipos de sistemas

Filtrado colaborativo



Filtrado basado en contenido



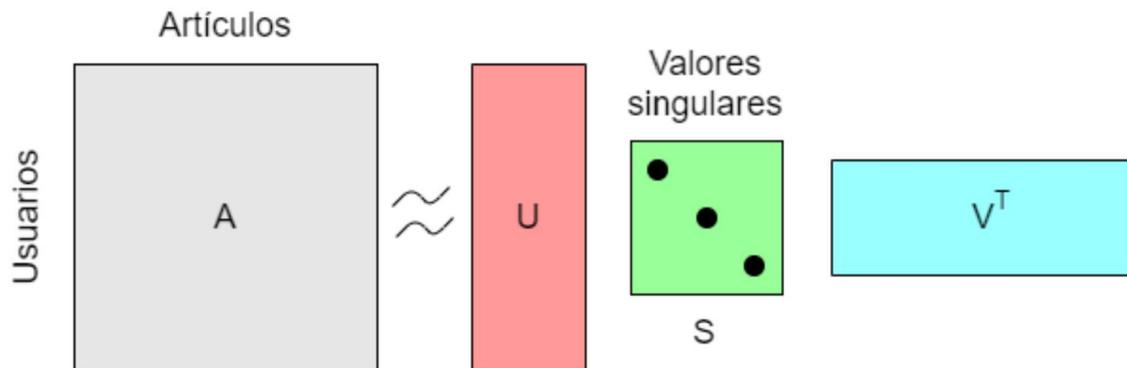
Filtrado colaborativo



- ▷ Basado en memoria.
- ▷ Basado en modelo:
 - Factorización de matrices.

Factorización de matrices

Técnica matemática SVD



Factorización de matrices

Algoritmo de optimización descenso por gradiente

$$e(u, i) = r(u, i) - \hat{r}(u, i)$$

$$p_u = p_u + \alpha \cdot [e(u, i) \cdot q_i - \lambda \cdot p_u]$$

$$q_i = q_i + \alpha \cdot [e(u, i) \cdot p_u - \lambda \cdot q_i]$$

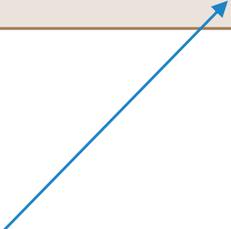
Tasa de aprendizaje



$$b_u = b_u + \alpha \cdot [e(u, i) - \lambda \cdot b_u]$$

$$b_i = b_i + \alpha \cdot [e(u, i) - \lambda \cdot b_i]$$

Parámetro regularizador

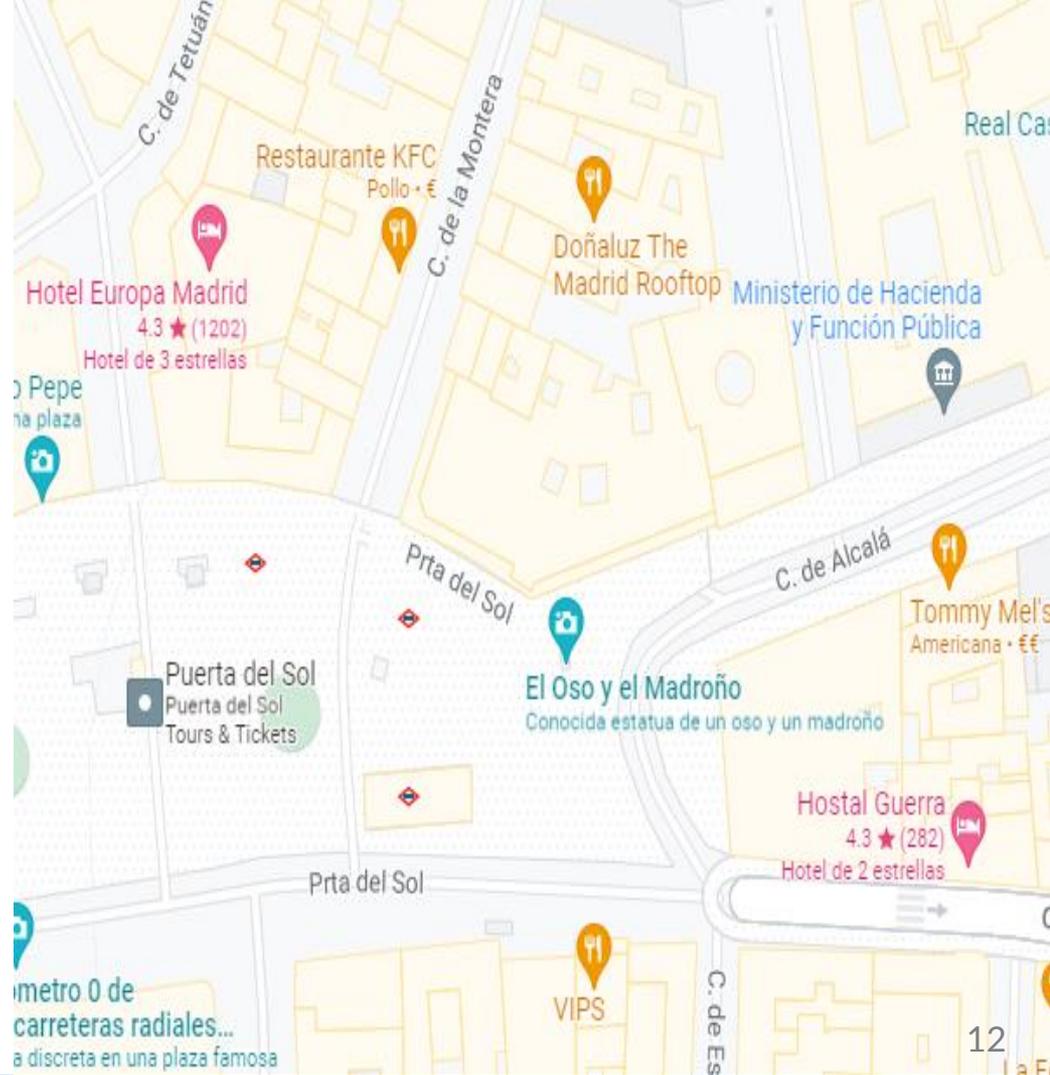


2.

MOTIVACIÓN

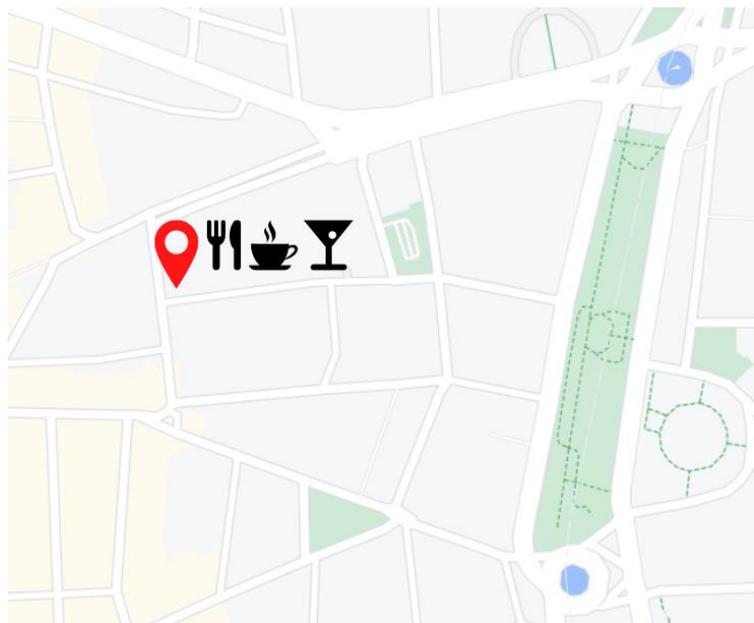
Location-Based Services (LBS)

- ▷ Nuevo dato con el que trabajar: las localizaciones.
- ▷ Características por cada *Point Of Interest (POI)*.

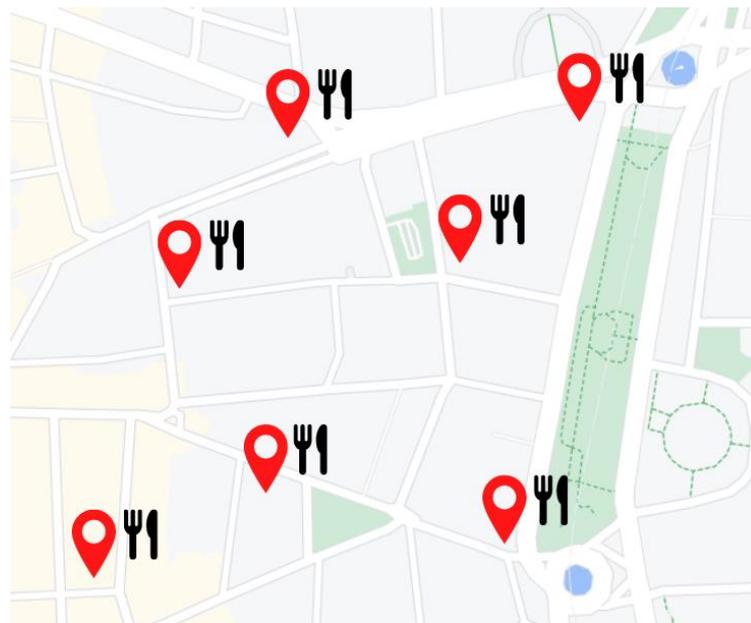


Recomendación basada en la localización

Recomendación de tipos de tiendas

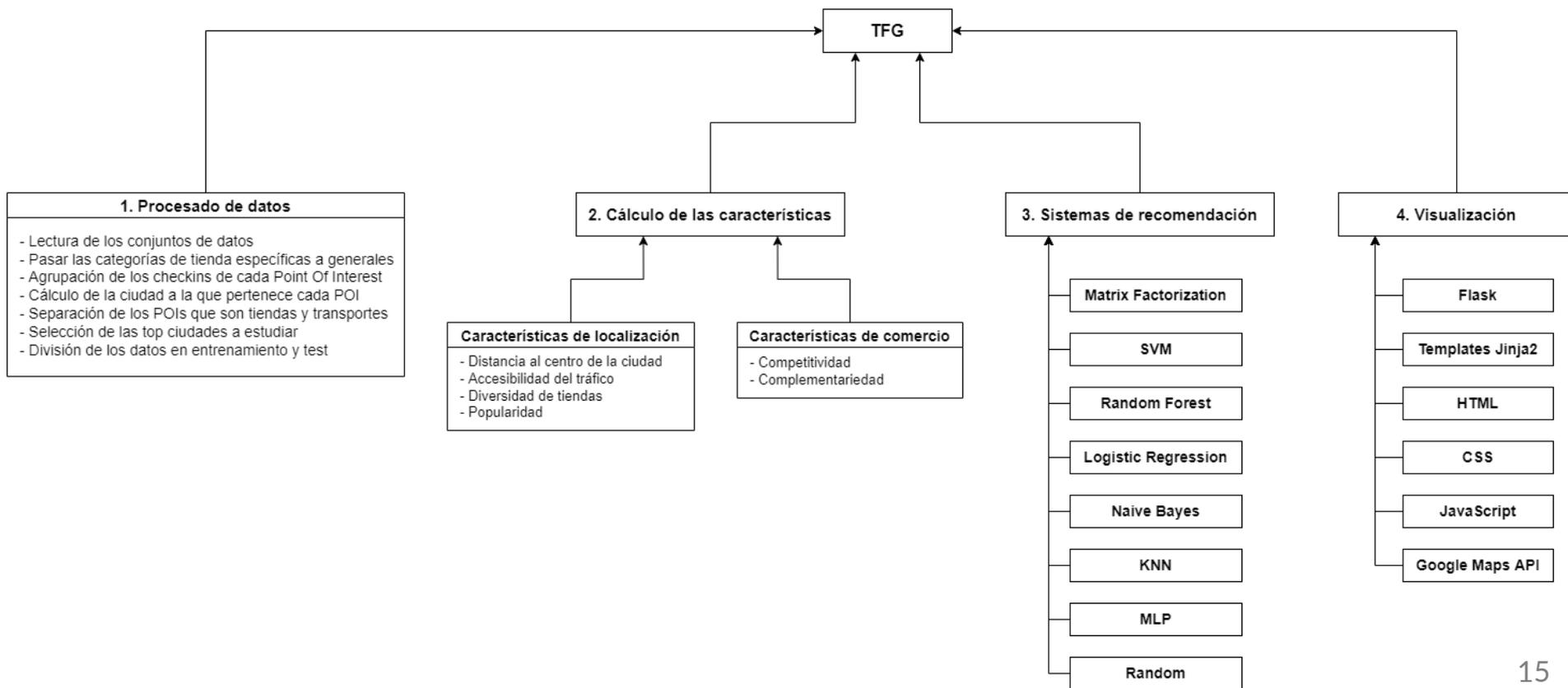


Recomendación de localizaciones



3. DISEÑO

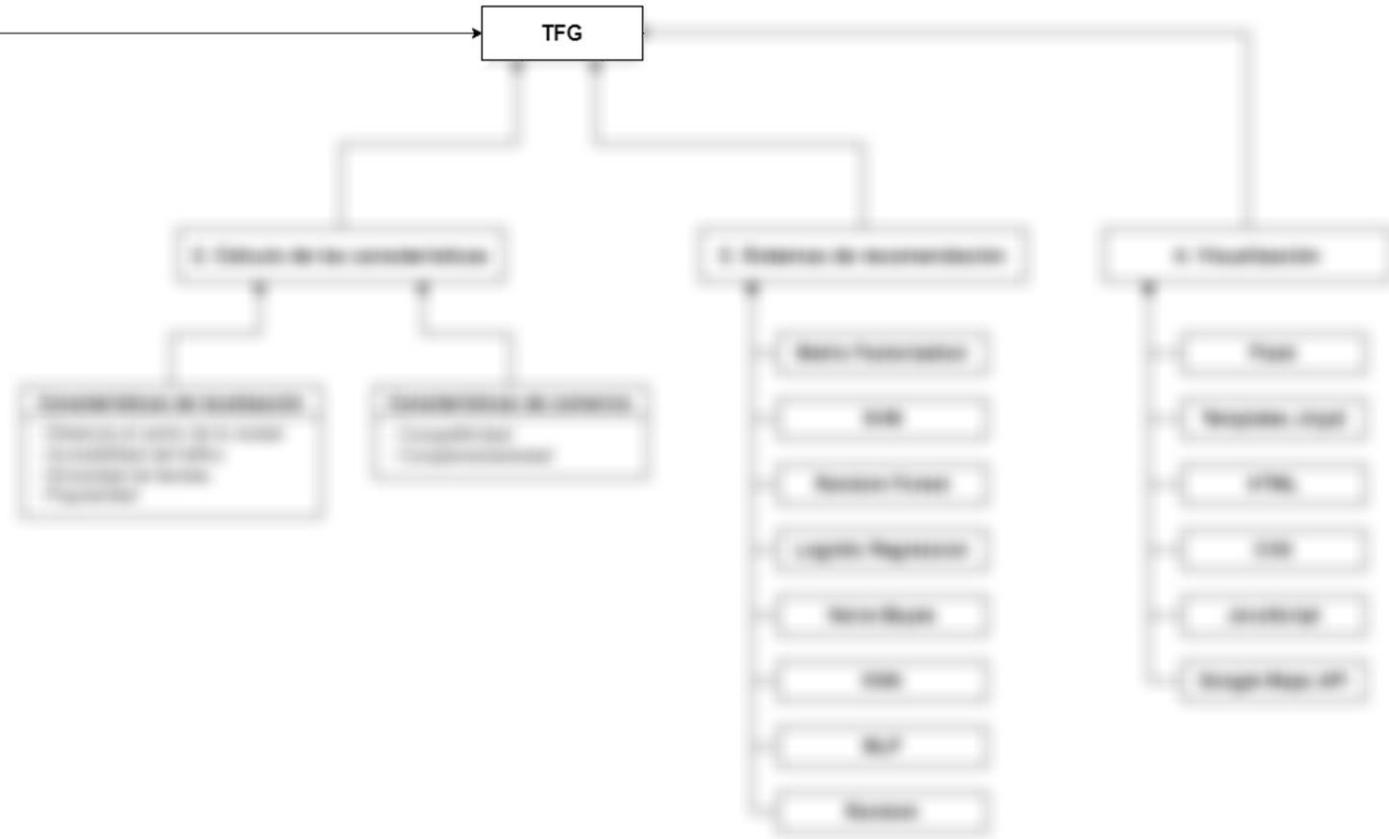
Estructura general



TFG

1. Procesado de datos

- Lectura de los conjuntos de datos
- Pasar las categorías de tienda específicas a generales
- Agrupación de los checkins de cada Point Of Interest
- Cálculo de la ciudad a la que pertenece cada POI
- Separación de los POIs que son tiendas y transportes
- Selección de las top ciudades a estudiar
- División de los datos en entrenamiento y test



Procesado de datos



▷ POIs:

Venue ID	Latitude	Longitude	Venue category name	Country code
3fd66200f964a52000e71ee3	40.73	-74	Jazz Club	US
3fd66200f964a52000e81ee3	40.76	-73.98	Gym	US
3fd66200f964a52000ea1ee3	40.73	-74	Indian Restaurant	US
3fd66200f964a52000ee1ee3	39.93	-75.16	Sandwich Place	US
3fd66200f964a52000f11ee3	40.65	-74	Bowling Alley	US

▷ Checkins:

User ID	Venue ID	UTC time	Timezone offset in minutes
50756	4f5e3a72e4b053fd6a4313f6	Tue Apr 03 18:00:06 +0000 2012	240
190571	4b4b87b5f964a5204a9f26e3	Tue Apr 03 18:00:07 +0000 2012	180
221021	4a85b1b3f964a520eefe1fe3	Tue Apr 03 18:00:08 +0000 2012	-240
66981	4b4606f2f964a520751426e3	Tue Apr 03 18:00:08 +0000 2012	-300
21010	4c2b4e8a9a559c74832f0de2	Tue Apr 03 18:00:09 +0000 2012	240

Procesado de datos



▷ Ciudades:

City name ▼	Latitude of city center ▼	Longitude of city center ▼	Country code ▼	Country name ▼	City type ▼
New York	40.71	-73.91	US	United States	Other
Harrisburg	40.27	-76.9	US	United States	Provincial capital
Trenton	40.22	-74.78	US	United States	Provincial capital
Philadelphia	39.93	-75.22	US	United States	Other
Boston	42.37	-71.1	US	United States	Provincial capital

▷ Categorías:

Level1_name ▼	Level2_name ▼	Level3_name ▼	Level4_name ▼
Food	Asian Restaurants	Japanese Restaurants	Donburi Restaurants
Food	Asian Restaurants	Japanese Restaurants	Japanese Curry Restaurants
Shops & Services	Food & Drink Shops	Cheese Shops	
Shops & Services	Food & Drink Shops	Fish Markets	
Shops & Services	Food & Drink Shops	Food Services	

Procesado de datos

Diferencia de datos

Conjuntos de datos	2015	2019
Checkins	33 263 633	22 809 624
POIs	3 680 126	11 180 160

Procesado de datos

Transformación de las categorías

Level1_name ▼	Level2_name ▼	Level3_name ▼	Level4_name ▼
Food	Asian Restaurants	Japanese Restaurants	Donburi Restaurants
Food	Asian Restaurants	Japanese Restaurants	Japanese Curry Restaurants
Shops & Services	Food & Drink Shops	Cheese Shops	
Shops & Services	Food & Drink Shops	Fish Markets	
Shops & Services	Food & Drink Shops	Food Services	

Procesado de datos

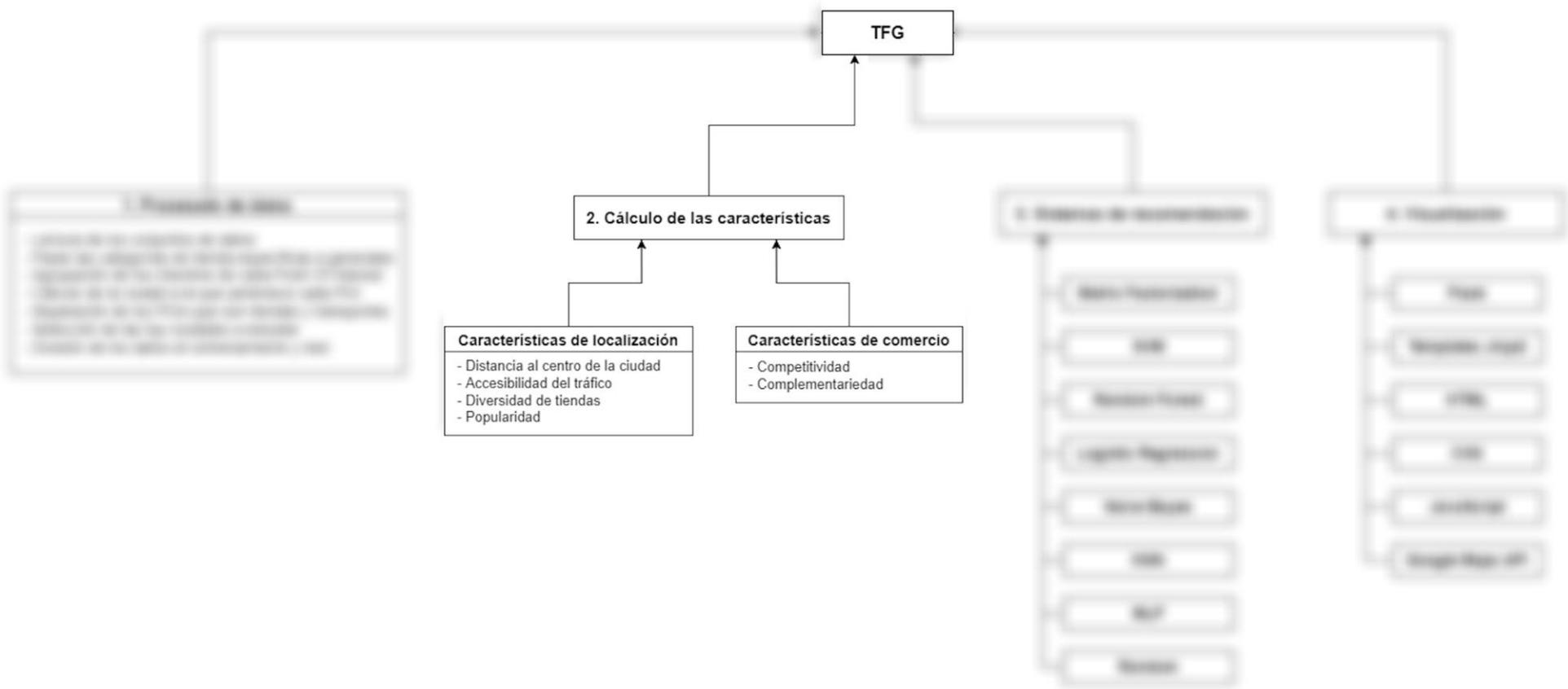
Estructura de datos una vez procesados

Venue ID	Latitude	Longitude	Venue category name	Checkins	Distance to city centre
3fd66200f964a52008e81ee3	40.76	-73.97	Dessert Shop	139	7.85
3fd66200f964a52008e91ee3	40.74	-73.99	Cafe	87	7.61
3fd66200f964a52008eb1ee3	40.78	-73.95	Bar	30	9.25
3fd66200f964a52000e81ee3	40.76	-73.98	Athletic & Sport	24	8.16
3fd66200f964a52001e91ee3	40.73	-73.95	Asian Restaurant	0	4.88

Procesado de datos

$$rating = \frac{\log(num_checkins)}{\log(limit)} \cdot 10$$

					
	9	0	0	0	0
	0	2	0	0	0
	0	0	0	0	4
	0	0	0	8	0
	0	0	6	0	0



Características de localización

- ▷ Distancia al centro de la ciudad:

$$DC = \frac{1}{\log(d)}$$

Distancia al centro de la ciudad

- ▷ Accesibilidad del tráfico:

$$AT = \sum_{t \in T} \frac{\log_2[n(t, r) + 1]}{\log_2[d(t)]}$$

Número de facilidades para acceder al transporte t en la zona con radio r

Distancia mínima al transporte t

Características de localización

▷ Diversidad de tiendas:

$$DT = - \sum_{s \in S} \left[\frac{n(s, r)}{n(r)} \times \log \frac{n(s, r)}{n(r)} \right]$$

Número de tiendas de tipo s
en la zona con radio r

Número total de tiendas
en la zona con radio r

▷ Popularidad:

$$P = \log \sum_{s \in R} C(s)$$

Número de checkins de la tienda s

Conjunto de tiendas de la zona

Características de comercio

▷ Competitividad:

$$C_s = \frac{n(s, r)}{n(r)}$$

Número de tiendas del tipo s
en la zona con radio r

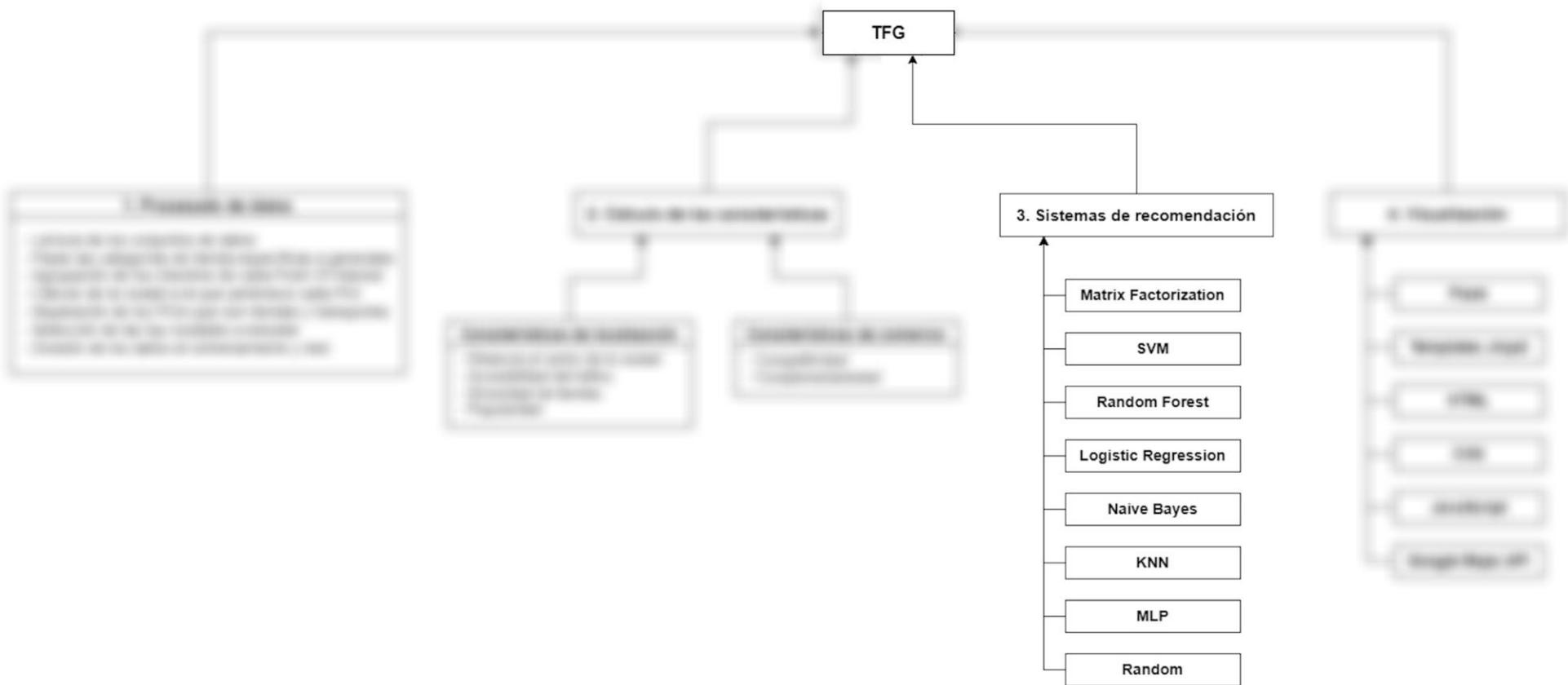
Número total de tiendas
en la zona con radio r

▷ Complementariedad:

$$a_{s \rightarrow s^*} = \frac{2 \cdot n(s, s^*)}{n \cdot (n + 1)}$$

Número de apariciones que tiene el par
de tiendas s y s^* en las distintas zonas

Número total de tiendas



Entrenamiento del recomendador

Sesgo global (valoración media)

Características de comercio

Características de localización

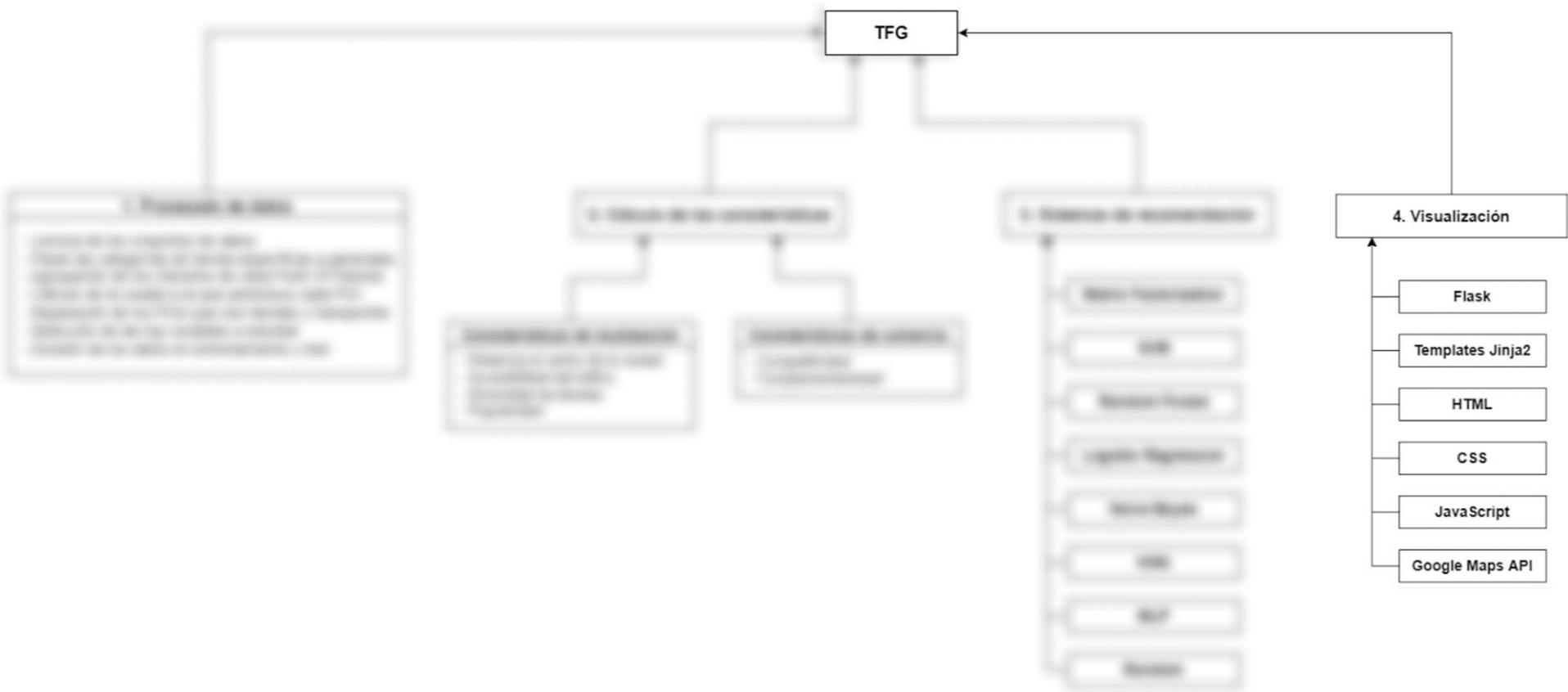
$$\hat{r}(t, l) = b + B(t, l)F(t, l) + P(t, l)C(t, l) + T_t^T L_l$$

Vectores de los sesgos para el tipo de tienda t y la localización l

Vectores de los factores latentes del tipo de tienda t y la localización l

Evaluación del recomendador

```
input : Tipo de tienda  $t$   
output: Ránking de localizaciones según el rating predicho  
1  $muestras \leftarrow$  POIs del entrenamiento con  $rating > 0$  y tipo de tienda  $t$ ;  
2 for  $m$  in  $muestras$  do  
3   |  $clase[m.rating] \leftarrow m$ ;  
4 end  
5 foreach  $l$  in  $localizaciones$  do  
6   |  $l[tipo\_tienda] \leftarrow t$ ;  
7   |  $C_l \leftarrow DC_l, AT_l, DT_l, P_l$ ;  
8   |  $F_l \leftarrow Compet_l, Complem_l$ ;  
9   | for  $r \leftarrow 1$  to 10 do  
10  | |  $dist[r] \leftarrow \sum_{c \in clase[r]} \frac{dist\_euclidea(C_l, C_c) + dist\_euclidea(F_l, F_c)}{long(clase[r])}$ ;  
11  | end  
12  |  $clase\_elegida \leftarrow clase[\min(dist)]$ ;  
13  |  $nuevos\_parametros \leftarrow \frac{clase\_elegida.parametros}{long(clase\_elegida)}$ ;  
14  |  $rating\_predicho \leftarrow predecir\_rating(C_l, F_l, nuevos\_parametros)$ ;  
15  |  $ranking \leftarrow$  añadir la localización  $l$  con su  $rating\_predicho$ ;  
16 end
```



Recomendador Matrix Factorization

¿No sabes dónde montar tu tienda?

Nosotros te ayudamos



Obtener las mejores localizaciones

Ciudad donde se quiere montar el negocio

Tokyo



Tipo de negocio que se quiere montar

Bar



Algoritmo con el que comparar

Ninguno



[VER RECOMENDACIONES](#)



Mapa

Satélite

¿Quieres hacer otra consulta?

Ciudad

Tokyo



Tipo de negocio

Bar

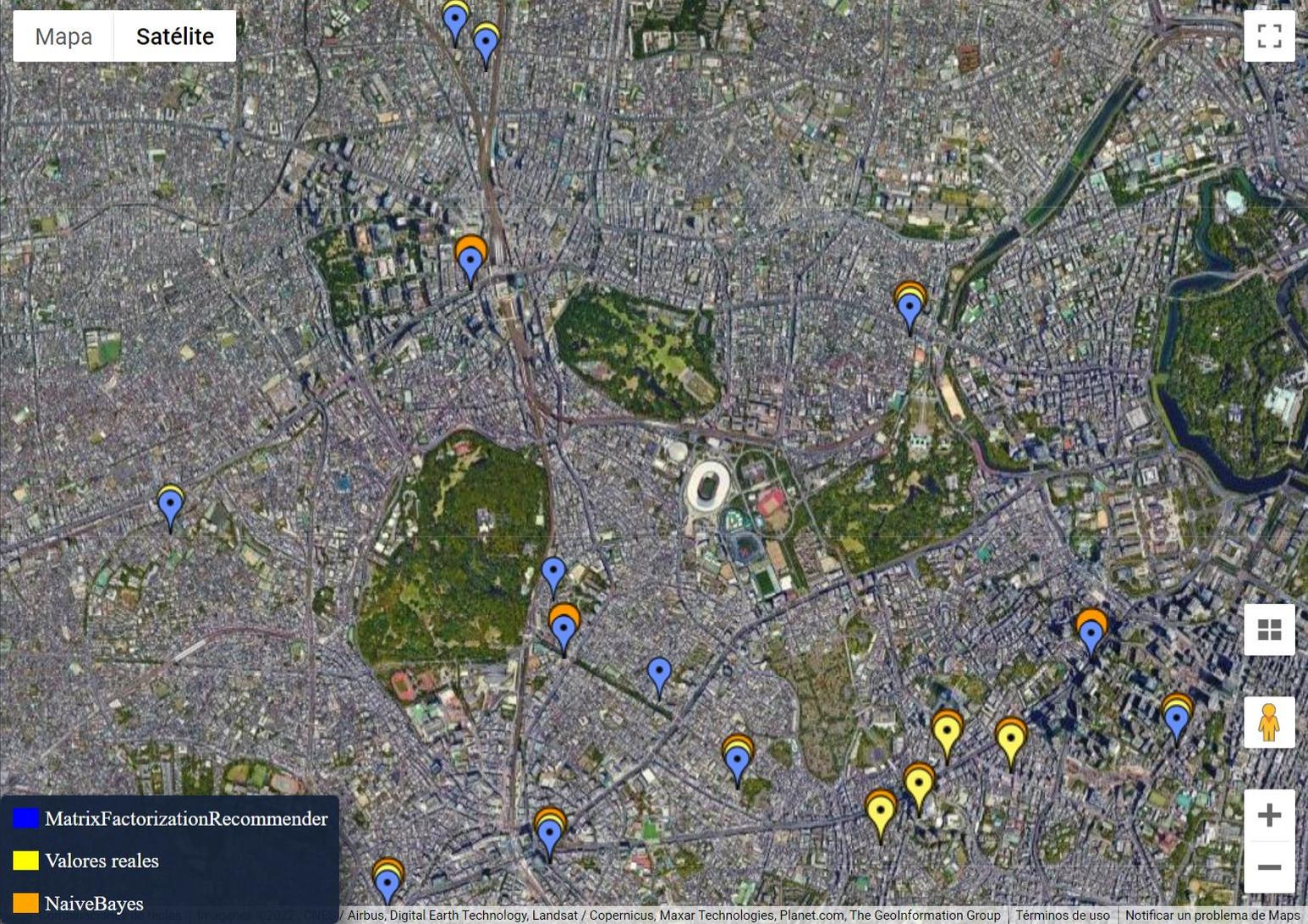


Algoritmo

Ninguno



CALCULAR



 MatrixFactorizationRecommender

 Valores reales

 NaiveBayes





Mapa

Satélite

¿Quieres hacer otra consulta?

Ciudad

Tokyo



Tipo de negocio

Bar

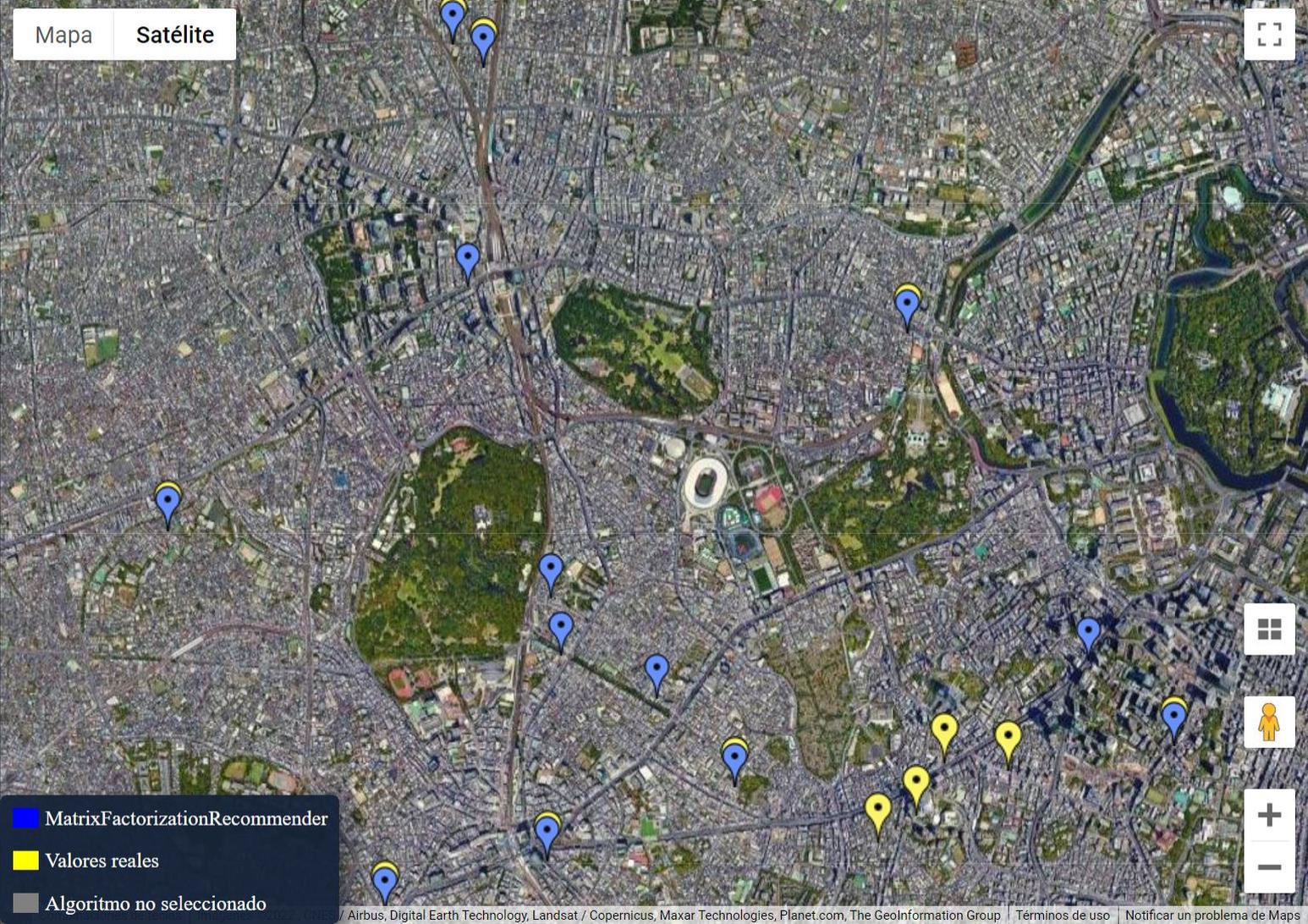


Algoritmo

Ninguno



CALCULAR



MatrixFactorizationRecommender

Valores reales

Algoritmo no seleccionado

4. PRUEBAS

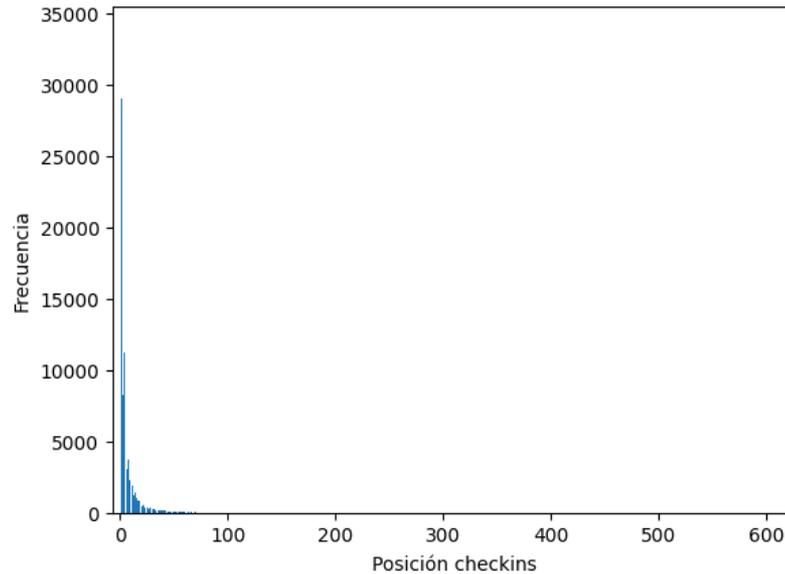
Adaptación de los datos

Entorno de pruebas

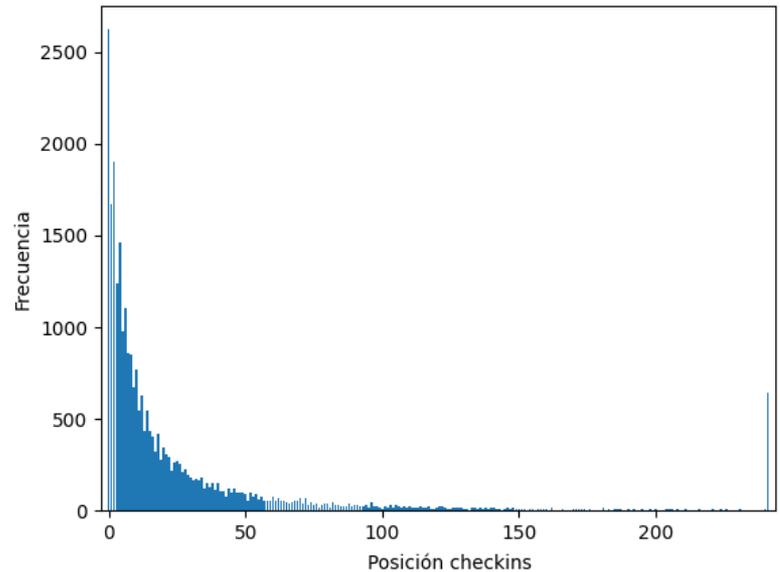
Recursos	Características
CPU	Intel(R) Xeon(R)
GPU	NVIDIA Tesla K80/T4/P100
RAM	13GB
S.O.	Ubuntu 18.04.5 LTS
Python	3.7.13

Adaptación de los datos

Distribución original



Distribución aplicando límites



Adaptación de los datos

Datos de las ciudades

Ciudad	POIs	Checkins
Nueva York	95 932	765 183
Tokio	174 463	2 440 946
Madrid	82 149	288 004

Originales



Ciudad	POIs	Checkins
Nueva York	13 558	532 085
Tokio	20 000	812 177
Madrid	4 434	151 410

Tras adaptar los datos

Adaptación de los datos

Conjuntos seleccionados:

Primer conjunto: Bar, Cafe, Dessert Shop, Food & Drink Shop, Fast Food Restaurant.

Segundo conjunto: Office, Bank, Bar, Cafe, Fast Food Restaurant.

Tercer conjunto: Athletic & Sport, Hotel, Clothing Store, Food & Drink Shop, Bar.

Ajuste de hiperparámetros

▷ Pruebas con:

- Conjunto de datos de Nueva York.
- Primer conjunto de tipos de tiendas.
- Datos de entrenamiento (70%) y datos de test (30%).

▷ Valor óptimo en base a:

- MAE y RMSE bajo.
 - Predicción alta.
- } Sin sobreentrenamiento



Experimentos con el primer conjunto

Ciudad	Modelo	R-Precision	MAP	MRR	Tiempo de ejecución (seg.)
Nueva York	Factorización de matrices	0.24	0.22	0.42	12 372.31
	SVM	0.27	0.24	0.39	9 932.84
	Random Forest	0.23	0.26	0.47	9 808.13
	Regresión logística	0.22	0.22	0.44	9 771.42
	Naive Bayes	0.25	0.24	0.56	9 754.07
	KNN	0.16	0.21	0.63	9 755.51
	Perceptrón multicapa	0.23	0.23	0.45	10 529.22
	Aleatorio	0.15	0.22	0.38	9 753.69
Tokio	Factorización de matrices	0.22	0.21	0.34	27 284.38
	SVM	0.24	0.24	0.35	23 778.35
	Random Forest	0.23	0.24	0.14	23 182.84
	Regresión logística	0.21	0.23	0.45	23 134.42
	Naive Bayes	0.19	0.21	0.19	23 114.70
	KNN	0.20	0.21	0.39	23 114.84
	Perceptrón multicapa	0.20	0.20	0.52	24 045.01
	Aleatorio	0.19	0.21	0.31	23 114.82
Madrid	Factorización de matrices	0.20	0.27	0.52	1 229.66
	SVM	0.24	0.28	0.49	788.35
	Random Forest	0.27	0.28	0.54	777.26
	Regresión logística	0.29	0.28	0.49	772.84
	Naive Bayes	0.21	0.24	0.34	767.49
	KNN	0.19	0.22	0.36	767.47
	Perceptrón multicapa	0.26	0.30	0.42	1 050.42
	Aleatorio	0.18	0.23	0.27	767.46

Experimentos con el segundo conjunto

Ciudad	Modelo	R-Precision	MAP	MRR	Tiempo de ejecución (seg.)
Nueva York	Factorización de matrices	0.23	0.23	0.38	13 637.90
	SVM	0.22	0.24	0.44	12 132.58
	Random Forest	0.23	0.24	0.64	12 006.91
	Regresión logística	0.24	0.23	0.52	11 971.15
	Naive Bayes	0.25	0.26	0.51	11 953.96
	KNN	0.21	0.21	0.38	11 953.94
	Perceptrón multicapa	0.21	0.23	0.37	12 739.49
	Aleatorio	0.17	0.23	0.33	11 953.68
Tokio	Factorización de matrices	0.20	0.21	0.31	21 186.69
	SVM	0.24	0.24	0.31	20 286.18
	Random Forest	0.25	0.25	0.47	19 697.19
	Regresión logística	0.27	0.25	0.29	19 638.62
	Naive Bayes	0.21	0.23	0.34	19 629.19
	KNN	0.20	0.22	0.35	19 629.07
	Perceptrón multicapa	0.21	0.22	0.48	20 447.97
	Aleatorio	0.20	0.21	0.30	19 629.20
Madrid	Factorización de matrices	0.28	0.33	0.71	1 427.60
	SVM	0.28	0.29	0.42	1 308.21
	Random Forest	0.28	0.28	0.30	1 297.75
	Regresión logística	0.30	0.30	0.53	1 292.17
	Naive Bayes	0.21	0.24	0.43	1 287.00
	KNN	0.17	0.22	0.26	1 287.02
	Perceptrón multicapa	0.27	0.28	0.50	1 564.01
	Aleatorio	0.20	0.23	0.24	1 286.91

Experimentos con el tercer conjunto

Ciudad	Modelo	R-Precision	MAP	MRR	Tiempo de ejecución (seg.)
Nueva York	Factorización de matrices	0.23	0.22	0.48	11 578.03
	SVM	0.16	0.17	0.13	10 531.94
	Random Forest	0.13	0.17	0.35	10 401.55
	Regresión logística	0.20	0.20	0.17	10 360.07
	Naive Bayes	0.16	0.19	0.32	10 353.23
	KNN	0.18	0.21	0.28	10 354.23
	Perceptrón multicapa	0.13	0.17	0.36	10 950.18
	Aleatorio	0.14	0.23	0.32	10 353.42
Tokio	Factorización de matrices	0.22	0.23	0.24	19 456.92
	SVM	0.20	0.21	0.35	18 907.40
	Random Forest	0.19	0.22	0.10	18 310.64
	Regresión logística	0.20	0.25	0.38	18 260.46
	Naive Bayes	0.23	0.25	0.34	18 242.00
	KNN	0.22	0.21	0.27	18 242.88
	Perceptrón multicapa	0.20	0.23	0.19	19 247.50
	Aleatorio	0.22	0.23	0.24	18 241.54
Madrid	Factorización de matrices	0.23	0.24	0.25	1 347.90
	SVM	0.33	0.29	0.53	1 225.04
	Random Forest	0.26	0.29	0.27	1 214.78
	Regresión logística	0.29	0.30	0.65	1 209.32
	Naive Bayes	0.31	0.29	0.24	1 204.05
	KNN	0.23	0.26	0.66	1 203.96
	Perceptrón multicapa	0.29	0.29	0.54	1 487.81
	Aleatorio	0.19	0.25	0.23	1 204.03

5. CONCLUSIONES

Análisis de los resultados

- ▷ Métricas muy similares a otros algoritmos de *Scikit-learn* u otros estudios con diferentes implementaciones.
- ▷ Alto coste en tiempo de ejecución.
- ▷ La eficacia del recomendador ha estado condicionada por:
 - Número de datos disponible de cada ciudad.
 - Tiempo limitado de las sesiones del entorno.



6.

TRABAJO FUTURO

Propuestas

- ▷ Realizar más pruebas.
- ▷ Utilizar otros datos.
- ▷ Optimizar el algoritmo.
- ▷ Permitir al experto decidir si utilizar la recomendación.
- ▷ Ampliar las características:
 - Iluminación.
 - Contaminación.



Muchas gracias

Turno de preguntas

Daniel González Pascual

Ingeniería Informática

Escuela Politécnica Superior