

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
“ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ”
Кафедра прикладної математики та інформатики

Методичні вказівки до виконання розрахункової роботи
з дисципліни “Інтелектуальна обробка Web даних (Web-
Mining)”

для магістрів, що навчаються на спеціальності
122 «Комп’ютерні науки», усіх форм навчання

УДК 004.4'2:004.912(072)

М 54

Методичні вказівки до виконання розрахункової роботи з дисципліни «Інтелектуальна обробка Web даних (Web-Mining)» для магістрів, що навчаються на спеціальності 122 «Комп'ютерні науки» усіх форм навчання / уклад. М.О. Александров. – Дрогобич : ДонНТУ, 2025. – 14 с.

Наведено методичні рекомендації, контрольні питання й завдання для виконання індивідуальної розрахункової роботи.

Укладач: Микита АЛЕКСАНДРОВ, д.ф., доцент кафедри ПМІ

Рецензент: Сергій КОВАЛЬОВ, к.т.н., доц., зав. каф. ЕТ і КІ

Відповідальний за випуск: Ярослав ДОРОГИЙ, д.т.н., проф., зав. каф. ПМІ

Затверджено навчально-методичним відділом ДВНЗ ДонНТУ,
протокол №5 від 25.03.2025

Розглянуто на засіданні кафедри ПМІ, протокол № 2 від 21.02.2025р.

© ДВНЗ ДонНТУ, 2025р

ЗМІСТ

ВСТУП	4
Завдання	5
Варіанти завдань до розрахункової роботи	6
Приклад виконання	8
Питання для самоконтролю	12
СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ	14

ВСТУП

Курс «Інтелектуальна обробка Web даних (Web-Mining)» є важливою складовою підготовки магістрів за спеціальністю 122 «Комп'ютерні науки». В умовах стрімкого зростання обсягів інформації у веб-середовищі особливого значення набувають методи збору, обробки, аналізу та інтерпретації відкритих Web-даних. Зокрема, все більшої популярності набувають інтелектуальні системи, здатні автоматично генерувати персоналізовані рекомендації на основі даних з Інтернету.

Метою виконання розрахункової роботи є набуття практичного досвіду побудови індивідуальної системи рекомендацій, яка використовує відкриті Web-дані та сучасні алгоритми фільтрації — контентної, колаборативної або гібридної. У процесі виконання проєкту студенти вивчають та застосовують методи веб-збирання даних (Web Scraping/Crawling), проводять попередню обробку інформації, виділяють релевантні ознаки, будують рекомендаційну модель та візуалізують отримані результати.

В межах роботи студенти можуть обрати одну із запропонованих тем або узгодити власну, використовуючи відкриті API, RSS-канали або публічні датасети. Результатом стане створення системи, що здатна рекомендувати фільми, музику, новини або інші об'єкти на основі інтересів користувача чи подібності контенту.

Методичні вказівки містять загальні вимоги до організації розрахункової роботи, приклади реалізації та орієнтовну структуру звіту. Вони спрямовані на допомогу студентам у формуванні системного підходу до аналізу веб-даних і створення корисних прикладних рішень із використанням інструментів Python та бібліотек для роботи з даними й візуалізацій..

Завдання

1. Обрати одну з тем проєкту (див. варіанти завдань – табл. 1) або запропонувати власну за погодженням з викладачем.
2. Визначити джерело Web-даних. Зібрати їх самостійно (web scraping / crawling), або використати відкриті API чи готові датасети.
3. Провести попередню обробку даних (очищення, нормалізація, аналіз структури).
4. Виділити ознаки для побудови рекомендаційної моделі (атрибути користувачів, продуктів тощо).
5. Реалізувати базову модель рекомендацій: Content-based фільтрація, Collaborative filtering, Гібридний підхід.
6. Провести оцінку ефективності системи (точність, recall, top-N рекомендації).
7. Побудувати візуалізацію результатів (таблиці, графіки, heatmaps, тощо).
8. Оформити звіт з описом підходу, коду і висновків.

Загальні вимоги:

1. Мова програмування - Python.
2. Файл з вихідними даними або посилання на джерело
3. Обов'язкове використання інструментів:
 - Мінімум 1 інструмент для збору або завантаження Web-даних
 - Попередня обробка даних (pandas, regex, nltk/spacy – за потреби)
 - Один з методів рекомендацій (content/collaborative/hybrid)
 - Візуалізація (matplotlib/seaborn/plotly)

Вимоги до звіту:

1. Мета, завдання та варіант.
2. Обрана тема і джерело даних.
3. Опис набору даних (кількість записів, поля).

4. Кроки обробки: очищення, обробка тексту, векторизація.
5. Побудова моделі (опис алгоритму).
6. Результати і приклади.
7. Візуалізації.
8. Висновки та можливі покращення.

Варіанти завдань до розрахункової роботи

Таблиця 1 – Варіанти завдань до розрахункової роботи

№	Завдання
1	Рекомендації фільмів на основі описів або жанрів Джерело: MovieLens Dataset, TMDb API Завдання: Рекомендувати фільми, схожі на заданий за описом або жанром.
2	Рекомендації статей за інтересами користувача Джерело: RSS-канали новин (BBC, NYTimes), News API. Завдання: користувач вводить ключові слова, а система рекомендує релевантні новини або статті.
3	Рекомендації музики на основі стилів або виконавців Джерело: Last.fm Dataset, Spotify API Завдання: Рекомендації схожих треків/виконавців на основі історії прослуховувань або жанру.
4	Рекомендації статей з новинних порталів. Джерело: Web scraping з BBC, Guardian, NPR, або використання News API. Завдання: Рекомендувати схожі новини за темами/ключовими словами.
5	Рекомендації відео з YouTube. Джерело: YouTube Data API

	Завдання: Рекомендації відео на основі заголовків, описів або метаданих (теги, категорії).
6	Рекомендації туристичних напрямків. Джерело: TripAdvisor, Kaggle travel datasets, OpenTripMap API. Завдання: Рекомендація подібних міст/місць на основі опису, рейтингу або інтересів.
7	Рекомендації вакансій за резюме або інтересами. Джерело: Web scraping з Jooble, Work.ua, або використання відкритих датасетів (наприклад, з Kaggle). Завдання: Система, яка пропонує вакансії на основі опису бажаної посади або навичок.
8	Рекомендації курсів або онлайн-навчання. Джерело: Coursera API (або парсинг сайтів Coursera, Udemy, EdX) Завдання: Рекомендувати курси, схожі на обраний курс за темами/описом.
9	Рекомендації страв або рецептів. Джерело: Spoonacular API, Kaggle Recipes dataset. Завдання: Рекомендації страв, схожих на улюблену (інгредієнти, кухня, складність).
10	Рекомендації ігор за жанром та оцінками Джерело: RAWG API, Steam API Завдання: Рекомендації ігор за типом, жанром, рейтингом або сюжетом.

Вибір власної теми

Студенти можуть обрати власну тему - за умови, що:

- вона базується на Web-даних;
- передбачає створення рекомендацій;
- джерело даних верифіковане і відкрите.

Приклад виконання

Приклад виконання (рекомендаційна система книг).

Мета: створити систему, яка на основі улюбленої книги або інтересів користувача рекомендує подібні книги.

Тип фільтрації: Content-based

Джерело даних: відкритий датасет - Goodreads Books Dataset (<https://www.kaggle.com/datasets/jealousleopard/goodreadsbooks?resource=download>)

1. Завантаження та первинна обробка даних.

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import linear_kernel
import numpy as np

# Завантаження CSV-файлу
books = pd.read_csv('books.csv', encoding='latin1', on_bad_lines='skip', quotechar='')

# Вибір релевантних колонок
books = books[['title', 'authors', 'average_rating', 'ratings_count', 'text_reviews_count']]

# Попередній перегляд даних
print(books.head())
```

Рисунок 1 – Приклад коду завантаження та первинної обробки даних.

Ми фокусуємось на назві, авторі, описі та рейтингу книг - це буде основа для рекомендацій.

2. Обробка текстових описів.

```
# Заповнення відсутніх значень у title та authors
books['title'] = books['title'].fillna('')
books['authors'] = books['authors'].fillna('')

# Об'єднуємо текстову інформацію в одне поле
books['text'] = books['title'] + ' ' + books['authors']

# Побудова TF-IDF-матриці на базі комбінованого тексту
tfidf = TfidfVectorizer(stop_words='english')
tfidf_matrix = tfidf.fit_transform(books['text'])
```

Рисунок 2 – Приклад коду обробки текстових описів

3. Обчислення схожості.

```
# Обчислення косинусної подібності
cosine_sim = linear_kernel(tfidf_matrix, tfidf_matrix)

# Індекс назв книг
indices = pd.Series(books.index, index=books['title']).drop_duplicates()
```

Рисунок 3 – Приклад коду обробки текстових описів

Використовуємо косинусну схожість для обчислення, наскільки тексти книг подібні один до одного.

4. Функція рекомендацій.

```
def recommend(title, cosine_sim=cosine_sim):
    if title not in indices:
        print(f"Книга '{title}' не знайдена в базі даних.")
        return None

    idx = indices[title]
    sim_scores = list(enumerate(cosine_sim[idx]))

    # Сортуюмо за схожістю та відбираємо топ 5 книг, ігноруючи дублікати
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)[1:6]
    book_indices = [i[0] for i in sim_scores]

    # Видаляємо дублікати (якщо книга є з різними видавництвами)
    unique_books = books.drop_duplicates(subset='title') # Унікальні книги за назвою
    recommended_books = unique_books.iloc[book_indices]

    # Виведення тексту
    print(f"\n📖 Рекомендації на основі книги: **{title}**\n")
    for i, row in recommended_books.iterrows():
        print(f"💎 Назва: {row['title']}")
        print(f"✍️ Автор(и): {row['authors']}")
        print(f"★ Середній рейтинг: {row['average_rating']}")
        print("-" * 60)

    return recommended_books[['title', 'authors', 'average_rating']]

# Приклад
print(recommend("The Lord of the Rings: The Art of the Fellowship of the Ring"))
```

Рисунок 4 – Приклад коду обробки текстових описів

В результаті система виведе 5 схожих книг за описом (рис. 5).

5. Результат роботи програми.

```
In [121]: runfile('C:/Users/Dartmoor/.spyder-py3/untitled2.py', wdir='C:/Users/Dartmoor/.spyder-py3')
          title ... text_reviews_count
0 Harry Potter and the Half-Blood Prince (Harry ... 27591
1 Harry Potter and the Order of the Phoenix (Har... 29221
2 Harry Potter and the Chamber of Secrets (Harry... 244
3 Harry Potter and the Prisoner of Azkaban (Harr... 36325
4 Harry Potter Boxed Set Books 1-5 (Harry Potte... 164

[5 rows x 5 columns]
```

 Рекомендації на основі книги: **The Lord of the Rings: The Art of the Fellowship of the Ring**

- ♦ Назва: The Lord of the Rings: Weapons and Warfare
✎ Автор(и): Chris Smith/Christopher Lee/Richard Taylor
★ Середній рейтинг: 4.53
-
- ♦ Назва: Vampire Hunter D Volume 04: Tale of the Dead Town
✎ Автор(и): Hideyuki Kikuchi/Yoshitaka Amano
★ Середній рейтинг: 3.92
-
- ♦ Назва: The Fellowship of the Ring (The Lord of the Rings #1)
✎ Автор(и): J.R.R. Tolkien
★ Середній рейтинг: 4.36
-
- ♦ Назва: In the Dark of the Night
✎ Автор(и): John Saul
★ Середній рейтинг: 3.88
-
- ♦ Назва: The Lord of the Rings: Complete Visual Companion
✎ Автор(и): Jude Fisher
★ Середній рейтинг: 4.50
-

```
          title ... average_rating
25 The Lord of the Rings: Weapons and Warfare ... 4.53
4628 Vampire Hunter D Volume 04: Tale of the Dead Town ... 3.92
23 The Fellowship of the Ring (The Lord of the Ri... 4.36
1853 In the Dark of the Night ... 3.88
26 The Lord of the Rings: Complete Visual Companion ... 4.50

[5 rows x 3 columns]
```

Рисунок 5 – Приклад результату роботи програми.

Лістинг.

```
import pandas as pd
import time
import random
import matplotlib.pyplot as plt
from pytrends.request import TrendReq

pytrends = TrendReq(hl='en-US', tz=360)
keywords = ["AI", "Machine Learning"]

for keyword in keywords:
    pytrends.build_payload([keyword], timeframe='today 12-m', geo='US')
    data = pytrends.interest_over_time()
    if data.empty:
        print(f"Немає даних для {keyword}")
        continue
    print(data.columns) # Перевіряємо назви колонок
    plt.plot(data.index, data.iloc[:, 0], marker='o', label=data.columns[0])
    time.sleep(5+random.randrange(0, 10, 1)) # Затримка між запитами

plt.legend()
plt.xticks(rotation=45)
plt.xlabel("Дата")
plt.ylabel("Інтерес")
plt.title("Тренди за 12 місяців")
plt.show()
```

Питання для самоконтролю

1. Що таке веб-тренди? Які фактори впливають на їхнє формування?
2. Які основні метрики використовуються для аналізу популярності пошукових запитів у Google Trends?
3. Як визначити сезонність пошукових запитів? Які методи використовуються для її виявлення?
4. Як глобальні події можуть впливати на популярність певних запитів у пошукових системах?
5. Які переваги та обмеження використання Google Trends для аналізу веб-трендів?
6. Що таке порівняльний аналіз трендів? Як його можна використати в бізнесі чи маркетингу?
7. Як можна визначити "зростаючі" та "спадаючі" тренди за допомогою аналізу пошукових запитів?
8. Які існують альтернативи Google Trends для збору даних про популярність пошукових запитів?
9. Чому важливо враховувати географічний фактор при аналізі веб-трендів?
10. Як можна застосувати аналіз трендів для прогнозування майбутніх змін у певній галузі?
11. Що таке веб-тренди? Які фактори впливають на їхнє формування?
12. Які основні метрики використовуються для аналізу популярності пошукових запитів у Google Trends?
13. Як визначити сезонність пошукових запитів? Які методи використовуються для її виявлення?
14. Як глобальні події можуть впливати на популярність певних запитів у пошукових системах?

15. Які переваги та обмеження використання Google Trends для аналізу веб-трендів?
16. Що таке порівняльний аналіз трендів? Як його можна використати в бізнесі чи маркетингу?
17. Як можна визначити "зростаючі" та "спадаючі" тренди за допомогою аналізу пошукових запитів?
18. Які існують альтернативи Google Trends для збору даних про популярність пошукових запитів?
19. Чому важливо враховувати географічний фактор при аналізі веб-трендів?
20. Як можна застосувати аналіз трендів для прогнозування майбутніх змін у певній галузі?

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Aggarwal C. C. Recommender Systems: The Textbook [Електронний ресурс]. - Springer, 2016. — Режим доступу : <https://link.springer.com/book/10.1007/978-3-319-29659-3>. - Назва з екрана.
2. Ricci F. Recommender Systems Handbook [Електронний ресурс] / F. Ricci, L. Rokach, B. Shapira. – Springer, 2015. - Режим доступу: . <https://link.springer.com/book/10.1007/978-1-4899-7637-6>. - Назва з екрана.
3. Pandas Documentation [Електронний ресурс]. – Режим доступу: <https://pandas.pydata.org/docs/> - Назва з екрана.
4. Scikit-learn Documentation [Електронний ресурс]. – Режим доступу: <https://scikit-learn.org/stable/documentation.html>. - Назва з екрана.
5. Matplotlib Documentation [Електронний ресурс]. – Режим доступу: <https://matplotlib.org/stable/index.html>. - Назва з екрана.
6. BeautifulSoup Web Scraping Tutorial [Електронний ресурс]. – Режим доступу: <https://realpython.com/beautiful-soup-web-scraper-python>. - Назва з екрана.
7. News API Documentation [Електронний ресурс]. – Режим доступу: <https://newsapi.org/docs>. - Назва з екрана.
8. Spotify for Developers – Web API Reference [Електронний ресурс]. – Режим доступу: <https://developer.spotify.com/documentation/web-api>. - Назва з екрана.
9. The Movie Database (TMDb) API Documentation [Електронний ресурс]. – Режим доступу: <https://developer.themoviedb.org/docs>. - Назва з екрана.
10. Kaggle Datasets [Електронний ресурс]. – Режим доступу: <https://www.kaggle.com/datasets>. - Назва з екрана.