

```
In [1]: import numpy as np
import pandas as pd
import math
import time
```

```
In [2]: df=pd.read_csv("df_Clean_TripAdvisor-2023-09-07_13_29_53.csv")
```

```
In [3]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9412 entries, 0 to 9411
Data columns (total 39 columns):
 #   Column                Non-Null Count  Dtype
--  --
 0   Unnamed: 0             9412 non-null   int64
 1   Nombre                9412 non-null   object
 2   Score                 9412 non-null   float64
 3   Excelente             9412 non-null   int64
 4   MuyBueno              9412 non-null   int64
 5   Normal                9412 non-null   int64
 6   Malo                  9412 non-null   int64
 7   Pesimo                9412 non-null   int64
 8   Num_Reviews           9412 non-null   int64
 9   Posicion              9397 non-null   float64
10   Rango_precio          9412 non-null   object
11   Tipo_Cocina           9412 non-null   object
12   Awards                9412 non-null   int64
13   Score_Comida          5598 non-null   float64
14   Score_Servicio         5578 non-null   float64
15   Calidad_Precio        5516 non-null   float64
16   Ambiente              1838 non-null   float64
17   Direccion             9412 non-null   object
18   Web_TripAdvisor       9412 non-null   object
19   Latina                9412 non-null   int64
20   Eur_Este              9412 non-null   int64
21   Africana              9412 non-null   int64
22   China                 9412 non-null   int64
23   Japonesa              9412 non-null   int64
24   Arabe                 9412 non-null   int64
25   Mexicana              9412 non-null   int64
26   Caribeña              9412 non-null   int64
27   Asiatica              9412 non-null   int64
28   America_Norte         9412 non-null   int64
29   America_Sur           9412 non-null   int64
30   Italiana              9412 non-null   int64
31   Española              9412 non-null   int64
32   Eur_Occidental        9412 non-null   int64
33   Asador                9412 non-null   int64
34   N/A                   9412 non-null   int64
35   India                 9412 non-null   int64
36   Num_Reviews_Spain     9412 non-null   int64
37   Latitud               6354 non-null   float64
38   Longitud              6354 non-null   float64
dtypes: float64(8), int64(26), object(5)
memory usage: 2.8+ MB
```

```
In [4]: #Hay que ordenar el df por las valoraciones para que ya se quede filtrado
#para ello tenemos una funcion
```

```
In [5]: #Hay que ajustar las columnas Score y Num_Reviews
*** Para calcular una métrica que capture la relación entre la puntuación y el
número de comentarios (reviews) de un restaurante, puedes utilizar el concepto
de "puntuación ponderada" donde le das más importancia al número de comentarios.
Una forma simple de hacerlo es multiplicar la puntuación por una función del número
de comentarios. Aquí tienes una posible fórmula:
Puntuación Ponderada = Puntuación * (1 + Factor * Log(Número de Comentarios + 1))
***
#Ejemplo de como funcionaria la formula
Factor=0.6 # se elige un factor para dar mayor o menor importancia a la puntuacion
Punt_poderada_list=[]
for x,y in zip(df["Score"][:810], df["Num_Reviews"][:810]):
    Punt_poderada=round(x*(1+Factor*math.log(y+1)),3)
    print(f'Score:{x},Num_Reviews:{y}, Punt_poderada:{Punt_poderada}')
    Punt_poderada_list.append(Punt_poderada)
# explicar con índice 0 e índice 3 Teoria vs Punt_Ponderada

Score:3.5,Num_Reviews:35, Punt_poderada:11.025
Score:4.5,Num_Reviews:15, Punt_poderada:11.625
Score:4.0,Num_Reviews:191, Punt_poderada:16.618
Score:5.0,Num_Reviews:25, Punt_poderada:14.774
Score:3.5,Num_Reviews:66, Punt_poderada:12.33
Score:2.0,Num_Reviews:1, Punt_poderada:2.832
Score:3.5,Num_Reviews:3, Punt_poderada:6.411
Score:4.5,Num_Reviews:384, Punt_poderada:19.345
Score:3.0,Num_Reviews:141, Punt_poderada:11.52
Score:4.0,Num_Reviews:89, Punt_poderada:14.8
```

```
In [6]: factor=0.6 # se elige un factor para dar mayor o menor importancia a la puntuacion
Punt_poderada_list=[]
for x,y in zip(df["Score"], df["Num_Reviews"]):
    Punt_poderada=round(x*(1+Factor*math.log(y+1)),3)
    Punt_poderada_list.append(Punt_poderada)
```

```
In [7]: #con el resultado de la formula anterior se crea una nueva columna que se llama valoracion
df["Valoracion"]=Punt_poderada_list
```

```
In [8]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9412 entries, 0 to 9411
Data columns (total 40 columns):
 #   Column                Non-Null Count  Dtype
--  --
 0   Unnamed: 0             9412 non-null   int64
 1   Nombre                9412 non-null   object
 2   Score                 9412 non-null   float64
 3   Excelente             9412 non-null   int64
 4   MuyBueno              9412 non-null   int64
 5   Normal                9412 non-null   int64
 6   Malo                  9412 non-null   int64
 7   Pesimo                9412 non-null   int64
 8   Num_Reviews           9412 non-null   int64
 9   Posicion              9397 non-null   float64
10   Rango_precio          9412 non-null   object
11   Tipo_Cocina           9412 non-null   object
12   Awards                9412 non-null   int64
13   Score_Comida          5598 non-null   float64
14   Score_Servicio         5578 non-null   float64
15   Calidad_Precio        5516 non-null   float64
16   Ambiente              1838 non-null   float64
17   Direccion             9412 non-null   object
18   Web_TripAdvisor       9412 non-null   object
19   Latina                9412 non-null   int64
20   Eur_Este              9412 non-null   int64
21   Africana              9412 non-null   int64
22   China                 9412 non-null   int64
23   Japonesa              9412 non-null   int64
24   Arabe                 9412 non-null   int64
25   Mexicana              9412 non-null   int64
26   Caribeña              9412 non-null   int64
27   Asiatica              9412 non-null   int64
28   America_Norte         9412 non-null   int64
29   America_Sur           9412 non-null   int64
30   Italiana              9412 non-null   int64
31   Española              9412 non-null   int64
32   Eur_Occidental        9412 non-null   int64
33   Asador                9412 non-null   int64
34   N/A                   9412 non-null   int64
35   India                 9412 non-null   int64
36   Num_Reviews_Spain     9412 non-null   int64
37   Latitud               6354 non-null   float64
38   Longitud              6354 non-null   float64
39   Valoracion            9412 non-null   float64
dtypes: float64(9), int64(26), object(5)
memory usage: 2.9+ MB
```

```
In [9]: #####
```

Recomendador por Tipo Cocina

```
In [10]: #codigo para saber si el usuario prefiere entrar a escoger el tipo de cocina
while True:
    try:
        recomienda_cocina = int(input("¿Tiene preferencias por algún tipo de cocina? (Si=1 / No=0): "))
        if recomienda_cocina != 0 and recomienda_cocina != 1:
            raise ValueError("Por favor, ingrese 0 o 1.")#evalua si alcanza otro valor diferente
        break # Solicita al usuario si la entrada es válida
    except ValueError:
        print("Entrada no válida. Por favor, ingrese 0 o 1.")
```

```
{Tiene preferencias por algún tipo de cocina? (Si=1 / No=0): 1
```

```
In [11]: # Define las opciones que el usuario puede calificar
if recomienda_cocina:
    opciones = ["Latina", "Asiatica", "Italiana", "Española",
                "America Norte", "Eur Este", "Eur Occidental", "Arabe",
                "Africana", "India"]

# Crea un diccionario para almacenar las calificaciones
calificaciones = {}

# Solicita al usuario que califique cada opción
print("Por favor, califique qué tipo generico de cocina le gusta(escala 1 a 10),No se puede repetir la calificacion")
print("Siendo 10: Me encanta y 1:No me apetece")
for opcion in opciones:
    calificacion = int(input(f"Califique '{opcion}' (1-10): "))
    if calificacion > 0 and calificacion <= 11:
        calificaciones[opcion] = calificacion/10
    else:
        print("La clasificacion es incorrecta")
        break

print("\nCalificaciones:", calificaciones)

else:
    print("Ya que no tiene preferencias por tipo de cocina le mostramos una lista ordenada por valoracion")
    num_rest= int(input("¿Cuántos restaurantes quiere en su lista(1-50)?"))
    df_ordenado = df.sort_values(by='Valoracion', ascending=False)
    df_sinpreferencia =df_ordenado[["Nombre", "Score", "Valoracion", "Tipo_Cocina"]].head(num_rest)#los 25
    print(df_sinpreferencia)
```

```
Por favor, califique qué tipo generico de cocina le gusta(escala 1 a 10),No se puede repetir la calificacion
Siendo 10: Me encanta y 1:No me apetece
Califique 'Latina' (1-10): 5
Califique 'Asiatica' (1-10): 6
Califique 'Italiana' (1-10): 3
Califique 'Española' (1-10): 4
Califique 'America Norte' (1-10): 5
Califique 'Eur Este' (1-10): 2
Califique 'Eur Occidental' (1-10): 5
Califique 'Arabe' (1-10): 4
Califique 'Africana' (1-10): 5
Califique 'India' (1-10): 3
Calificaciones: {'Latina': 0.5, 'Asiatica': 0.6, 'Italiana': 0.3, 'Española': 0.4, 'America Norte': 0.5, 'Eur Este': 0.2, 'Eur Occidental': 0.5, 'Arabe': 0.4, 'Africana': 0.5, 'India': 0.3}
```

```
In [12]: #Funcion ordena diccionarios
def ordena_diccionario(diccionario):
    # Ordenar el diccionario de mayor a menor
    diccionario_ordenado = dict(sorted(diccionario.items(), key=lambda item: item[1], reverse=True))

    return diccionario_ordenado
```

```
In [13]: # Mstrar el diccionario ordenado
calificaciones_ordenado=ordena_diccionario(calificaciones)
calificaciones_ordenado
```

```
Out[13]: {'Asiatica': 0.6,
'Latina': 0.5,
'America Norte': 0.5,
'Eur Occidental': 0.5,
'Africana': 0.5,
'Española': 0.4,
'Arabe': 0.4,
'Italiana': 0.3,
'India': 0.3,
'Eur Este': 0.2}
```

```
In [14]: #Funcion para puntuar los tipos de cocina segun deseo del usuario
def puntuacion_cocinas(lista):
    # Solicita al usuario que califique cada opción
    print(f"Por favor, califique las opciones de cocina: {lista} (escala 1 a 10)")
    print("Siendo 10: Me encanta y 1:No me apetece")

    calificaciones1={}
    for opcion in lista:
        calificacion = int(input(f"Califique '{opcion}' (1-10): "))
        if calificacion > 0 and calificacion <= 11:
            calificaciones1[opcion] = calificacion/10
        else:
            print("La clasificacion es incorrecta")
            break
    return calificaciones1

# Verifica si la clave deseada está entre las tres primeras posiciones
clave = ["Latinoamericana", "Asiatica"]
primeras_tres_claves = list(calificaciones_ordenado.keys())[:3]
latinoamerica = ["America Sur", "Asador", "Caribeña", "Latina", "Mexicana"]
Asiatica = ["Asiatica", "Japonesa", "Japonesa"]

for x in clave:
    if x in primeras_tres_claves:
        if x == "latinoamericana":
            print(f"En las calificaciones '{x}'")
            calif_latina=puntuacion_cocinas(Latinoamerica)
            calif_latina_ordenado=ordena_diccionario(calif_latina)
            lista_latin= list(calif_latina_ordenado.keys())[:2]
            #calif_latina_ordenado
            #print(calif_latina)

        elif x == "Asiatica":
            print(f"En las calificaciones '{x}'")
            calif_asiatca=puntuacion_cocinas(Asiatica)
            calif_asiatca_ordenado=ordena_diccionario(calif_asiatca)
            lista_asia= list(calif_asiatca_ordenado.keys())[:2]
            #calif_asiatca_ordenado
            #print(calif_asiatca)

    else:
        lista_pref= list(calificaciones_ordenado.keys())[:4]
        #print(f'La clave "{x}" no se encuentra entre las tres primeras posiciones.')
```

```
Calificaciones Asiatica
Por favor, califique las opciones de cocina: ['Asiatica', 'China', 'Japonesa'] (escala 1 a 10)
Siendo 10: Me encanta y 1:No me apetece
Califique 'Asiatica' (1-10): 2
Califique 'China' (1-10): 3
Califique 'Japonesa' (1-10): 4
```

```
In [16]: #calif_latina_ordenado=ordena_diccionario(calif_latina)
#calif_latina_ordenado
```

```
In [17]: #lista_latin= list(calif_latina_ordenado.keys())[:2]
#lista_latin
```

```
In [18]: #calif_asiatca_ordenado=ordena_diccionario(calif_asiatca)
#calif_asiatca_ordenado
```

```
In [19]: #lista_asia= list(calif_asiatca_ordenado.keys())[:2]
#lista_asia
```

```
In [20]: #tomo las 3 primeros de la lista
#lista_pref= list(calificaciones_ordenado.keys())[:4]
#lista_pref
```

```
In [21]: #Para organizar por orden de peso la lista y sublista de tipos de puntuacion
tipo_cocina_preferida_list=[]
for x in lista_pref:
    if x != "Latinoamericana" and x!="Asiatica":
        tipo_cocina_preferida_list.append(x)
    elif x == "Latinoamericana":
        tipo_cocina_preferida_list.extend(lista_latin)
    elif x == "Asiatica":
        tipo_cocina_preferida_list.extend(lista_asia)
print(tipo_cocina_preferida_list)

['Japonesa', 'China', 'Latina', 'America Norte', 'Eur Occidental']
```

```
In [24]: resultado=pd.DataFrame()
dicc_df={}
z= int(input(f"¿Cuántas recomendaciones quiere por tipo de cocina?(1-10)?"))#numero de recomendaciones por tipo de cocina
for x in tipo_cocina_preferida_list:
    for x in df.columns[19:4]:#se corresponde con las columnas tipo cocina
        if x == y:
            df_recomendador1 = df[df[x]==1]
            print(f"El tipo de cocina Seleccionada: {y}")
            df_ordenado = df_recomendador1.sort_values(by='Valoracion', ascending=False)
            print(df_ordenado.head(z))
            dicc_df[x]=df_ordenado.head(z)

#Cuántas recomendaciones quiere por tipo de cocina?(1-10)3
El tipo de cocina Seleccionada: Japonesa
    Unnamed: 0      Nombre  Score  Excelente  MuyBueno  Normal  Malo  Pesimo  Num_Reviews  Posicion  ...  Italiana  Española  Eur Occidental  Asador  N/A  India  Num_Reviews.Spain  Latitud  Longitud  Valoracion
1519      1519      Sakale      5.0      2      206      57      30      3309      630.0  ...      0      0      0      0      0      0      0      334      40.423291      -3.703706      29.314
2055      2055      Sibuya Urban Sushi Bar Madrid C/Orense      5.0      697      61      10      5      12      801      16.0  ...      0      0      0      0      0      0      0      785      40.614659      -4.039260      25.061
2931      2931      Yakitoro Reina      4.0      1      1      888      375      273      5530      630.0  ...      0      1      1      0      0      0      0      1538      40.713156      -3.994980      24.683
```

```
[3 rows x 40 columns]
El tipo de cocina Seleccionada: China
    Unnamed: 0      Nombre  Score  Excelente  MuyBueno  Normal  Malo  Pesimo  Num_Reviews  Posicion  ...  Italiana  Española  Eur Occidental  Asador  N/A  India  Num_Reviews.Spain  Latitud  Longitud  Valoracion
6833      6833      Restaurante Luxury      4.5      261      135      45      14      16      545      453.0  ...      0      0      0      0      0      0      0      471      40.430267      -3.704908      21.517
9162      9162      Royal Cantones      4.0      173      218      77      26      17      529      733.0  ...      0      0      0      0      0      0      0      511      40.383079      -3.702529      19.055
7601      7601      Asia Gallery      4.0      162      166      75      28      23      521      988.0  ...      0      0      0      0      0      0      0      454      NaN      NaN      19.018
```

```
[3 rows x 40 columns]
El tipo de cocina Seleccionada: Italiana
    Unnamed: 0      Nombre  Score  Excelente  MuyBueno  Normal  Malo  Pesimo  Num_Reviews  Posicion  ...  Italiana  Española  Eur Occidental  Asador  N/A  India  Num_Reviews.Spain  Latitud  Longitud  Valoracion
1519      1519      Sakale      5.0      2      206      57      30      3309      630.0  ...      0      0      0      0      0      0      0      334      40.423291      -3.703706      29.314
2055      2055      Sibuya Urban Sushi Bar Madrid C/Orense      5.0      697      61      10      5      12      801      16.0  ...      0      0      0      0      0      0      0      785      40.614659      -4.039260      25.061
2931      2931      Yakitoro Reina      4.0      1      1      888      375      273      5530      630.0  ...      0      1      1      0      0      0      0      1538      40.713156      -3.994980      24.683
```

```
[3 rows x 40 columns]
El tipo de cocina Seleccionada: Latina
    Unnamed: 0      Nombre  Score  Excelente  MuyBueno  Normal  Malo  Pesimo  Num_Reviews  Posicion  ...  Italiana  Española  Eur Occidental  Asador  N/A  India  Num_Reviews.Spain  Latitud  Longitud  Valoracion
3058      3058      Grama Lounge      5.0      2      211      39      16      29      4073      63.0  ...      0      0      0      0      0      0      0      297      40.415520      -3.701731      29.937
2691      2691      Mul El Placer De La Carne      4.5      1      226      41      13      22      2956      59.0  ...      0      0      0      0      1      0      0      303      40.419458      -3.703946      26.078
7332      7332      Santita Virgen de los Peligros      5.0      759      94      10      9      1060      61.0  ...      0      0      0      0      0      1      0      886      40.418829      -3.699498      25.901
```

```
[3 rows x 40 columns]
El tipo de cocina Seleccionada: America Norte
    Unnamed: 0      Nombre  Score  Excelente  MuyBueno  Normal  Malo  Pesimo  Num_Reviews  Posicion  ...  Italiana  Española  Eur Occidental  Asador  N/A  India  Num_Reviews.Spain  Latitud  Longitud  Valoracion
2258      2258      Steakburger Gran Via      5.0      327      55      2      211      39      429      40.415980      ...      0      0      0      0      0      0      0      297      40.415520      -3.701731      29.937
2996      2996      Steakburger Arenal      5.0      947      61      10      9      1060      61.0  ...      0      0      0      0      0      1      0      886      40.418829      -3.699498      25.901
2617      2617      Steakburger Gran Via      5.0      1      90      21      10      1060      61.0  ...      0      0      0      0      0      1      0      886      40.418829      -3.699498      25.901
```

```
[3 rows x 40 columns]
El tipo de cocina Seleccionada: Eur Occidental
    Unnamed: 0      Nombre  Score  Excelente  MuyBueno  Normal  Malo  Pesimo  Num_Reviews  Posicion  ...  Italiana  Española  Eur Occidental  Asador  N/A  India  Num_Reviews.Spain  Latitud  Longitud  Valoracion
496      496      Zenith Brunch & Cocktails      5.0      951      110      2      2273      73.0  ...      0      0      0      0      0      0      0      511      40.383079      -3.702529      19.055
2157      2157      Taberna Más Al Sur      4.5      1      372      41      13      22      2956      59.0  ...      0      0      0      0      1      0      0      303      40.419458      -3.703946      26.078
```

```
[3 rows x 40 columns]
El tipo de cocina Seleccionada: Eur Occidental
    Unnamed: 0      Nombre  Score  Excelente  MuyBueno  Normal  Malo  Pesimo  Num_Reviews  Posicion  ...  Italiana  Española  Eur Occidental  Asador  N/A  India  Num_Reviews.Spain  Latitud  Longitud  Valoracion
496      496      Zenith Brunch & Cocktails      5.0      951      110      2      2273      73.0  ...      0      0      0      0      0      0      0      511      40.383079      -3.702529      19.055
2157      2157      Taberna Más Al Sur      4.5      1      372      41      13      22      2956      59.0  ...      0      0      0      0      1      0      0      303      40.419458      -3.703946      26.078
```

```
[3 rows x 40 columns]
El tipo de cocina Seleccionada: Eur Occidental
    Unnamed: 0      Nombre  Score  Excelente  MuyBueno  Normal  Malo  Pesimo  Num_Reviews  Posicion  ...  Italiana  Española  Eur Occidental  Asador  N/A  India  Num_Reviews.Spain  Latitud  Longitud  Valoracion
496      496      Zenith Brunch & Cocktails      5.0      951      110      2      2273      73.0  ...      0      0      0      0      0      0      0      511      40.383079      -3.702529      19.055
2157      2157      Taberna Más Al Sur      4.5      1      372      41      13      22      2956      59.0  ...      0      0      0      0      1      0      0      303      40.419458      -3.703946      26.078
```

```
[3 rows x 40 columns]
El tipo de cocina Seleccionada: Eur Occidental
    Unnamed: 0      Nombre  Score  Excelente  MuyBueno  Normal  Malo  Pesimo  Num_Reviews  Posicion  ...  Italiana  Española  Eur Occidental  Asador  N/A  India  Num_Reviews.Spain  Latitud  Longitud  Valoracion
496      496      Zenith Brunch & Cocktails      5.0      951      110      2      2273      73.0  ...      0      0      0      0      0      0      0      511      40.383079      -3.702529      19.055
2157      2157      Taberna Más Al Sur      4.5      1      372      41      13      22      2956      59.0  ...      0      0      0      0      1      0      0      303      40.419458      -3.703946      26.078
```

```
[3 rows x 40 columns]
El tipo de cocina Seleccionada: Eur Occidental
    Unnamed: 0      Nombre  Score  Excelente  MuyBueno  Normal  Malo  Pesimo  Num_Reviews  Posicion  ...  Italiana  Española  Eur Occidental  Asador  N/A  India  Num_Reviews.Spain  Latitud  Longitud  Valoracion
496      496      Zenith Brunch & Cocktails      5.0      951      110      2      2273      73.0  ...      0      0      0      0      0      0      0      511      40.383079      -3.702529      19.055
2157      2157      Taberna Más Al Sur      4.5      1      372      41      13      22      2956      59.0  ...      0      0      0      0      1      0      0      303      40.419458      -3.703946      26.078
```

```
3 rows x 40 columns
```