ДОДАТОК А

Графічний матеріал кваліфікаційної роботи

Харківський національний університет радіоелектроніки
Кафедра ЕОМ

Методи та інструменти видобутку веб-контенту

Кваліфікаційна робота
Другий (магістерський) рівень

Автор:
студ. гр. СПм-21-1
Філенко В.П.

Керівник:
доц. Ільїна І.В.

2 Мета і задачі роботи

Мета: Ознайомлення з напрямом видобутку веб-контенту галузі data mining, його методів та інструментів.

Задачі:

- Огляд та порівняння методів та інструментів видобутку даних;
- Огляд проблем, які можуть виникнути в процесі автоматичного видобутку веб-контенту та представлення варіантів їх вирішення.

3 Актуальність та новизна

Актуальність: Видобуток веб-контенту застосовується практично в усіх сферах діяльності користувачів в Інтернеті:

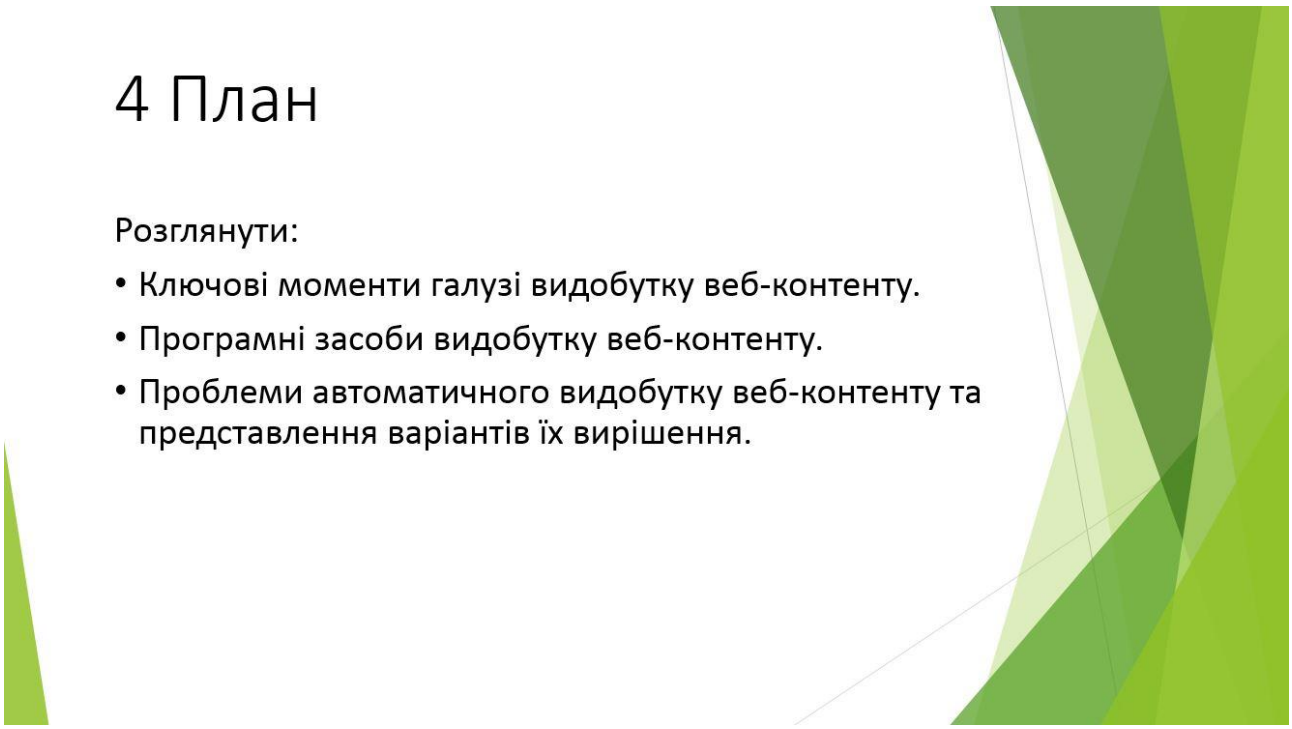
- аналіз цін на товари;
- освіта;
- аналіз коментарів в соц. мережах та форумах;
- покращення продаж тощо.

Новизна: Власна систематизація знань з області.

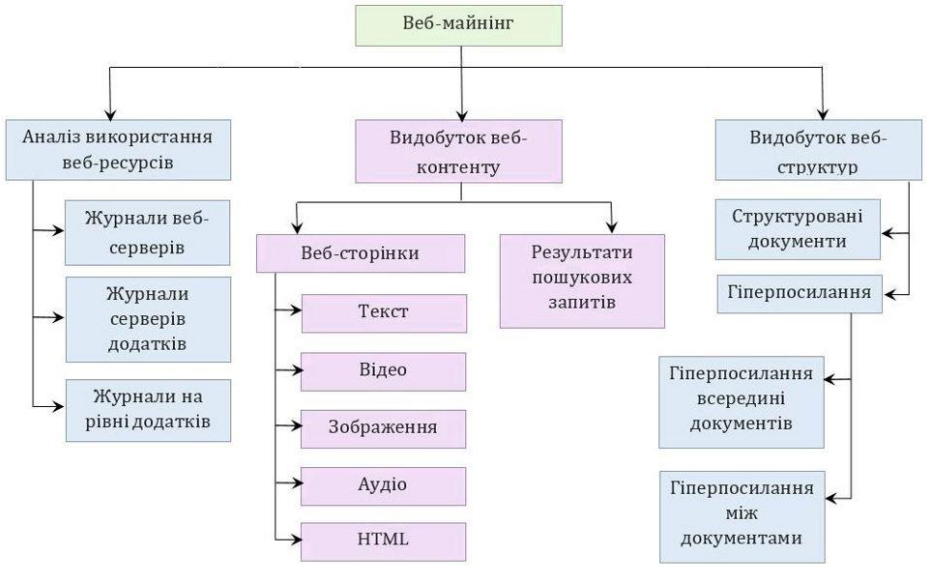
4 План

Розглянути:

- Ключові моменти галузі видобутку веб-контенту.
- Програмні засоби видобутку веб-контенту.
- Проблеми автоматичного видобутку веб-контенту та представлення варіантів їх вирішення.

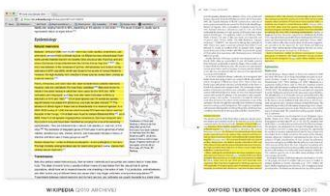


5 Видобуток веб-контенту як напрям веб-майнінгу

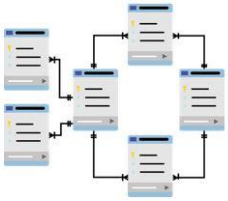


6 Типи даних для видобутку

- Неструктуровані дані



- Структуровані дані



- Напівструктуровані дані

```
{
    "orders": [
        {
            "orderno": "748745375",
            "date": "June 30, 2088 1:54:23 AM",
            "trackingno": "TN0039291",
            "custid": "11045",
            "customer": [
                {
                    "custid": "11045",
                    "fname": "Sue",
                    "lname": "Hatfield",
                    "address": "1409 Silver Street",
                    "city": "Ashland",
                    "state": "NE",
                    "zip": "68003"
                }
            ]
        }
    ]
}
```

- Мультимедійні дані

7 Методи видобутку веб-контенту

Неструктуровані дані	Вилучення інформації (Information Extraction)
	Відстежування тем (Topic Tracking)
	Резюмування (Summarization)
	Категоризація (Categorization)
	Кластеризація (Clustering)
	Візуалізація інформації (Information Visualization)
Структуровані дані	Пошуковий робот (Web Crawler)
	Врапер (Wrapper Generation)
	Видобуток контенту сторінки (Page Content Mining)
Напівструктуровані дані	Модель обміну об'єктами (Object Exchange Model)
	Вилучення зверху вниз (Top Down Extraction)
	Мова вилучення веб-даних (Web Extraction Lan-ge)
Мультимедійні дані	SKICAT
	Майнер мультимедіа (Multimedia Miner)
	Виявлення границь (Shot Boundary Detection)

8 Програмні інструменти видобутку веб-контенту

- Комерційні

- Некомерційні

import.io

IROBOT SOFT

WebExtractor360
using Regular Expressions

ContextMiner

9 Python як інструмент видобутку веб-контенту

+ підходить для масштабних та невеликих проектів;
+ висока швидкість роботи;
+ сильна підтримка спільноти.

+ ідеально підходить для невеликих проектів;
+ проста в опануванні;
+ хороша підтримка спільноти.

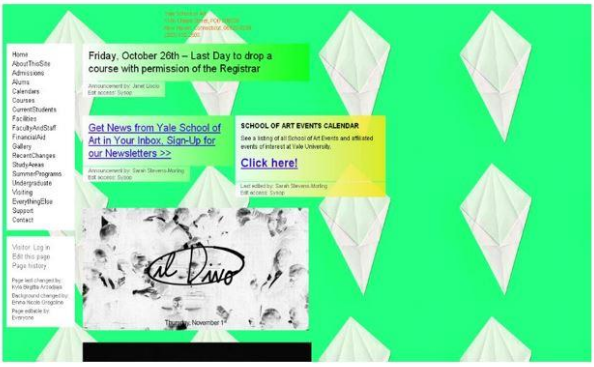
+ підтримує сценарії;
+ працює з динамічним контентом;
- повільна робота з видобутку контенту.

10 Проблеми автоматичного видобутку веб-контенту

- Проблема розмітки.
- Проблема навігації.
- Проблема забезпечення однорідності даних.
- Проблема дублікатів даних.

11 Проблема розмітки (1)

Порушення рекомендацій по верстці веб-сторінок



12 Проблема розмітки (2)

Помилки в коді сторінок сайту (1)

- Помилки в тегах посилань:

```
<http://www.***.ua/****/*****/>Текст</a>
```

↓

```
<a "http://www.***.ua/****/*****/">Текст</a>
```

a href=

- Помилки в тегах зображень:

```
<img src="…" alt="перший рядок<br/>другий рядок<br¶/>інші¶ рядки">
```

↓

```
<img src="…" alt="перший рядок<br/>другий рядок<br¶/>інші¶ рядки">
```

- використання розмітки HTML всередині значення атрибута;
- кілька «жорстких» переходів рядка (позначені символом «¶»);

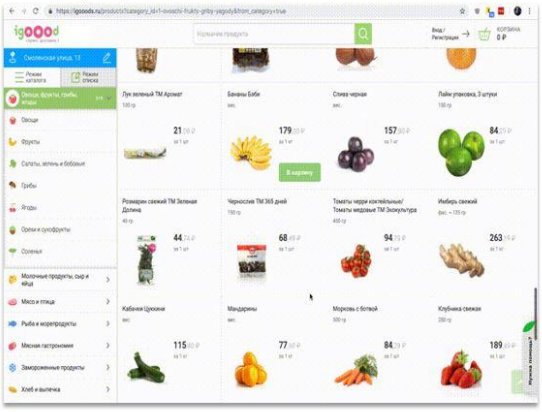
13 Проблема розмітки (3)

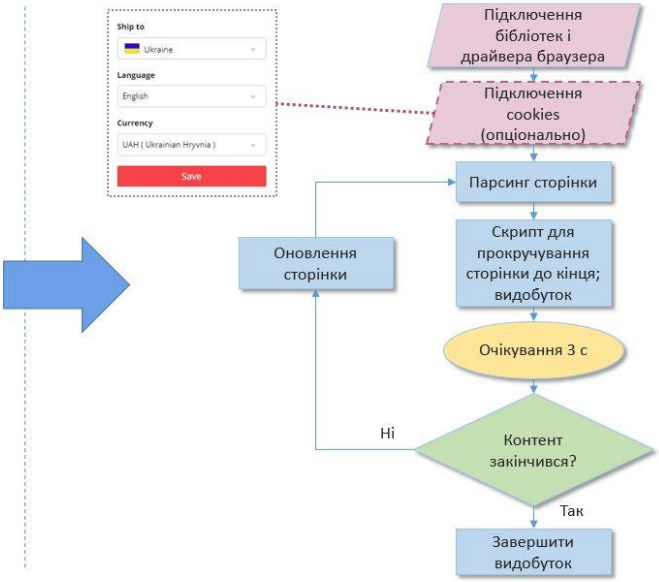
Помилки в коді сторінок сайту (2)

14 Проблема навігації (1)

Динамічна підгрузка контенту

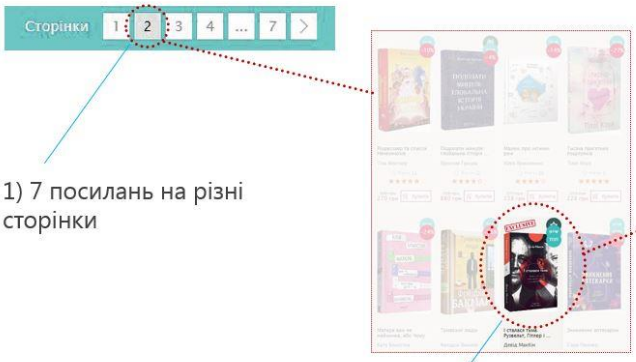
AJAX підгрузка контенту сторінки





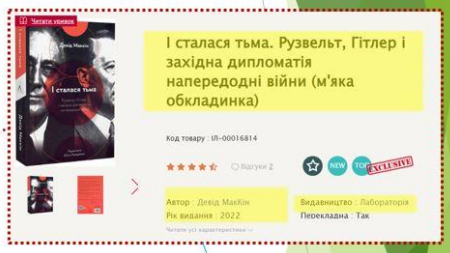
15 Проблема навігації (2)

Вкладені посилання (1)



1) 7 посилань на різні сторінки

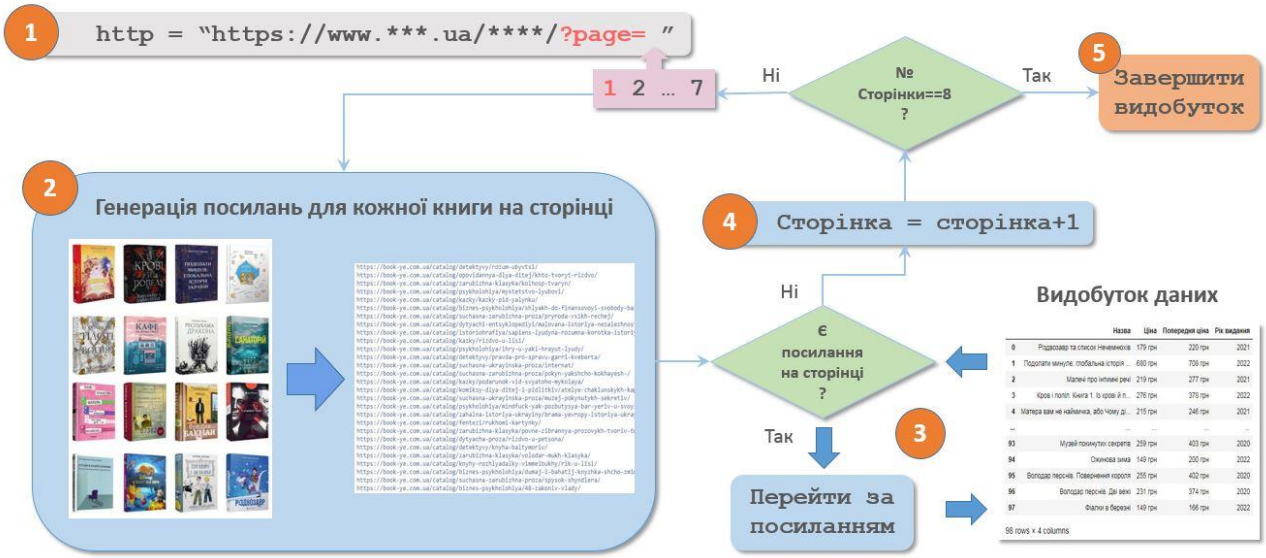
2) На кожній сторінці список книг і посилання на окрему сторінку з детальною інформацією для кожної книги



3) Потрібно видобути деяку окрему інформацію

16 Проблема навігації (3)

Вкладені посилання (2)



17 Проблема забезпечення однорідності даних

1) Кластеризація даних:

File	Author <string>	Date <date_type>	Size (MB) <float>
file1.doc	J. A. Saimon	15.10.2020	3,16
L. M. Derek	17.10.2020	2,20	<NONE>
file3.doc	A. Richard	20.11.2021	2,83
file4.doc	F. L. March	1,6	<NONE>

2) Нормалізація даних:

- Номери телефонів
  - +38 (0XX) XXX-XX-XX
  - +380 XX XXX-XX-XX
  - 0XXXXXXXXX
- Дати
  - ДД/ММ/РР
  - РР/ММ/ДД
  - День Місяць, Рік
- Населені пункти
  - м. Санкт-Петербург
  - Санкт-Петербург
  - Пітер

18 Проблема дублікатів даних

19 Висновки

Висновок: Видобуток веб-контенту є актуальною темою через величезний обсяг неструктурованих знань в Інтернеті. Для їх вилучення використовують численні методи та програмні засоби видобутку даних. Методи, запропоновані в роботі, вирішують деякі проблеми автоматичного видобутку веб-контенту.

Пропозиції для вдосконалення: Алгоритми розпаралелювання даних.

ДОДАТОК Б

Програмний код кваліфікаційної роботи

Б.1 Приклад вирішення проблеми HTML-розмітки сайту

```
####      ПІДКЛЮЧЕННЯ БІБЛІОТЕК      ####

#requests для запиту сторінки
import requests
#BeautifulSoup для парсингу веб-сторінки
from bs4 import BeautifulSoup
#basename для збереження зображень під спеціальними іменами
from os.path import basename
#figure, imshow, axis для відображення зображень
from matplotlib.pyplot import figure, imshow, axis
#imread для зчитування зображень
from matplotlib.image import imread



####        ВИДОБУТОК ДАНИХ          ####

#Отримання посилання на сайт
http = "https://book-ye.com.ua/catalog/vydavnytstva/filter/top-is-
true/"
response = requests.get(http)
html = response.content
#Внесення змін в html розмітку сторінки
html = html.replace(b'class="product__media"',
b'class="product_media"')
html = html.replace(b'src="', b'src="<br>')

#Парсинг сторінки сайту
soup = BeautifulSoup(html, "html.parser")

file_names = []
url = 'https://book-ye.com.ua'

#Знаходження зображень з обкладинками книг
image_tags = soup.find_all('img', class_="product__media")

#Спроба зчитати та завантажити зображення
#Якщо знайдено хоча б одне зображення,
if (len(image_tags)!=0):
    for image_tag in image_tags:
        #завантажити зображення на комп'ютер
        link = url+image_tag['data-src']
```

```
        with open(basename(link), "wb") as f:
            f.write(requests.get(link).content)
        file_names.append(basename(link))

#Якщо не знайдено зображень,
else:

    #шукати зображення з іншим класом,
    image_tags = soup.find_all('img', class_="product_media")

    for image_tag in image_tags:

        link = url+image_tag['data-src']
        #спробувати завантажити зображення на комп'ютер,
        try:
            with open(basename(link), "wb") as f:
                f.write(requests.get(link).content)
            file_names.append(basename(link))

        #якщо не виходить, виправити вміст посилання,
        except:

            link_fixed = link.replace('<br>', '')

            with open(basename(link_fixed), "wb") as f:
                f.write(requests.get(link_fixed).content)
            #зберегти ім'я файлу
            file_names.append(basename(link_fixed))



###    ВІЗУАЛІЗАЦІЯ ДАНИХ ОБКЛАДИНОК КНИГ    ###

fig = figure(figsize=(18, 6))
number_of_files = len(file_names)
for i in range(number_of_files):
        a=fig.add_subplot(2, number_of_files/2, i+1)
        image = imread(file_names[i])
        imshow(image, aspect='auto')
        axis('off')
```

Б.2 Приклад вирішення проблеми динамічного завантаження контенту веб-сторінки

```
####      ПІДКЛЮЧЕННЯ БІБЛІОТЕК      ####

#time для очікування
import time
#selenium для скриптів
from selenium import webdriver
from bs4 import BeautifulSoup as bs
import pandas as pd



####        ВИДОБУТОК ДАНИХ          ####

#Підключення драйвера для браузера Opera, необхідно для роботи
selenium
browser = webdriver.Opera(executable_path='C:\Program
Files\Opera\operadriver\operadriver.exe')
browser.get("https://aliexpress.ru/category/202000104/laptops.html
?g=n&page=3&spm=a2g0o.category_nav.1.220.464a5d8bsqZ2")
#Підключення cookie
cookie = {'name': 'aep_usuc_f', 'value':
'isfm=y&site=rus&c_tp=UAH&isb=y&region=UA&b_locale=uk_UA',
'domain': '.aliexpress.com'}
browser.add_cookie(cookie)
#Парсинг сторінки за допомогою BeautifulSoup
source_data = browser.page_source
soup = bs(source_data)
#Скрипт для прокручування сторінки до кінця
lenOfPage = browser.execute_script("window.scrollTo(0,
document.body.scrollHeight);var
lenOfPage=document.body.scrollHeight;return lenOfPage;")

#Продовжувати цикл, допоки не закінчиться контент каталогу з
товарами
match=False
while(match==False):
      lastCount = lenOfPage
      #Очікування 3 секунди для прогрузки наступного контенту
сторінки
      time.sleep(3)
      #Запуск скрипту
      lenOfPage
      #Якщо контент закінчився, завершити скрипт
      if lastCount==lenOfPage:
            match=True

#Парсинг оновленої сторінки за допомогою BeautifulSoup
source_data = browser.page_source
```

```
soup = bs(source_data)
#Знаходження всіх назв товарів та їх цін з каталогу
models=soup.find_all('div', {'class':['product-
snippet_ProductSnippet__name__1ettdy']})
price=soup.find_all('div', {'class':['snow-
price_SnowPrice__mainM__18x8np']})
#Довавлення назв товарів і цін в dataframe для візуалізації
df = pd.DataFrame({'Назва товару': models, 'Ціна': price})
#Закрити сторінку
browser.close()
#Вивести вміст dataframe з товарами
df
```

Б.3 Приклад вирішення проблеми розміщення даних на багатьох сторінках та вкладених сторінках; двовимірна та тривимірна візуалізація даних

```
####      ПІДКЛЮЧЕННЯ БІБЛІОТЕК      ####

#Підключення бібліотек
#%matplotlib notebook - для інтеракції з 3d графом
%matplotlib notebook
import requests
from bs4 import BeautifulSoup
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt



####        ВИДОБУТОК ДАНИХ          ####

#Парсинг першої сторінки
http = "https://book-ye.com.ua/catalog/vydavnytstva/filter/top-is-
true/"
response = requests.get(http)
html = response.content
soup = BeautifulSoup(html, "lxml")

#Створення змінних для ітерації від першої сторінки до останньої
page = 1
last_page = int(soup.find_all("a", class_="pagination__item")[-
2].get_text())

#Створення списків для подальшої візуалізації даних
titles_list = []
prices_list = []
prices_old_list = []
years_list = []
best_book = []

#Цикл ітерації від першої сторінки до останньої
while page != last_page+1:
      #Парсинг нумерованої сторінки
      http_page = f"https://book-
ye.com.ua/catalog/vydavnytstva/filter/top-is-true/?PAGEN_1={page}"
      response = requests.get(http_page)
      html = response.content
      soup = BeautifulSoup(html, "lxml")

      #Змінна, що дорівнює кількості книг на сторінці
      books_on_Page = len(soup.find_all("a",
class_="product__name"))

      #Цикл для отримання посилань на всі книги для кожної
```

```
сторінки сайту
      for i in range(0,books_on_Page):
            #Отримання посилання на окрему сторінку кожної книги
            url = 'https://book-ye.com.ua' + soup.find_all("a",
class_="product__name")[i]['href']
            #Парсинг сторінки книги
            response = requests.get(url)
            soup1 = BeautifulSoup(response.text, "html.parser")

            #Створення словника для зберігання даних книги
            book = {}
            #Знаходження заголовку, ціни, старої ціни, року
публікації книг
            title = soup.find_all("a",
class_="product__name")[i]['title']
            price = soup.find_all("div", class_="product__price-
current")[i]
            price_old = soup.find_all("div",
class_="product__price-old")[i]
            publ_year = soup1.find("meta",
itemprop="copyrightYear", content=True)['content']

            #Добавлення даних в словник і окремі списки
            book["Назва"] = title
            titles_list.append(title)
            #Добавлення даних про ціну, перетворивши з типу string
на float, без "грн"
            book["Ціна"] = float(price.get_text().replace(' ',
'').replace('грн', ''))
            prices_list.append(book["Ціна"])
            #Якщо не вказана попередня ціна, заповнити її поточною
            try:
                book["Попередня ціна"] =
float(price_old.get_text().replace('грн', ''))
            except:
                book["Попередня ціна"] = book["Ціна"]
            prices_old_list.append(book["Попередня ціна"])
            book["Рік видання"] = publ_year
            years_list.append(int(publ_year))

            #Добавити дані про книгу зі словника в єдиний список,
            #що буде містити всі книги одночасно
            best_book.append(book)

      #Dataframe для гістограми
      years_series = pd.Series(years_list)

      page = page + 1



####       ГІСТОГРАМА       ####
```

```
years_df = years_series.value_counts().to_frame().reset_index()
years_df.rename(columns={"index":"Year", 0:"Published books"},
inplace=True)

plt.figure(figsize=(14, 10))
plt.title("Кількість бестселлерів за роком видання", fontsize =
20)
plt.yticks(np.arange(0, 275, step=5))
plt.xticks(rotation= 70)
plt.ylabel("Кількість бестселлерів", fontsize = 16)
plt.xlabel("Рік видання", fontsize = 16)
plt.bar(years_df["Year"], years_df["Published books"], width =
0.5, color = "#ef6c70", edgecolor = "black")
plt.grid(color='#59656F', linestyle='--', linewidth=1, axis='y',
alpha=0.7)



####        3D ГРАФ          ####

#Підключення бібліотек
from mpl_toolkits.mplot3d.axes3d import Axes3D
import matplotlib.patches as mpatches

#Створення 3d графу
fig = plt.figure(figsize=(10, 10))
ax = Axes3D(fig)
ax.set_box_aspect(aspect = (1.5,1,1))

#Дані для осей:
#    x - Заголовки книг
#    y - Роки видання
#    z - Знижки
discounts_list = []
for i in range(0, len(prices_list)):
    discount = prices_old_list[i] - prices_list[i]
    discounts_list.append(discount)
xs=titles_list
ys=years_list
zs=discounts_list

#Налаштування кольорів і розмірів точок графу
color = []
size = []
for s in range(0, len(ys)):
    coef = zs[s]*zs[s]*0.3
    size.append(50+coef)
    if(ys[s]==2019):
        color.append('cyan')
    elif(ys[s]==2020):
        color.append('green')
    elif(ys[s]==2021):
        color.append('blue')
```

```
    else:
        color.append('red')

#Налаштування позначок x та y осей
ax.set_xticklabels(xs, rotation=45, ha='right')
ax.locator_params(axis='y', nbins=len(np.unique(ys)))
ax.set(xticks=range(len(xs)), xticklabels=xs)

#Розподілення точок на графі
ax.scatter(range(len(xs)), ys, zs, s=size, c=color)

#Спадаючі лінії від точок до площини xy
xs_int = []
for i in range(0, len(ys)):
    xs_int.append(i)
zs_l = np.asarray([[i, -1] for i in zs])
for i, p in enumerate(zs_l):
    ax.plot(xs=[xs_int[i]]*2,ys=[ys[i]]*2,zs=zs_l[i],alpha=0.2,
linewidth=2, c=color[i])

#Налаштування легенди
cyan_patch = mpatches.Patch(color='cyan', label='2019 рік')
green_patch = mpatches.Patch(color='green', label='2020 рік')
blue_patch = mpatches.Patch(color='blue', label='2021 рік')
red_patch = mpatches.Patch(color='red', label='2022 рік')
ax.legend(handles=[cyan_patch, green_patch, blue_patch,
red_patch], loc="center left")

#Налаштування зовшнішнього вигляду 3d графу
ax.set_ylabel('Рік видання', size=13, labelpad=8)
ax.set_zlabel('Знижка (грн)', size=13)
ax.xaxis._axinfo["grid"].update({"linewidth":0.2})
ax.yaxis._axinfo["grid"].update({"linewidth":0.01})
ax.zaxis._axinfo["grid"].update({"linewidth":0.01})
ax.dist = 11
```