

Python [conda env:DataScience]

Desde la comprensión hasta la preparación de los Datos.

Descripción del conjunto de Datos:

- Los datos fueron compilados por un investigador llamado Yong-Yeol Ahn, quien extrajo decenas de miles de recetas de comida (cocinas e ingredientes) de tres sitios web diferentes, a saber:
 - www.allrecipes.com
 - www.epicurious.com
 - www.menupan.com
- Todos los datos recopilados fueron agrupados en: ***recipes.csv***
- Nuestro conjunto de datos consta de diversas recetas y sus respectivos ingredientes.
- Cada fila representa una receta, y para cada receta, se documenta la zona geográfica correspondiente y si existen o no los ingredientes en la receta, comenzando con **almendras** (almond) y terminando con **zucchini** (zucchini).
- Queremos determinar cuáles son los ingredientes más usados en cada zona geográfica.
- Por tal razón se eliminó el nombre de cada receta, quedando solo la zona geográfica y los ingredientes.

Importación de los paquetes y bibliotecas necesarios:

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

# Deshabilitar advertencias
import warnings
warnings.filterwarnings('ignore')

pd.set_option('display.max_columns', None)

print('Todos los los paquetes importados!')
```

Todos los los paquetes importados!

```
In [2]: recetas = pd.read_csv("recipes.csv")
print("Datos leídos al DataFrame = 'recetas'!")
```

Datos leídos al DataFrame = 'recetas'!

Chequeo de las primeras filas.

```
In [3]: recetas.head()
```

```
Out[3]:
```

	country	almond	angelica	anise	anise_seed	apple	apple_brandy	apricot	arr
0	Vietnamese	No	No	No	No	No	No	No	
1	Vietnamese	No	No	No	No	No	No	No	
2	Vietnamese	No	No	No	No	No	No	No	
3	Vietnamese	No	No	No	No	No	No	No	
4	Vietnamese	No	No	No	No	No	No	No	

Obtengo las dimensiones del DataFrame.

```
In [4]: recetas.shape
```

```
Out[4]: (57691, 384)
```

Entonces nuestro conjunto de datos consta de 57.691 recetas. Cada fila representa una receta, y para cada receta, se documenta la cocina correspondiente y si existen o no 384 ingredientes en la receta, comenzando con almendras y terminando con Zucchini.

Preparación de Datos:

Prepararemos los datos para el modelado, esta etapa implica explorar más los datos y asegurarse de que estén en el formato correcto para el algoritmo de aprendizaje automático que seleccionaremos en la etapa del enfoque analítico.

- Vamos a chequear los Datos para ver si es necesario limpiarlos.

```
In [5]: totales = recetas["country"].value_counts()

for item, valor in totales.items():
    print(item, valor)
```

American 40150
Mexico 1754
Italian 1715
Italy 1461
Asian 1176
French 996
east_asian 951
Canada 774
korean 767
Mexican 622
western 450
Southern_SoulFood 346
India 324
Jewish 320
Spanish_Portuguese 291
Mediterranean 289
UK-and-Ireland 282
Indian 274
France 268
MiddleEastern 248
Central_SouthAmerican 241
Germany 237
Eastern-Europe 235
Chinese 226
Greek 225
English_Scottish 204
Caribbean 183
Thai 164
Scandinavia 158
EasternEuropean_Russian 146
Cajun_Creole 146
Moroccan 137
Japanese 136
China 130
Thailand 125
African 115
Southwestern 108
South-America 103
japanese 99
Scandinavian 92
chinese 86
Irish 86
Japan 85
Spain 75
italian 74
Vietnamese 65
North-African 60
German 52
Portugal 50
Philippines 43
Korea 32
Netherlands 32
Lebanon 31
Vietnam 30
Austria 21
Iran 21
Switzerland 20
Pakistan 19
Malaysia 18
asian 17

Turkey 16
South-African 16
mexico 14
West-African 13
Indonesia 12
Belgium 11
East-African 11
Israel 9
Bangladesh 4

Al observar los datos, podemos obtener las siguientes conclusiones:

- La columna de cocina está etiquetada como country (País), lo cual es inexacto.
- Los nombres de las cocinas no son consistentes ya que no todos comienzan con una primera letra mayúscula.
- Algunas cocinas se duplican como variación del nombre del país, como Vietnam y Vietnamese.
- Algunas cocinas tienen muy pocas recetas.

Solucionemos estos problemas.

- Corregimos el nombre de la columna country por 'cocina'.

```
In [6]: nombres_de_columnas = recetas.columns.values  
nombres_de_columnas[0] = "cocina"  
recetas.columns = nombres_de_columnas
```

- Cambiamos todos los nombres de la columna 'cocina' para que comiencen con mayúsculas.

```
In [7]: recetas["cocina"] = recetas["cocina"].str.capitalize()  
  
cocinas = recetas['cocina'].unique()  
  
for cocina in cocinas:  
    print(cocina)
```

Vietnamese
Indian
Spanish_portuguese
Jewish
French
Central_southamerican
Cajun_creole
Thai
Scandinavian
Greek
American
African
Middleeastern
Easterneuropean_russian
Italian
Irish
Mexican
Chinese
German
Mediterranean
Japanese
Moroccan
Southern_soulfood
English_scottish
Asian
Southwestern
Mexico
East_asian
Western
Korean
Canada
Turkey
Caribbean
Bangladesh
India
France
Italy
Israel
Korea
Iran
Eastern-europe
South-african
Uk-and-ireland
China
Belgium
Germany
South-america
Spain
Netherlands
Scandinavia
Philippines
Indonesia
East-african
Vietnam
Thailand
Switzerland
West-african
North-african
Pakistan
Portugal

Lebanon
Malaysia
Austria
Japan

Make the cuisine names consistent.

```
In [8]: recetas.loc[recetas["cocina"] == "American", "cocina"] = "North-america"
recetas.loc[recetas["cocina"] == "Austria", "cocina"] = "Austrian"
recetas.loc[recetas["cocina"] == "Belgium", "cocina"] = "Belgian"
recetas.loc[recetas["cocina"] == "China", "cocina"] = "Chinese"
recetas.loc[recetas["cocina"] == "Canada", "cocina"] = "North-america"
recetas.loc[recetas["cocina"] == "Netherlands", "cocina"] = "Dutch"
recetas.loc[recetas["cocina"] == "France", "cocina"] = "French"
recetas.loc[recetas["cocina"] == "Germany", "cocina"] = "German"
recetas.loc[recetas["cocina"] == "India", "cocina"] = "Indian"
recetas.loc[recetas["cocina"] == "Indonesia", "cocina"] = "Indonesian"
recetas.loc[recetas["cocina"] == "Iran", "cocina"] = "Iranian"
recetas.loc[recetas["cocina"] == "Italy", "cocina"] = "Italian"
recetas.loc[recetas["cocina"] == "Japan", "cocina"] = "Japanese"
recetas.loc[recetas["cocina"] == "Israel", "cocina"] = "Israeli"
recetas.loc[recetas["cocina"] == "Korea", "cocina"] = "Korean"
recetas.loc[recetas["cocina"] == "Lebanon", "cocina"] = "Lebanese"
recetas.loc[recetas["cocina"] == "Malaysia", "cocina"] = "Malaysian"
recetas.loc[recetas["cocina"] == "Mexico", "cocina"] = "Mexican"
recetas.loc[recetas["cocina"] == "Pakistan", "cocina"] = "Pakistani"
recetas.loc[recetas["cocina"] == "Philippines", "cocina"] = "Philippine"
recetas.loc[recetas["cocina"] == "Scandinavia", "cocina"] = "Scandinavian"
recetas.loc[recetas["cocina"] == "Spain", "cocina"] = "Spanish_portuguese"
recetas.loc[recetas["cocina"] == "Portugal", "cocina"] = "Spanish_portugu
recetas.loc[recetas["cocina"] == "Switzerland", "cocina"] = "Swiss"
recetas.loc[recetas["cocina"] == "Thailand", "cocina"] = "Thai"
recetas.loc[recetas["cocina"] == "Turkey", "cocina"] = "Turkish"
recetas.loc[recetas["cocina"] == "Vietnam", "cocina"] = "Vietnamese"
recetas.loc[recetas["cocina"] == "Uk-and-ireland", "cocina"] = "Uk-and-ir
recetas.loc[recetas["cocina"] == "Irish", "cocina"] = "Uk-and-irish"

recetas
```

Out[8]:

	cocina	almond	angelica	anise	anise_seed	apple	apple_brandy	apricot
0	Vietnamese	No	No	No	No	No	No	No
1	Vietnamese	No	No	No	No	No	No	No
2	Vietnamese	No	No	No	No	No	No	No
3	Vietnamese	No	No	No	No	No	No	No
4	Vietnamese	No	No	No	No	No	No	No
...
57686	Japanese	No	No	No	No	No	No	No
57687	Japanese	No	No	No	No	No	No	No
57688	Japanese	No	No	No	No	No	No	No
57689	Japanese	No	No	No	No	No	No	No
57690	Japanese	No	No	No	No	No	No	No

57691 rows × 384 columns

- Eliminamos las cocinas con <50 recetas.

```
In [9]: # Obtener una lista de recetas para conservar.
conteo_de_recetas = recetas["cocina"].value_counts()
indices_de_cocinas = conteo_de_recetas > 50

cocinas_para_conservar = list(np.array(conteo_de_recetas.index.values)[np
```

```
In [10]: filas_antes = recetas.shape[0] # Numero de filas original del DataFrame.
print(f"Numero de filas del DataFrame original: {filas_antes}.")

recetas = recetas.loc[recetas['cocina'].isin(cocinas_para_conservar)] # F

filas_despues = recetas.shape[0] # Número de filas del DataFrame procesad
print(f"Numero de filas del DataFrame procesado: {filas_despues}.")

print(f"{filas_antes - filas_despues} filas removidas!")
```

Numero de filas del DataFrame original: 57691.
 Numero de filas del DataFrame procesado: 57394.
 297 filas removidas!

- Codificamos todos los Yes en 1 y los No en 0

```
In [11]: recetas = recetas.replace(to_replace = "Yes", value = 1)
recetas = recetas.replace(to_replace = "No", value = 0)
```

```
In [12]: recetas.head()
```

```
Out[12]:
```

	cocina	almond	angelica	anise	anise_seed	apple	apple_brandy	apricot	ari
0	Vietnamese	0	0	0	0	0	0	0	
1	Vietnamese	0	0	0	0	0	0	0	
2	Vietnamese	0	0	0	0	0	0	0	
3	Vietnamese	0	0	0	0	0	0	0	
4	Vietnamese	0	0	0	0	0	0	0	

Analizamos los datos un poco más para conocerlos mejor y anotar cualquier observación preliminar interesante.

- Chequearemos las recetas que contengan **rice**, **soy_sauce**, **wasabi**, **seaweed** y **carrot** y analizaremos los resultados.

```
In [13]: chequeo_de_recetas = recetas.loc[(recetas["rice"] == 1) & (recetas["soy_sauce"] == 1) & (recetas["wasabi"] == 1) & (recetas["seaweed"] == 1) & (recetas["carrot"] == 1)]

chequeo_de_recetas
```

```
Out[13]:
```

	cocina	almond	angelica	anise	anise_seed	apple	apple_brandy	apricot	ari
11321	Japanese	0	0	0	0	0	0	0	
11361	Japanese	0	0	0	0	0	0	0	
12171	Asian	0	0	0	0	0	0	0	
12385	Asian	0	0	0	0	0	0	0	
13159	Asian	0	0	0	0	0	0	0	
13586	Japanese	0	0	0	0	0	0	0	
14495	East_asian	0	0	0	0	0	0	0	

- Basados en los resultados obtenidos, las recetas con: **rice**, **soy_sauce**, **wasabi**, **seaweed** y **carrot** pertenecen a la Región de Japón, Asia y Este Asiático.

Contamos los ingredientes en todas las recetas.

```
In [14]: # Sumamos cada columna.
ingrediente_presente = recetas.iloc[:, 1:].sum(axis=0)
```

```
In [15]: # Definimos cada columna como una serie de pandas.
ingrediente = pd.Series(ingrediente_presente.index.values, index = np.arange(len(ingrediente_presente)))
recuento = pd.Series(list(ingrediente_presente), index = np.arange(len(ingrediente_presente)))

# Creamos un DataFrame
ingredientes_df = pd.DataFrame(dict(ingrediente = ingrediente, recuento = recuento))
ingredientes_df = ingredientes_df[["ingrediente", "recuento"]]
print(ingredientes_df.to_string())
```


	ingrediente	recuento
0	almond	2306
1	angelica	1
2	anise	223
3	anise_seed	87
4	apple	2420
5	apple_brandy	37
6	apricot	620
7	armagnac	11
8	artemisia	13
9	artichoke	391
10	asparagus	460
11	avocado	660
12	bacon	2169
13	baked_potato	9
14	balm	3
15	banana	989
16	barley	266
17	bartlett_pear	23
18	basil	3842
19	bay	1463
20	bean	1992
21	beech	1
22	beef	4902
23	beef_broth	845
24	beef_liver	10
25	beer	307
26	beet	233
27	bell_pepper	5979
28	bergamot	7
29	berry	183
30	bitter_orange	85
31	black_bean	494
32	black_currant	11
33	black_mustard_seed_oil	30
34	black_pepper	9825
35	black_raspberry	8
36	black_sesame_seed	26
37	black_tea	44
38	blackberry	170
39	blackberry_brandy	4
40	blue_cheese	396
41	blueberry	466
42	bone_oil	50
43	bourbon_whiskey	156
44	brandy	395
45	brassica	114
46	bread	4571
47	broccoli	929
48	brown_rice	346
49	brussels_sprout	92
50	buckwheat	90
51	butter	20715
52	buttermilk	1634
53	cabbage	1026
54	cabernet_sauvignon_wine	17
55	cacao	35
56	camembert_cheese	12
57	cane_molasses	7741
58	caraway	233

59	cardamom	352
60	carnation	3
61	carob	7
62	carrot	3689
63	cashew	208
64	cassava	19
65	catfish	71
66	cauliflower	332
67	caviar	28
68	cayenne	8253
69	celery	3625
70	celery_oil	1005
71	cereal	204
72	chamomile	3
73	champagne_wine	100
74	chayote	27
75	cheddar_cheese	3027
76	cheese	3278
77	cherry	1082
78	cherry_brandy	32
79	chervil	52
80	chicken	5436
81	chicken_broth	3603
82	chicken_liver	52
83	chickpea	402
84	chicory	156
85	chinese_cabbage	166
86	chive	1333
87	cider	1132
88	cilantro	2473
89	cinnamon	5593
90	citrus	167
91	citrus_peel	4
92	clam	476
93	clove	10
94	cocoa	4798
95	coconut	1801
96	coconut_oil	17
97	cod	180
98	coffee	719
99	cognac	67
100	concord_grape	12
101	condiment	9
102	coriander	1647
103	corn	4828
104	corn_flake	225
105	corn_grit	163
106	cottage_cheese	347
107	crab	574
108	cranberry	920
109	cream	10170
110	cream_cheese	2840
111	cucumber	1895
112	cumin	3274
113	cured_pork	315
114	currant	241
115	date	377
116	dill	1105
117	durian	0
118	eel	20

119	egg	21022
120	egg_noodle	317
121	elderberry	5
122	emmental_cheese	1
123	endive	115
124	enokidake	106
125	fennel	912
126	fenugreek	924
127	feta_cheese	623
128	fig	139
129	fish	2110
130	flower	32
131	frankfurter	37
132	fruit	479
133	galanga	49
134	gardenia	9
135	garlic	17351
136	gelatin	1417
137	geranium	1
138	gin	68
139	ginger	4358
140	goat_cheese	260
141	grape	346
142	grape_brandy	8
143	grape_juice	824
144	grapefruit	121
145	green_bell_pepper	2582
146	green_tea	35
147	gruyere_cheese	45
148	guava	13
149	haddock	31
150	ham	1300
151	hazelnut	284
152	herring	10
153	holy_basil	3
154	honey	2551
155	hop	3
156	horseradish	396
157	huckleberry	10
158	jamaican_rum	1
159	japanese_plum	13
160	jasmine	8
161	jasmine_tea	2
162	juniper_berry	33
163	kaffir_lime	1
164	kale	100
165	katsuobushi	63
166	kelp	179
167	kidney_bean	442
168	kiwi	109
169	kohlrabi	6
170	kumquat	33
171	lamb	482
172	lard	3051
173	laurel	2
174	lavender	62
175	leaf	9
176	leek	422
177	lemon	3043
178	lemon_juice	5065

179	lemon_peel	729
180	lemongrass	217
181	lentil	247
182	lettuce	1206
183	licorice	21
184	lilac_flower_oil	1
185	lima_bean	149
186	lime	1160
187	lime_juice	1618
188	lime_peel_oil	108
189	lingonberry	9
190	litchi	12
191	liver	42
192	lobster	131
193	long_pepper	2
194	lovage	142
195	macadamia_nut	102
196	macaroni	3115
197	mace	117
198	mackerel	44
199	malt	37
200	mandarin	279
201	mandarin_peel	15
202	mango	418
203	maple_syrup	478
204	marjoram	527
205	mate	1
206	matsutake	57
207	meat	987
208	melon	163
209	milk	12869
210	milk_fat	959
211	mint	1012
212	mozzarella_cheese	1288
213	mung_bean	24
214	munster_cheese	27
215	muscat_grape	1
216	mushroom	3370
217	mussel	168
218	mustard	4119
219	mutton	3
220	nectarine	51
221	nira	67
222	nut	1255
223	nutmeg	2506
224	oat	1265
225	oatmeal	61
226	octopus	45
227	okra	102
228	olive	1798
229	olive_oil	9874
230	onion	18078
231	orange	1724
232	orange_flower	17
233	orange_juice	1726
234	orange_peel	596
235	oregano	3180
236	ouzo	9
237	oyster	406
238	palm	46

239	papaya	57
240	parmesan_cheese	3173
241	parsley	5550
242	parsnip	139
243	passion_fruit	20
244	pea	1180
245	peach	531
246	peanut	509
247	peanut_butter	1014
248	peanut_oil	308
249	pear	484
250	pear_brandy	11
251	pecan	2176
252	pelargonium	1
253	pepper	9230
254	peppermint	142
255	peppermint_oil	8
256	pimenta	2
257	pimento	270
258	pineapple	1638
259	pistachio	219
260	plum	288
261	popcorn	97
262	porcini	106
263	pork	2056
264	pork_liver	5
265	pork_sausage	1369
266	port_wine	49
267	potato	3528
268	potato_chip	65
269	prawn	24
270	prickly_pear	20
271	provolone_cheese	168
272	pumpkin	803
273	quince	29
274	radish	525
275	raisin	1889
276	rapeseed	3
277	raspberry	784
278	raw_beef	2
279	red_algae	2
280	red_bean	33
281	red_kidney_bean	59
282	red_wine	1395
283	rhubarb	169
284	rice	3856
285	roasted_almond	3
286	roasted_beef	227
287	roasted_hazelnut	1
288	roasted_meat	15
289	roasted_nut	1
290	roasted_peanut	202
291	roasted_pecan	1
292	roasted_pork	124
293	roasted_sesame_seed	593
294	romano_cheese	275
295	root	101
296	roquefort_cheese	23
297	rose	56
298	rosemary	1892

299	rum	599
300	rutabaga	34
301	rye_bread	92
302	rye_flour	131
303	saffron	234
304	sage	904
305	sake	680
306	salmon	451
307	salmon_roe	15
308	sassafras	18
309	sauerkraut	185
310	savory	128
311	scallion	4782
312	scallop	300
313	sea_algae	4
314	seaweed	215
315	seed	1340
316	sesame_oil	1693
317	sesame_seed	778
318	shallot	1304
319	sheep_cheese	2
320	shellfish	27
321	sherry	706
322	shiitake	595
323	shrimp	1679
324	smoke	463
325	smoked_fish	6
326	smoked_salmon	100
327	smoked_sausage	268
328	sour_cherry	50
329	sour_milk	46
330	soy_sauce	3799
331	soybean	1195
332	soybean_oil	2
333	spearmint	6
334	squash	572
335	squid	238
336	star_anise	131
337	starch	2731
338	strawberry	1080
339	strawberry_jam	1
340	strawberry_juice	2
341	sturgeon_caviar	1
342	sumac	11
343	sunflower_oil	8
344	sweet_potato	529
345	swiss_cheese	519
346	tabasco_pepper	976
347	tamarind	1672
348	tangerine	52
349	tarragon	478
350	tea	108
351	tequila	142
352	thai_pepper	136
353	thyme	3043
354	tomato	9920
355	tomato_juice	176
356	truffle	52
357	tuna	463
358	turkey	900

359	turmeric	1290
360	turnip	188
361	vanilla	9009
362	veal	197
363	vegetable	1703
364	vegetable_oil	11103
365	vinegar	8060
366	violet	5
367	walnut	2728
368	wasabi	135
369	watercress	150
370	watermelon	110
371	wheat	20775
372	wheat_bread	82
373	whiskey	148
374	white_bread	370
375	white_wine	2205
376	whole_grain_wheat_flour	731
377	wine	1026
378	wood	33
379	yam	85
380	yeast	3385
381	yogurt	1033
382	zucchini	1102

- Ahora tenemos un marco de datos de ingredientes y sus recuentos totales en todas las recetas.
- Ordenemos este marco de datos en orden descendente.

```
In [16]: ingredientes_df.sort_values(["recuento"], ascending=False, inplace=True)
ingredientes_df.reset_index(inplace=True, drop=True)

print(ingredientes_df)
```

	ingrediente	recuento
0	egg	21022
1	wheat	20775
2	butter	20715
3	onion	18078
4	garlic	17351
..
378	strawberry_jam	1
379	sturgeon_caviar	1
380	kaffir_lime	1
381	beech	1
382	durian	0

[383 rows x 2 columns]

- Los 5 ingredientes que más se usan en estas recetas son:
 - Huevo: 21114 veces
 - Trigo: 20896 veces
 - Mantequilla: 20814 veces
 - Cebollas: 18205 veces
 - Ajos: 17465 veces

En la tabla anterior hay aproximadamente 40.000 recetas pertenecientes a North-america en nuestro conjunto de datos, lo que significa que los datos están sesgados hacia los ingredientes de esa region.

Por lo tanto, haremos un resumen más objetivo de los ingredientes observando los ingredientes por cocina.

Creemos un *perfil* para cada cocina.

Intentamos descubrir qué ingredientes suelen utilizar los Chinos y qué es la comida **caribeña**, por ejemplo.

```
In [17]: cocinas = recetas.groupby("cocina").mean()  
cocinas.head()
```

```
Out[17]:
```

	almond	angelica	anise	anise_seed	apple	apple_brar
cocina						
African	0.156522	0.0	0.000000	0.000000	0.034783	
Asian	0.007544	0.0	0.000838	0.002515	0.012573	
Cajun_creole	0.000000	0.0	0.000000	0.000000	0.006849	
Caribbean	0.016393	0.0	0.010929	0.000000	0.010929	
Central_southamerican	0.053942	0.0	0.000000	0.020747	0.000000	

Creamos un marco de datos donde cada fila es una cocina y cada columna (excepto la primera columna) es un ingrediente, y los valores de las filas representan el porcentaje de cada ingrediente en la cocina correspondiente.

Por ejemplo:

- La *almendra* (almond) está presente en el 15,65% de todas las recetas **Africanas** (Africans).
- El *ajo* (garlic) está presente en el 56,84% de todas las recetas de **Sudamérica Central** (Central_southamerican).

Vamos a imprimir el perfil de cada cocina mostrando los cuatro ingredientes principales de cada una.

```
In [28]: numero_de_ingredientes = 4 # número de ingredientes principales para impr  
  
# Función para imprimir los ingredientes principales de cada cocina  
def imprimir_ingredientes_mas_usados(row):  
    print(f'{row.name.upper()}:')  
    fila_ordenada = row.sort_values(ascending=False)*100  
    ingredientes_mas_usados = list(fila_ordenada.index.values)[0:numero_d  
    fila_ordenada = list(fila_ordenada)[0:numero_de_ingredientes]  
  
    for indice, ingrediente in enumerate(ingredientes_mas_usados):  
        print(f'{ingrediente}: {fila_ordenada[indice]:.0f}%')  
    print("\n")
```



```
# Aplico función al marco de datos de cocinas  
crear_perfiles_de_cocina = cocinas.apply(imprimir_ingredientes_mas_usados
```

AFRICAN:

onion: 53%
olive_oil: 52%
garlic: 50%
cumin: 43%

ASIAN:

soy_sauce: 50%
ginger: 49%
garlic: 48%
rice: 41%

CAJUN_CREOLE:

onion: 70%
cayenne: 56%
garlic: 49%
butter: 36%

CARIBBEAN:

onion: 51%
garlic: 51%
vegetable_oil: 31%
black_pepper: 31%

CENTRAL_SOUTHAMERICAN:

garlic: 57%
onion: 54%
cayenne: 52%
tomato: 41%

CHINESE:

soy_sauce: 69%
ginger: 53%
garlic: 53%
scallion: 48%

EAST_ASIAN:

garlic: 55%
soy_sauce: 50%
scallion: 50%
cayenne: 48%

EASTERN-EUROPE:

wheat: 53%
egg: 52%
butter: 48%
onion: 45%

EASTERNEUROPEAN_RUSSIAN:

butter: 60%
egg: 51%
wheat: 49%

onion: 38%

ENGLISH_SCOTTISH:

butter: 67%

wheat: 62%

egg: 53%

cream: 41%

FRENCH:

butter: 50%

egg: 44%

wheat: 37%

olive_oil: 28%

GERMAN:

wheat: 65%

egg: 61%

butter: 47%

onion: 35%

GREEK:

olive_oil: 76%

garlic: 44%

onion: 36%

lemon_juice: 34%

INDIAN:

cumin: 60%

turmeric: 51%

onion: 50%

coriander: 48%

ITALIAN:

olive_oil: 61%

garlic: 53%

tomato: 39%

onion: 33%

JAPANESE:

soy_sauce: 57%

rice: 44%

vinegar: 37%

vegetable_oil: 35%

JEWISH:

egg: 59%

wheat: 49%

butter: 31%

onion: 30%

KOREAN:

garlic: 59%
scallion: 52%
cayenne: 52%
soy_sauce: 49%

MEDITERRANEAN:
olive_oil: 80%
garlic: 51%
onion: 39%
tomato: 35%

MEXICAN:
cayenne: 74%
onion: 68%
garlic: 62%
tomato: 59%

MIDDLEEASTERN:
olive_oil: 60%
garlic: 47%
wheat: 38%
lemon_juice: 36%

MOROCCAN:
olive_oil: 73%
cumin: 55%
onion: 50%
garlic: 46%

NORTH-AFRICAN:
onion: 55%
olive_oil: 50%
cumin: 48%
garlic: 47%

NORTH-AMERICA:
butter: 41%
egg: 40%
wheat: 40%
onion: 29%

SCANDINAVIAN:
butter: 64%
wheat: 58%
egg: 53%
cream: 29%

SOUTH-AMERICA:
onion: 43%
garlic: 37%
egg: 35%
milk: 31%

SOUTHERN_SOULFOOD:
butter: 58%
wheat: 49%
egg: 42%
corn: 30%

SOUTHWESTERN:
cayenne: 81%
garlic: 62%
onion: 61%
cilantro: 52%

SPANISH_PORTUGUESE:
olive_oil: 58%
garlic: 54%
onion: 47%
bell_pepper: 35%

THAI:
garlic: 60%
fish: 53%
cayenne: 47%
cilantro: 42%

UK-AND-IRISH:
butter: 60%
wheat: 58%
egg: 48%
milk: 33%

VIETNAMESE:
fish: 74%
garlic: 73%
rice: 49%
cilantro: 43%

WESTERN:
egg: 51%
wheat: 46%
butter: 46%
black_pepper: 36%

Registro de cambios

Fecha (DD/MM/AAAA)	Versión	Descripción del cambio
--------------------	---------	------------------------

Fecha (DD/MM/AAAA)	Versión	Descripción del cambio
15/05/2023	1.0	Creación del Notebook
23/09/2023	1.1	Se agrega el Función: imprimir_ingredientes_mas_usados

In []: