



Instituto Tecnológico
de Buenos Aires

27/06/2024

82.20 Análisis Predictivo Avanzado - Segundo Trabajo Práctico

Azul de los Angeles Makk (61589) y Bruno Soifer (62423)

—

Agenda

01

Introducción

Problema de negocio a resolver

03

EDA

Gráficos exploratorios

05

Conclusión

Acciones que se podrían tomar para mitigar posibles problemas

02

Dataset a trabajar

Introducción a la base y a sus variables

04

Sistema de recomendación

Desarrollo de la solución

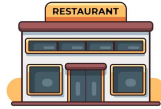
INTRODUCCIÓN












Introducción

Problema de negocio a resolver: poder aplicar un sistema de recomendación para sugerir ítems relevantes a usuarios.

Entendiendo a:

1. Un usuario como un posible cliente, con distintas características.
2. Un ítem como un restaurant



DATASET



Dataset a trabajar



- El dataset fue generado con el objetivo de :
 - Generar una lista con los mejores restaurants de acuerdo a las preferencias de los consumidores.
 - Definir los features más importantes
- Se procesaron dos datasets a trabajar:
 - **Locales:** contiene información de los distintos restaurantes
 - **Usuarios:** contiene información sobre el consumidor
- Además, se tuvieron en cuenta:
 - **Ratings** de los usuarios a los locales que asistieron

Dataset a trabajar

- Usuarios

```
Index(['userID', 'latitude', 'longitude', 'smoker', 'drink_level',  
      'dress_preference', 'ambience', 'transport', 'marital_status', 'hijos',  
      'birth_year', 'interest', 'personality', 'religion', 'activity',  
      'color', 'weight', 'budget', 'height', 'Rcuisine', 'Upayment', 'rating',  
      'food_rating', 'service_rating'],  
      dtype='object')
```

- Local

```
Index(['placeID', 'latitude', 'longitude', 'the_geom_meter', 'name', 'address',  
      'city', 'state', 'country', 'fax', 'zip', 'alcohol', 'smoking_area',  
      'dress_code', 'accessibility', 'price', 'url', 'Rambience', 'franchise',  
      'area', 'other_services', 'Rpayment', 'Rcuisine', 'hours', 'days',  
      'parking_lot', 'rating', 'food_rating', 'service_rating'],  
      dtype='object')
```

Dataset a trabajar

Locales: 130 filas y 29 columnas

'placeID', 'latitude', 'longitude', 'name', 'address', 'city', 'state', 'country', 'fax', 'zip', 'alcohol', 'smoking_area', 'dress_code', 'accessibility', 'price', 'url', 'Rambience', 'franchise', 'area', 'other_services', 'Rpayment', 'Rcuisine', 'hours', 'days', 'parking_lot', 'rating', 'food_rating', 'service_rating'

name	address	city	state	country	fax	...	area	other_services	Rpayment	Rcuisine	hours	day	parking_lot	rating	food_rating	service_rating
Kiku Cuernavaca	Revolucion	Cuernavaca	Morelos	Mexico	?	...	closed	none	NaN	Japanese	11:00-21:00;	Mon;Tue;Wed;Thu;Fr	none	1.600000	1.600000	1.600000
puesto de tacos	esquina santos degollado y leon guzman	s.l.p.	s.l.p.	mexico	?	...	open	none	cash	Mexican	09:00-12:00;	Mon;Tue;Wed;Thu;Fr	none	1.281250	1.343750	0.937500
El Rincón de San Francisco	Universidad 169	San Luis Potosi	San Luis Potosi	Mexico	?	...	open	none	cash	Mexican	18:00-23:30;	Mon;Tue;Wed;Thu;Fr	none	1.200000	1.200000	1.200000
little pizza Emilio Portes Gil	calle emilio portes gil	victoria	tamaulipas	?	?	...	closed	none	cash	Armenian	00:00-23:30;	Mon;Tue;Wed;Thu;Fr	none	1.250000	2.000000	1.250000
carnitas_mata	lic. Emilio portes gil	victoria	Tamaulipas	Mexico	?	...	closed	none	cash	Mexican	08:00-16:00;	Mon;Tue;Wed;Thu;Fr	yes	1.166667	1.333333	1.000000
...
Chaires	Ricardo B. Anaya	San Luis Potosi	San Luis Potosi	Mexico	?	...	closed	none	cash	Bakery	09:00-22:00;	Mon;Tue;Wed;Thu;Fr	yes	1.400000	1.400000	1.400000
Sushi Itto	Venustiano Carranza 1809 C Polanco	San Luis Potosi	SLP	Mexico	?	...	closed	none	cash	Japanese	13:00-23:00;	Mon;Tue;Wed;Thu;Fr	none	1.250000	1.375000	1.250000

Dataset a trabajar

Usuarios: 138 filas y 24 columnas

'userID', 'latitude', 'longitude', 'smoker', 'drink_level', 'dress_preference', 'ambience', 'transport', 'marital_status', 'hijos', 'birth_year', 'interest', 'personality', 'religion', 'activity', 'color', 'weight', 'budget', 'height', 'Rcuisine', 'Upayment', 'rating', 'food_rating', 'service_rating'

	userID	latitude	longitude	smoker	drink_level	dress_preference	ambience	transport	marital_status	hijos	...	activity	color	weight	budget	height	Rcuisine	Upayment	rating	food_rating	service_rating
0	U1001	22.139997	-100.978803	false	abstemious	informal	family	on foot	single	independent	...	student	black	69	medium	1.77	American	cash	1.111111	1.222222	1.222222
1	U1002	22.150087	-100.983325	false	abstemious	informal	family	public	single	independent	...	student	red	40	low	1.87	Mexican	cash	1.400000	1.400000	1.000000
2	U1003	22.119847	-100.946527	false	social drinker	formal	family	public	single	independent	...	student	blue	60	low	1.69	Mexican	cash	1.615385	1.692308	1.461538
3	U1004	18.867000	-99.183000	false	abstemious	informal	family	public	single	independent	...	professional	green	44	medium	1.53	Bakery	cash	1.875000	1.875000	1.750000
4	U1005	22.183477	-100.959891	false	abstemious	no preference	family	public	single	independent	...	student	black	65	medium	1.69	American	cash	1.333333	1.444444	1.000000
...
133	U1134	22.149654	-100.998610	false	casual drinker	no preference	family	public	single	independent	...	student	black	52	medium	1.65	Mexican	cash	1.437500	1.187500	1.187500
134	U1135	22.170396	-100.949936	false	casual drinker	informal	family	on foot	single	kids	...	student	purple	66	low	1.54	Organic-Healthy	cash	0.000000	0.000000	0.000000
135	U1136	22.149607	-100.997235	true	social drinker	no preference	friends	car owner	single	independent	...	student	black	50	low	1.60	Mexican	cash	1.600000	1.700000	1.800000
136	U1137	22.144803	-100.944623	false	social drinker	formal	family	public	single	independent	...	student	blue	72	low	1.78	Mexican	cash	1.857143	1.785714	1.928571
137	U1138	22.152884	-100.939663	false	social drinker	formal	friends	public	single	independent	...	student	blue	54	medium	1.55	Pizzeria	cash	1.666667	2.000000	1.333333

Dataset a trabajar

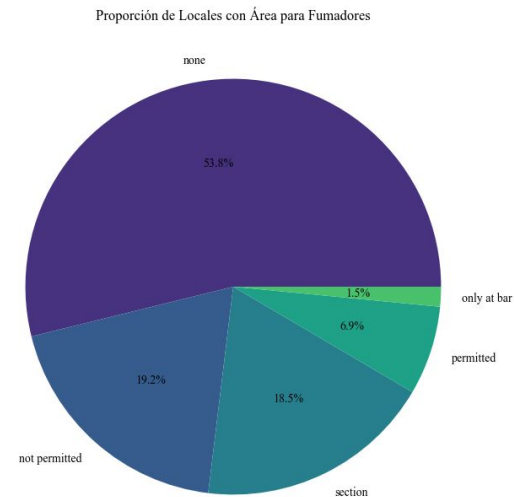
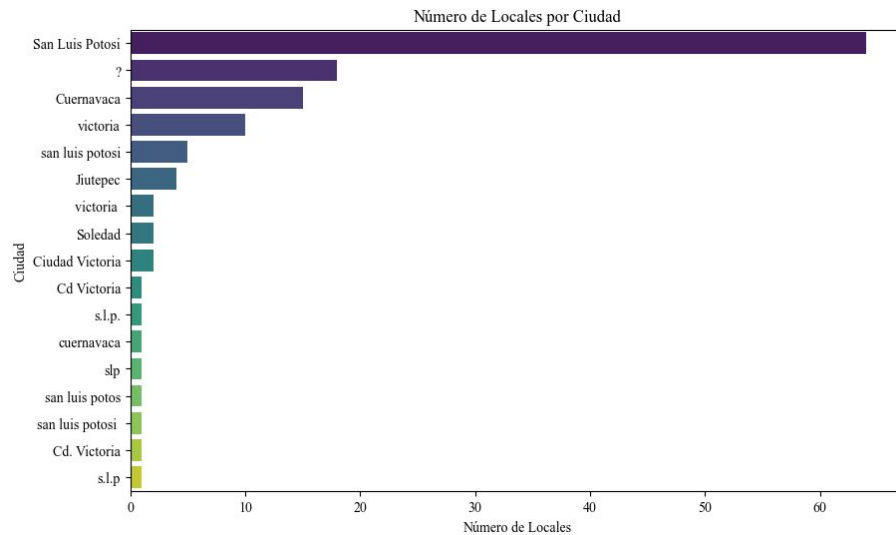
Ratings de usuarios: 1162 ratings en total

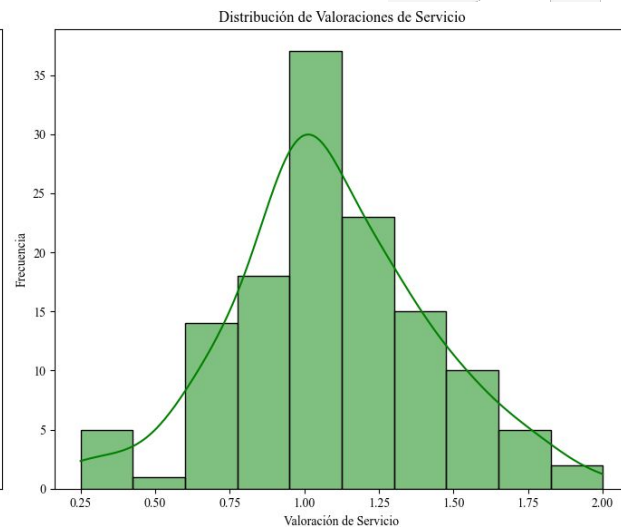
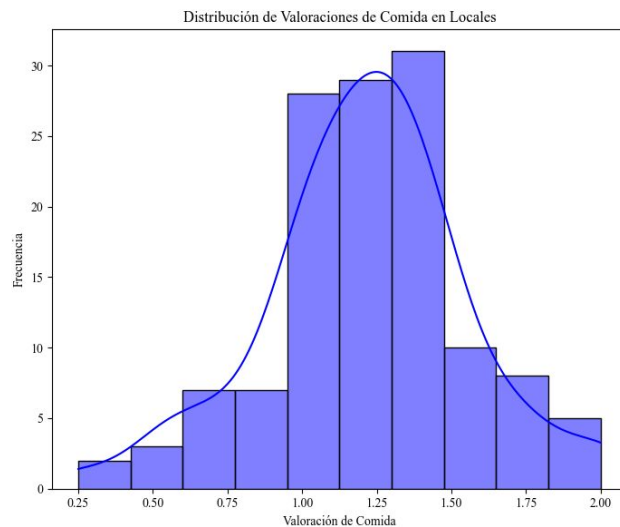
userID	placeID	rating	food_rating	service_rating
U1077	135085	2	2	2
U1077	135038	2	2	1
U1077	132825	2	2	2
U1077	135060	1	2	2
U1068	135104	1	1	2
U1068	132740	0	0	0
U1068	132663	1	1	1
U1068	132732	0	0	0
U1068	132630	1	1	1
U1067	132584	2	2	2
U1067	132733	1	1	1
U1067	132732	1	2	2
U1067	132630	1	0	1
U1067	135104	0	0	0

EDA

1/3/21

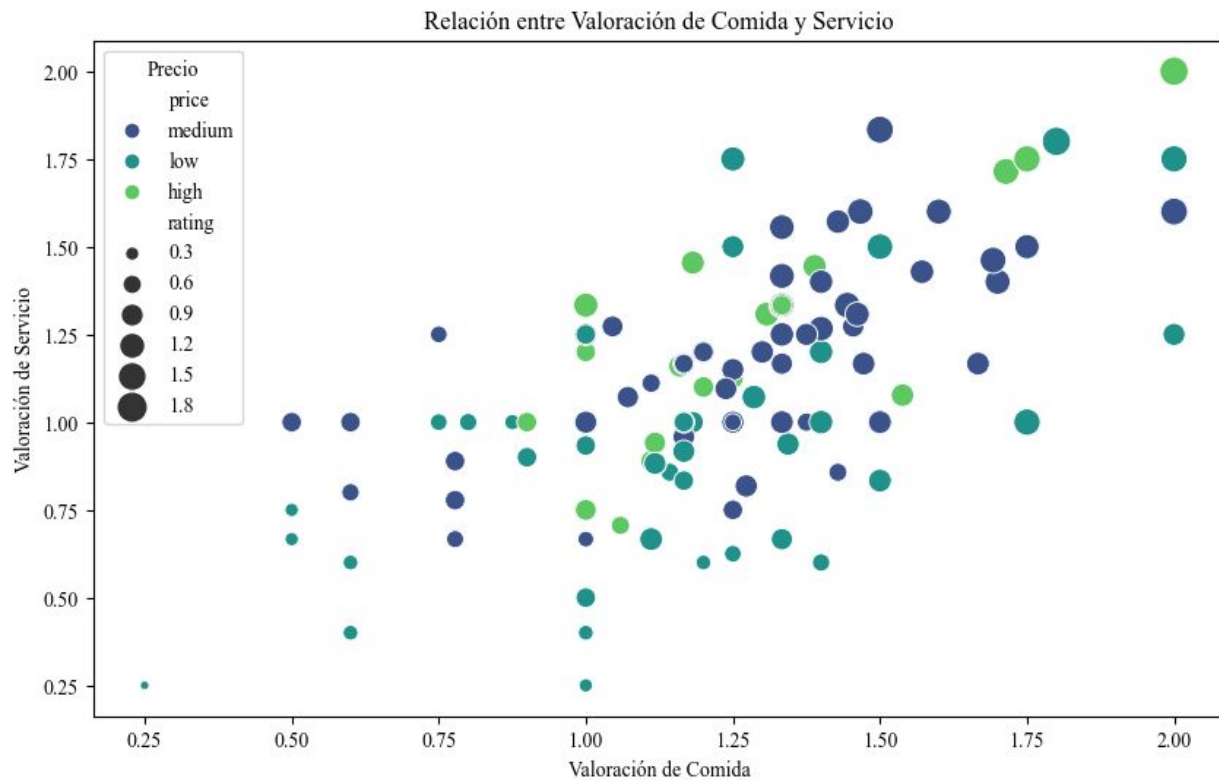
Locales





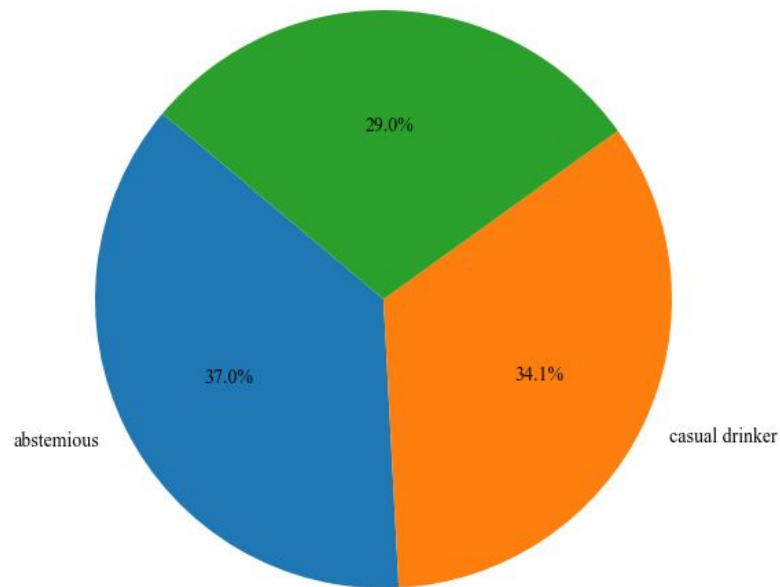
EDA

Locales

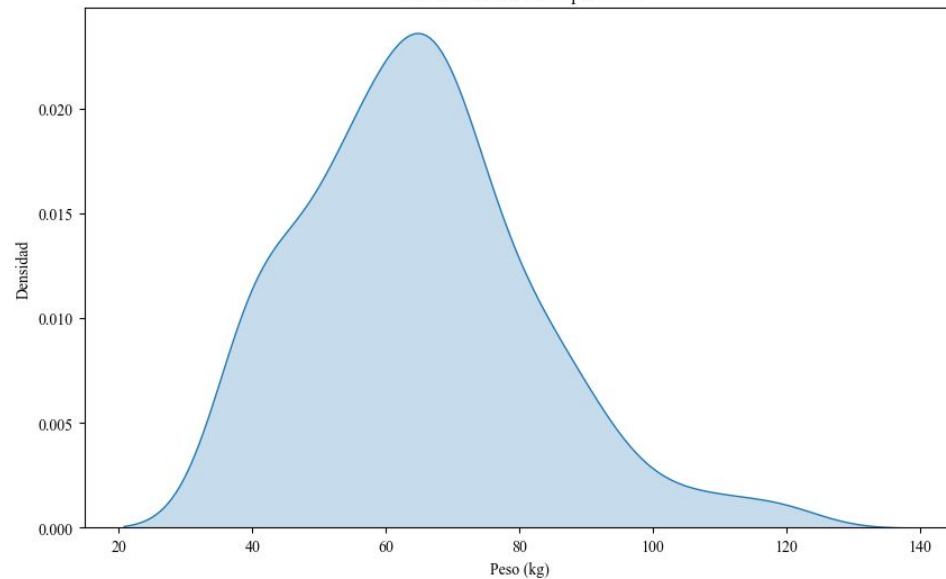


Usuarios

Distribución de Preferencia de Bebida
social drinker

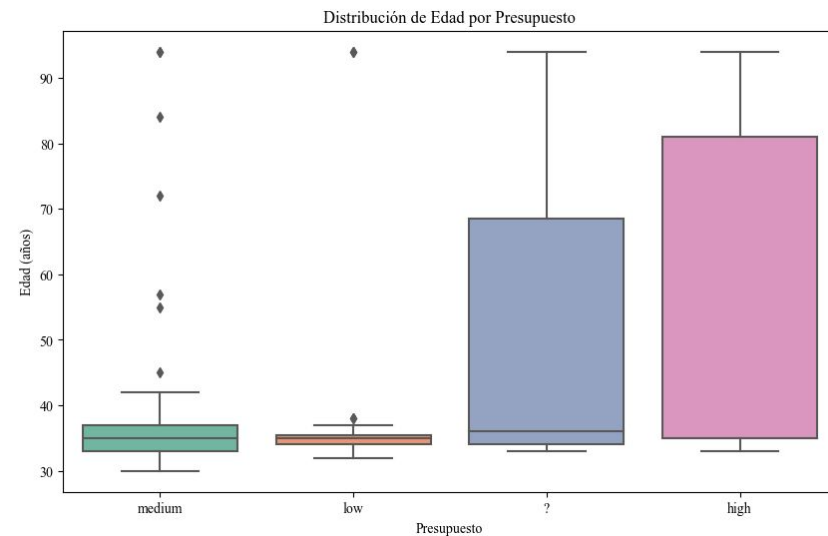
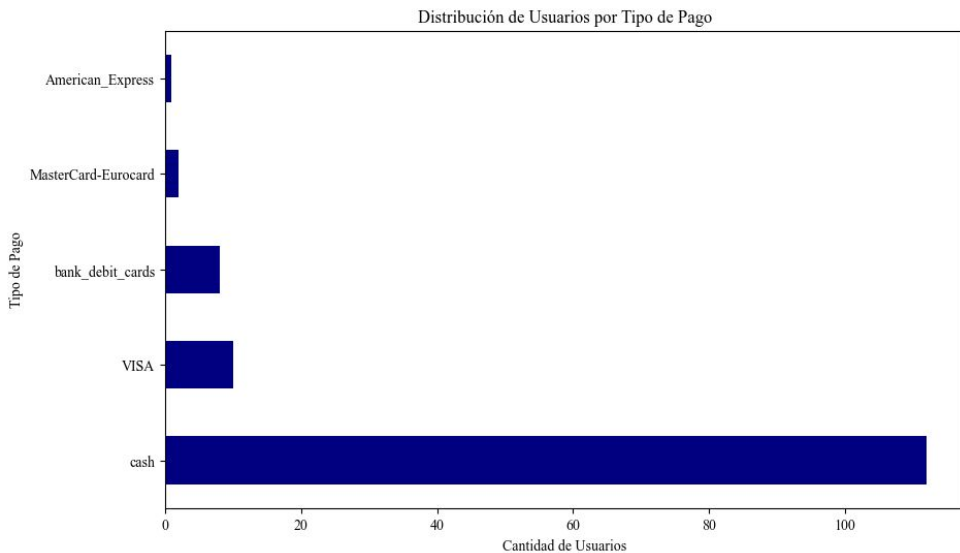


Densidad de Usuarios por Peso



EDA

Usuarios



SISTEMA DE RECOMENDACIÓN

Sistema de recomendación

Similitud coseno

- Cálculo de **similitud del coseno** para estimar la similitud entre contenidos
 - Utilización de *StandardScaler* para estandarizar características numéricas y *OneHotEncoder* para codificar variables categóricas.
 - Cálculo de la Matriz de Similitud entre los locales
 - Si dos vectores de datos son cercanos, el ángulo entre los mismos es pequeño, y la similitud coseno será alta

```
[[1.          0.29052311 0.26067155 ... 0.43389833 0.7050241  0.17912732]
 [0.29052311 1.          0.62682767 ... 0.13611863 0.49574583 0.57941364]
 [0.26067155 0.62682767 1.          ... 0.22696956 0.15812585 0.33536408]
 ...
 [0.43389833 0.13611863 0.22696956 ... 1.          0.51621685 0.15246621]
 [0.7050241  0.49574583 0.15812585 ... 0.51621685 1.          0.43144178]
 [0.17912732 0.57941364 0.33536408 ... 0.15246621 0.43144178 1.          ]]
```

Sistema de recomendación

Similitud coseno

- Función de recomendación
 - Toma el nombre de un local como entrada
 - Utiliza la matriz de similitud para encontrar locales similares
 - Devuelve el nombre, la dirección, la ciudad y los distintos ratings de los locales más similares

```
def recommend_locales_based_on_similarity(local_input, n=15):  
    try:  
        local_index = locales.loc[locales.name == local_input].index[0]  
    except IndexError:  
        return "Local no encontrado. Asegúrate de que el nombre sea correcto."  
  
    similarity_score = list(enumerate(similarity_matrix[local_index]))  
    similarity_score = sorted(similarity_score, key=lambda x: x[1], reverse=True)  
    similarity_score = similarity_score[1:n+1]  
    local_indices = [i[0] for i in similarity_score]  
  
    return locales[['name', 'address', 'city', 'rating', 'food_rating', 'service_rating']].iloc[local_indices]
```

Sistema de recomendación

Similitud coseno

Función de recomendación: ejemplo

```
recomendaciones = recommend_locales_based_on_similarity('carnitas_mata', n=15)
print(recomendaciones)
```

	name	address	city	rating	food_rating	service_rating
522	Hamburguesas Valle Dorado	Av. Coral	San Luis Potosi	0.800000	1.400000	0.600000
639	Hamburguesas saul	Av. Saan Luis entre moctezuma y salinas	San Luis Potosi	0.600000	0.600000	0.400000
57	Abondance Restaurante Bar	Industrias 908 Valle Dorado	San Luis Potosi	0.500000	0.500000	0.750000
645	Rincon Huasteco	?	?	0.916667	1.166667	0.916667
669	Cafe Chaires	?	San Luis Potosi	1.000000	1.000000	0.933333
705	Restaurant Orizatlan	Pascual M. Hernandez 240	San Luis Potosi	0.875000		

Sistema de recomendación

SR usando surprise

Para trabajar con surprise, se crea un objeto dataset con:

1. Los IDs de los usuarios
2. Los IDs de cada restaurante
3. El rating correspondiente (escala de 0 a 2)

Se entrena un modelo SVD (algoritmo de factorización de matrices) con cross validation para la matriz de puntuaciones. Una vez entrenado el modelo en todo el conjunto de datos, se pueden realizar predicciones para un ID específico de un cliente

```
from surprise import Dataset
from surprise import Reader

reader = Reader(rating_scale=(0, 2))
data = Dataset.load_from_df(df_ratings[['userID', 'placeID', 'rating']], reader)

from surprise import SVD
from surprise.model_selection import cross_validate

svd = SVD(verbose=True, n_epochs=10)
cross_validate(svd, data, measures=['RMSE', 'MAE'], cv=3, verbose=True)
```

RMSE (testset)	0.7360	0.7209	0.7010	0.7193	0.0143
MAE (testset)	0.6345	0.6118	0.6032	0.6165	0.0132
Fit time	0.00	0.00	0.00	0.00	0.00
Test time	0.00	0.00	0.00	0.00	0.00

Sistema de recomendación

Generar recomendaciones

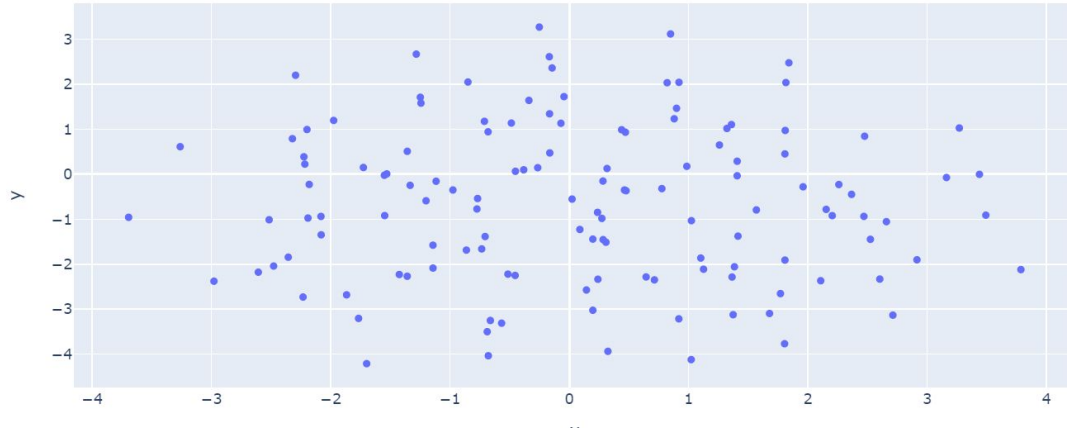
Se genera una función, *generate_recommendation*, que genera una recomendación de distintos restaurants dado un ID de usuario. La función itera a través de los nombres de los restaurants y predice las valoraciones del usuario para cada restaurant.

```
def generate_recommendation(user_id, model, metadata, thresh=4):  
  
    book_titles = list(metadata['title'].values)  
    random.shuffle(book_titles)  
  
    for book_title in book_titles:  
        rating = predict_review(user_id, book_title, model, metadata)  
        if rating >= thresh:  
            book_id = get_book_id(book_title, metadata)  
            return get_book_info(book_id, metadata)
```

Sistema de recomendación

Visualizando similitudes entre restaurantes utilizando t-SNE

Luego de aplicar el algoritmo SVD, se utiliza una técnica de reducción de la dimensionalidad. t-SNE es utilizado para representar a cada restaurante como un punto bidimensional.



Sistema de recomendación

Comparación de desempeño de algoritmos

	Algoritmo	RMSE CV	Precision	Recall	Rating Type
0	Random	0.998	0.402807	0.062074	rating
1	Baseline	0.508	0.201087	0.023027	rating
2	SVD	0.054	0.184783	0.016863	rating
3	KNN_msd	0.000	0.804348	0.190523	rating
4	Random	1.016	0.459129	0.069663	food_rating
5	Baseline	0.530	0.217391	0.022974	food_rating
6	SVD	0.055	0.206522	0.020409	food_rating
7	KNN_msd	0.000	0.833333	0.176832	food_rating
8	Random	1.017	0.368605	0.059156	service_rating
9	Baseline	0.534	0.150362	0.015390	service_rating
10	SVD	0.054	0.126812	0.011291	service_rating
11	KNN_msd	0.000	0.768116	0.188906	service_rating

Mejor modelo: Best params for KNN_msd: {'k': 20, 'sim_options': {'name': 'msd', 'user_based': False}}

Sistema de recomendación

Deploy



Streamlit

Sistema de Recomendación de Restaurantes

Ingrese su userID (por ejemplo, U1077):

U1075

Obtener Recomendaciones

	Nombre	Dirección	Calificación Estimada
0	Restaurante Tiberius	Munoz Sn Centro	1.6785
1	El angel Restaurante	Venustiano Carranza 1625 Jardin	1.6667
2	la Cochinita Pibil Restaurante Yucateco	Venustiano Carranza 2175 Jardin	1.625
3	Restaurante Bar El Gallinero	Pascual M. Hernandez 210 Centro	1.6154
4	Restaurant Bar Hacienda los Martinez	Santos Degollado 745 los Alamos	1.6047
5	Restaurante 75	Villa de Pozos 4497 Villa de Pozos	1.5823

<https://predictivoavanzado-omklnqj7jwrpu7f89akr7t.streamlit.app/>

Sistema de recomendación - mejora



Deploy

Place Recommendation System

Smoker:

false

Drink Level:

abstemious

Dress Preference:

informal

Ambience:

family

Transport:

on foot

Marital Status:

single

Hijos:

independent

Birth Year:

Top Recommendations

Restaurante Pueblo Bonito

Address: Mexico 2015 Providencia

Rating: 2

Restaurante Guerra

Address: 20 de Noviembre 1817 Tlaxcala

Rating: 2

Restaurant Oriental Express

Address: Tangamanga 7 Tangamanga

Rating: 2

Cafeteria y Restaurant El Pacifico

Address: Constitucion 200 Centro

Rating: 2

<https://predictivoavanzado.onrender.com/>



Flask
web development,
one drop at a time



Render

CONCLUSIONES

Conclusiones

- Al utilizar la similitud coseno, se obtiene robustez ante las distintas escalas de valores, y además brinda una interpretación intuitiva y fácil de calcular. Sin embargo, es sensible debido a la alta dimensionalidad de los datos.
- La escala de los ratings hace que en ocasiones sea difícil explicar lo que los números reflejan. Además, en ocasiones los restaurantes más recomendados para un usuario dado pueden tener la misma calificación estimada.
- Nuevos datos (ya sea de nuevos clientes o nuevas opiniones de los que ya están) podrían hacer más robusto al modelo.
- Más allá de las métricas, una buena de medir el sistema es con una evaluación basada en el juicio humano

Mejoras futuras

- Ampliar el rango de calificaciones posibles, a fines de obtener una mayor variabilidad en las evaluaciones.
- Ampliar las dimensiones de los datasets de usuarios y locales.
- Realizar una implementación de ambos deploys unificados, que integre utilidad tanto para usuarios nuevos como para ya registrados.
- Mejoras en desarrollo web
- Poder armar un solo indicador de rating que tenga en cuenta distintos aspectos de un restaurante.
 - Definir un criterio de feature importance para usuarios a fines de entender qué hace que dos clientes sean “similares” a la hora de elegir dónde o qué comer



Instituto Tecnológico
de Buenos Aires

¡Gracias!

Link del repositorio:

<https://github.com/bsoifer/PredictivoAvanzado>