

Лабораторна робота №4

Тематичне дослідження - сприятливі слова з відгуків про ресторани

In [1]:

```
# import the required libraries here
import pandas as pd
```

Найпопулярніша категорія ресторанів

Сподіваємось, ви подивилися словник даних тут (https://www.yelp.com/dataset/documentation/main) і дійшли висновку, що файл business.json найбільше підходить для отримання відповіді на питання про те, що є найпопулярнішою категорією. Ми завантажуюмо дані з файлу. Якщо файл даних знаходиться у папці JSON поряд з цими зоштами, то доступ буде за замовчуванням. Після цього необхідно виконати кілька типових швидких оглядів.

In [2]:

```
# For convenience here, we are accessing the data in the working directory that contains
# Normal best practise is to keep your data separate, but this keeps things simple.
business = pd.read_json('JSON/business.json', lines=True)
```

In [3]:

```
business.shape
```

Out[3]: (192609, 14)

In [4]:

```
business.head()
```

	business_id	name	address	city	state	postal_code	latitude	longitude
0	1SWheh84yJXfytovlXLOAQ	Arizona Biltmore Golf Club	2818 E Camino Acequia Drive	Phoenix	AZ	85016	33.522143	-112.018481
1	QXAEGFB4oINsVuTFxEYKFQ	Emerald Chinese Restaurant	30 Eglinton Avenue W	Mississauga	ON	L5R 3E7	43.605499	-79.652289
2	gnKJwL_1w79qoiV3lC_xQQ	Musashi Japanese Restaurant	10110 Johnston Rd, Ste 15	Charlotte	NC	28210	35.092564	-80.859132
3	xvX2CttrVhyG2z1dFg_0xw	Farmers Insurance - Paul Lorenz	15655 W Roosevelt St, Ste 237	Goodyear	AZ	85338	33.455613	-112.395596
4	HhyxOkGAM07SRYtiQ4wMFQ	Queen City Plumbing	4209 Stuart Andrew Blvd, Ste F	Charlotte	NC	28217	35.190012	-80.887223

In [5]:

```
business.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 192609 entries, 0 to 192608
Data columns (total 14 columns):
#   Column                Non-Null Count  Dtype
---  -
0   business_id           192609 non-null object
1   name                  192609 non-null object
2   address               192609 non-null object
3   city                  192609 non-null object
4   state                 192609 non-null object
5   postal_code           192609 non-null object
6   latitude              192609 non-null float64
7   longitude             192609 non-null float64
8   stars                 192609 non-null float64
9   review_count          192609 non-null int64
10  is_open               192609 non-null int64
11  attributes             163773 non-null object
12  categories            192127 non-null object
13  hours                 147779 non-null object
dtypes: float64(3), int64(2), object(9)
memory usage: 20.6+ MB
```

У цьому наборі даних не всі стовпці нам корисні. Ми майже впевнені, що нас не цікавлять конкретні адреси (address), атрибути (attributes), години роботи (hours), чи все ще відкритий заклад (is_open), широта (latitude) та довгота (longitude) та назва ресторану (name). Ми також вирішуємо, що не будемо проводити аналіз на рівні поштового індексу (postal_code). Тому ці колонки не розглядаємо. Тому наступною дією повинно бути створення списку цих стовпців і видалення їх з DataFrame. Запам'ятайте вісь (axis) та замініть аргументи! Також зауважте, що існує менше ненульових категорій, ніж є business_id. Ми повернемося до цього питання пізніше.

In [6]:

```
# task: create a list of columns to drop, call it drop_cols
# then go ahead and drop those columns from the DataFrame
# two lines of code here:
drop_cols = ['address', 'attributes', 'hours', 'is_open', 'latitude', 'longitude', 'postal_code']
business.drop(columns=drop_cols, inplace=True)
```

In [7]:

```
business.head()
```

	business_id	name	city	state	stars	review_count	categories
0	1SWheh84yJXfytovlXLOAQ	Arizona Biltmore Golf Club	Phoenix	AZ	3.0	5	Golf, Active Life
1	QXAEGFB4oINsVuTFxEYKFQ	Emerald Chinese Restaurant	Mississauga	ON	2.5	128	Specialty Food, Restaurants, Dim Sum, Imported...
2	gnKJwL_1w79qoiV3lC_xQQ	Musashi Japanese Restaurant	Charlotte	NC	4.0	170	Sushi Bars, Restaurants, Japanese
3	xvX2CttrVhyG2z1dFg_0xw	Farmers Insurance - Paul Lorenz	Goodyear	AZ	5.0	3	Insurance, Financial Services
4	HhyxOkGAM07SRYtiQ4wMFQ	Queen City Plumbing	Charlotte	NC	4.0	4	Plumbing, Shopping, Local Services, Home Servi...

Тепер ми маємо в таблиці корисні дані. Пам'ятайте наше перше завдання - визначити найпопулярніший тип ресторану. Зараз ми дивимось на стовпчик категорій. Ми могли б просто порухувати кожну окрему категорію в цьому стовпчику, як це робили в опитуванні власників домашніх тварин. Але наш стовець категорій не у такому форматі, як ці дані. Кожен елемент у категоріях насправді є рядком, як ми демонструємо нижче, причому кожен пункт, який нас цікавить, знаходиться у рядку і розділений комами.

In [8]:

```
# print the first element of business['categories']
print(business.categories[0])
```

Golf, Active Life

In [9]:

```
# what is the type of this element?
type(business.categories[0])
```

Out[9]: str

In [10]:

```
# if the first element were a list, this would return 'Tours', not 'T'
business.categories[0][0]
```

Out[10]: 'G'

Тепер ми збираємось використовувати тут метод pandas Series для того, щоб розбити ці рядки на списки.

In [11]:

```
# task: split the categories column (a pandas Series) into lists of categories,
# e.g. the string "Category1, Category2" becomes the list ["Category1", "Category2"]
# hint: pass the argument pat = ', ?' in to split on a comma which may be followed by
business['categories'] = business['categories'].str.split(pat=', ?')
```

In [12]:

```
business['categories'].head()
```

Out[12]:

```
0      [Golf, Active Life]
1  [Specialty Food, Restaurants, Dim Sum, Importe...
2      [Sushi Bars, Restaurants, Japanese]
3      [Insurance, Financial Services]
4  [Plumbing, Shopping, Local Services, Home Serv...
Name: categories, dtype: object
```

Зараз ми одержали списки типу pandas Series. Далі необхідно застосувати технологію, яку ми використали при опитуванні власників домашніх тварин. У нас немає необхідності зберігати початковий довгий список категорій. Будемо використовувати метод, який підрхаує і виводить перших 20 елементів за один прохід. Це займе для вашого комп'ютера приблизно 1-2 хвилини.

In [13]:

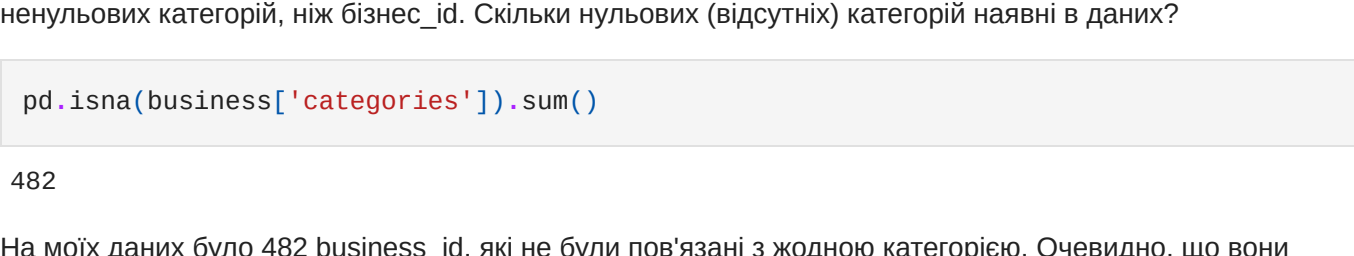
```
# task: count the frequency of each item in categories and keep the top twenty
# hint: the latter can be done by passing an argument to the head method
# assign the result to the variable category_counts - one line of code here:
category_counts = business['categories'].apply(pd.Series).stack().reset_index(drop=True)
```

Тепер познайомимось з новим видом графіків: бар-графік. Він дає нам візуальне зображення частоти перших 20 категорій.

In [14]:

```
category_counts.plot(kind='bar')
```

Out[14]: <AxesSubplot:>



Перший сюрприз - наш набір даних містить більше закладів, ніж просто бажана галузь (Ресторани). Ми отримали не лише ті дані, які нам потрібні. З часом це буде дивувати вас все менше і менше. Насправді, як науковець, ви здивуєтесь, коли хтось надасть вам чистий набір даних, що містять лише ті дані, які вам потрібні! Ресторани, як видно з графіку, є нашим найчастішим видом бізнесу, але вони становлять менше 60000 з розглянутих нами 188000 підприємств. Тепер поставимо перед собою завдання відфільтрувати та зберегти лише ті підприємства, які мають у своєму складі "Ресторан".

Перш ніж ми розпочнемо фільтрацію за категоріями, згадайте те, що на початку було менше ненульових категорій, ніж бізнес_id. Скільки нульових (відсутніх) категорій наявні в даних?

In [15]:

```
pd.isna(business['categories']).sum()
```

Out[15]: 482

На моїх даних було 482 business_id, які не були пов'язані з жодною категорією. Очевидно, що вони не принесуть ніякої користі. По-перше, ми глянемо на кілька з них, щоб перевірити

In [16]:

```
business.loc[pd.isna(business['categories'])].head()
```

	business_id	name	city	state	stars	review_count	categories
265	xOR85RiCYj64ZO3_j7Hhgg	Phoenix Valuations	Scottsdale	AZ	3.5	3	None
597	tWvep1usNNi3NYkoRbSb7g	Airdrie East Side Storage	Airdrie	AB	1.0	3	None
1020	lIG2YxnevtFhW7LVa4QJXQ	Highfield Open Mri	Pittsburgh	PA	2.0	5	None
1332	zi2CV7isWyGRxVXjdDQXg	Ceva Freight	Charlotte	NC	1.5	3	None
1770	oyPZZX2fQYpSBHk25L26A	Joe's Pastizzi Plus	Etobicoke	ON	4.5	6	None

Впевнились, що в стовпці категорій для цих підприємств немає інформації. Тепер видалимо ці 482 рядки з DataFrame. Запишемо результат повернення у початкову структуру DataFrame.

In [17]:

```
# hint, the opposite of pd.isna() is ~pd.isna()
# one line of code here, assigning the result back to business
business = business.loc[pd.notna(business['categories'])]
```

Згадаймо, як ми відфільтрували опитування власників домашніх тварин, щоб вибрати лише власників собак. Тепер ми збираємось фільтрувати наш бізнес DataFrame, щоб зберегти лише ті рядки (підприємства), які є ресторанами. Ми зробимо це так само, як робили при тестуванні того, чи "собака" в списку, хоча зберігатимемо всі стовпці, не будемо обчислювати підсумкову статистику та замінимо наш початковий DataFrame. Пам'ятаймо, що зараз у нас хороший акуратний список категорій.

In [18]:

```
business = business.loc[business['categories'].apply(lambda x: 'Restaurants' in x), :]
```

In [19]:

```
business.shape
```

Out[19]: (59371, 7)

І тепер ми можемо порухувати категорії, оскільки вони зараз мають бути ресторанами

In [20]:

```
restaurant_category_counts = business['categories'].apply(pd.Series).stack().reset_index(drop=True)
```

In [21]:

```
restaurant_category_counts.head()
```

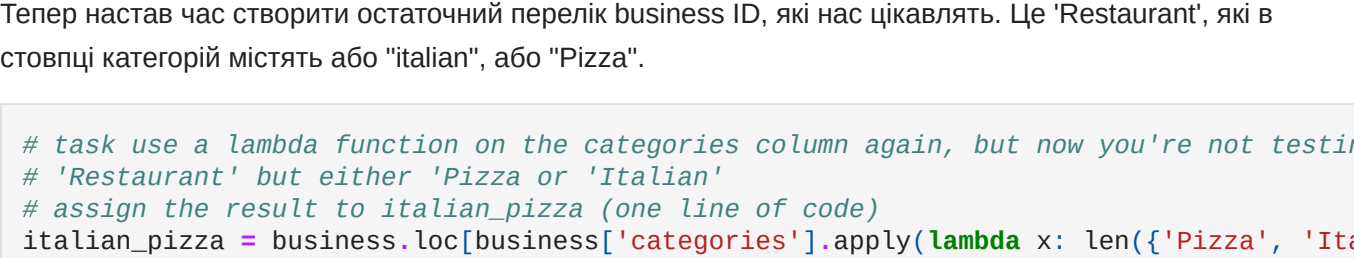
Out[21]:

```
Restaurants    59371
Food           14800
Nightlife       8562
Bars            8182
Sandwiches      7332
dtype: int64
```

In [22]:

```
# 'Restaurants', of course, is our most frequent entry because, by definition, it is
# present in every case! Plot a bar chart now of the frequencies of the other values.
restaurant_category_counts[1:].plot(kind='bar')
```

Out[22]: <AxesSubplot:>



Ми бачимо, що в нас ще є кілька категорій шуму, такі як "Food" та "Nightlife". Але ми також бачимо, що ресторани піци, як здається, були б гарною ставкою для нашого випробування маркетингової кампанії! Однак ми також помічаємо, що слово italian тут часто використовується, і, використовуючи наші знання про область дослідження, ми вважаємо, що ми будемо використовувати бізнес, який згадує або слово Pizza або italian.

Збережіть потрібний business ID у файл

Тепер настав час створити остаточний перелік business ID, які нас цікавлять. Це 'Restaurant', які в стовпці категорій містять або "italian", або "Pizza".

In [23]:

```
# task use a lambda function on the categories column again, but now you're not testing
# 'Restaurant' but either 'Pizza' or 'Italian'
# assign the result to italian_pizza (one line of code)
italian_pizza = business.loc[business['categories'].apply(lambda x: len({'Pizza', 'Italian'}) > 0)]
```

In [24]:

```
italian_pizza.head()
```

	business_id	name	city	state	stars	review_count	categories
13	fweCYi8FmbJXHCqlnWuk8w	Marco's Pizza	Mentor-on-the-Lake	OH	4.0	16	[Italian, Restaurants, Pizza, Chicken Wings]
17	PZ-LZzSlhSe9utkQYU8pFg	Carluccio's Tivoli Gardens	Las Vegas	NV	4.0	40	[Restaurants, Italian]
35	BvYU3jyGdOTJ7lyZdfiN2Q	Manzett's Tavern	Charlotte	NC	3.5	16	[Sandwiches, Italian, American (Traditional), ...]
80	PkDghu4aan2_wxrhXJTEgg	MiraKuru	Calgary	AB	3.5	16	[Nightlife, Italian, Restaurants, Japanese, Lo...
83	RfBMvEkR8IU9PJ8sWrwHg	Tavolino	Toronto	ON	4.0	18	[Italian, Restaurants]

In [25]:

```
# Check the shape of the DataFrame
italian_pizza.shape
```

Out[25]: (9224, 7)

In [26]:

```
# task: use panda's to_csv DataFrame method to save the result to a file
# use the index argument to suppress saving the index
# use the columns argument to save only the business_id, review_count, stars, and state
# one line of code here
italian_pizza.to_csv('out/04_lab4_italian_pizza.csv', index=None, columns=['business_id', 'review_count', 'stars', 'state'])
```

Підсумок

Ми працювали з достатньо суперечливими даними і зробили деякий дослідницький аналіз цих даних. Ми прочитали дані і витягували список категорій з одного стовпця, що є нетривіальною задачею. Потім ми вивчили дані та виявили популярний тип кухні, а потім використали їх для вилучення лише відповідних business ID для того, щоб використовувати їх як ключові елементи на наступному кроці.