Міністерство освіти і науки України
Національний технічний університет України «КПІ» імені Ігоря Сікорського
Кафедра обчислювальної техніки ФІОТ

ЗВІТ
з модульної контрольної
з навчальної дисципліни «Data Science Technology»

Виконав:
Студент 3 курсу кафедри ОТ ФІОТ,
Навчальної групи ІМ-13
Тавлуй Д. О.

Перевірив:
Професор кафедри ОТ ФІОТ
Писарчук О.О.

Київ 2023

І. Білет № 8

ІІ. Завдання:
1. Метод Монте-Карло. Сутність та реалізація.
2. Технології Data Mining.
3. Реалізувати скрипт із визначенням частоти появи слів у тексті та побудови гістограми частоти.

ІІІ. Результати виконання модульної контрольної роботи.

3.1. Відповідь на теоретичне питання №1.

Метод Монте-Карло - це статистичний метод, який використовує випадкові величини для моделювання різних явищ з метою отримання числових результатів. Цей метод отримав свою назву на честь казино в Монте-Карло, оскільки випадкові числа використовуються для вирішення певних завдань так само, як в азартних іграх.
Сутність методу Монте-Карло полягає в тому, щоб генерувати випадкові дані, використовувати їх для створення моделі або симуляції певного явища, а потім аналізувати ці дані для отримання результатів або прогнозів.
Основні етапи реалізації методу Монте-Карло включають:

- формування моделі;
- генерація випадкових величин;
- виконання симуляції;
- аналіз результатів;
- формування висновків.

Загалом, метод Монте-Карло може бути використаний для вирішення різноманітних завдань, таких як обчислення інтегралів, апроксимація значень математичних функцій, моделювання фізичних явищ, оцінка ризиків у фінансах та багато інших. Важливим перевагою методу є його універсальність і можливість застосування в різних областях.

3.2. Відповідь на теоретичне питання №2.

Data Mining технології застосовуються для виявлення раніше невідомих, але потенційно корисних зв'язків у обширних даних. Цей процес включає в себе застосування різноманітних методів аналізу даних, статистики, машинного навчання та інших технік з метою виявлення закономірностей та корисної інформації.
Завдання Data Mining іноді називають закономірностями або техніками. Зазвичай вони включають такі аспекти, як класифікація, кластеризація, прогнозування, асоціація, візуалізація, аналіз та виявлення відхилень, оцінювання, аналіз зв’язків та підведення підсумків.
Класифікація є найпоширенішою і простою задачею, де виявляються ознаки, які характеризують групи об’єктів у наборі даних (класи), і новий об’єкт призначається до конкретного класу відповідно до цих ознак.
Кластеризація є логічним розширенням класифікації, де класи об’єктів спочатку не визначені, і результатом є розбиття об’єктів на групи.
У відмінності від попередніх задач, асоціація визначає закономірності між подіями, які відбуваються одночасно, а не на основі властивостей аналізованого об’єкта.
Послідовність дозволяє виявити тимчасові закономірності між транзакціями, встановлюючи високу ймовірність ланцюжка подій, пов’язаних у часі.

Задача прогнозування оцінює майбутні значення цільових числових показників на основі історичних даних.
Виявлення відхилень полягає в ідентифікації даних, які відрізняються від загальної множини, і подальший аналіз цих відхилень.
Завдання оцінювання передбачає прогнозування неперервних значень ознаки.
Аналіз зв'язків включає в себе знаходження залежностей в наборі даних.
Візуалізація створює графічний образ аналізованих даних, використовуючи графічні методи для виявлення закономірностей.
Підведення підсумків полягає в описі конкретних груп об'єктів у наборі даних.

3.3. Відповідь на практичне питання №3.

3.3.1. Математична модель.

Моє завдання полягає у тому, що потрібно обчислити частоту слів у заданому тексті та побудувати її гістограму.
Спочатку створено функцію яка зчитує файл формату .txt.
Потім функція calculate_word_frequencies обчислює частоту кожного слова у тексті. Вона використовує регулярний вираз для розділення тексту на слова, видаливши знаки пунктуації. Також вона включає в собі параметр, в залежності того чи потрібно відрізняти регістр слів у тексті, чи ні.
Після цього є функція яка будує саму гістограму на основі результатів обчислення.
У головній частині коду здійснюється читання тексту з файлу, обчислення частот слів та виведення їх на екран. Потім викликається функція яка будує гістограму.

3.3.2. Опис структурних рішень.

Я використав монолітну архітектуру проектування, бо сама програма не дуже складна та громістка. Уся логіка реалізована у одному файлі під назвою main.py. Також проект складається з файлу input.txt, у якому задається певний текст.

3.3.3. Результати роботи програми відповідно до завдання.

Перш за все, я не міг визначитися, чи потрібно розрізняти слова за регістром, тому вирішив зробити так, щоб можна було обрати режим роботи:

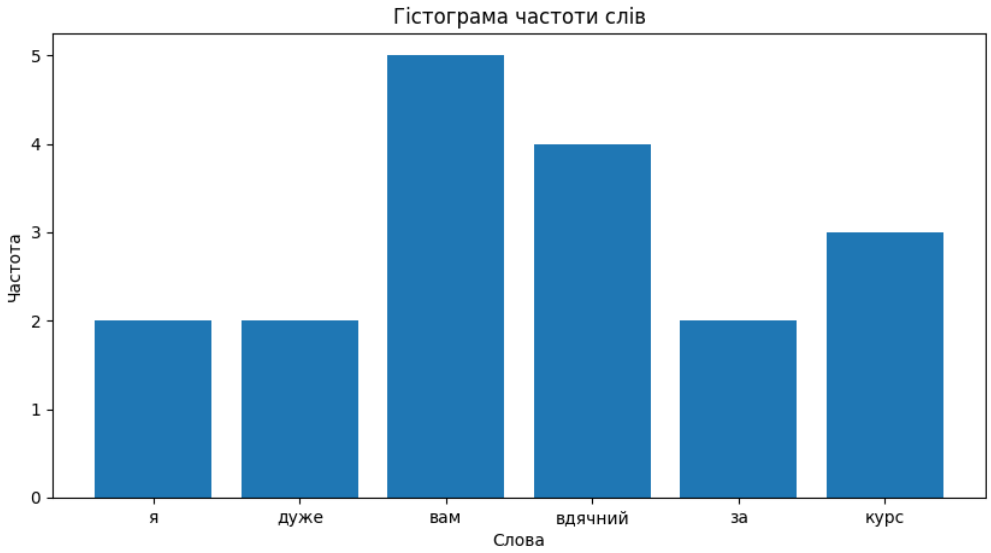
```
Оберіть як визначити частоту слів:

1 - Не розрізняючи регістр
2 - Розрізняючи регістр
```

Ось текст який буде оброблюватися: я я дуже дуже Вам Вам вам вам вам вдячний Вдячний вдячний вдячний за за курс курс курс. У ньому містяться як слова з великої літери, так і з малої.
Якщо обрати перший режим роботи, виведеться у консоль такий результат:

```
Режим:1
я: 2 разів
дуже: 2 разів
вам: 5 разів
вдячний: 4 разів
за: 2 разів
курс: 3 разів
```

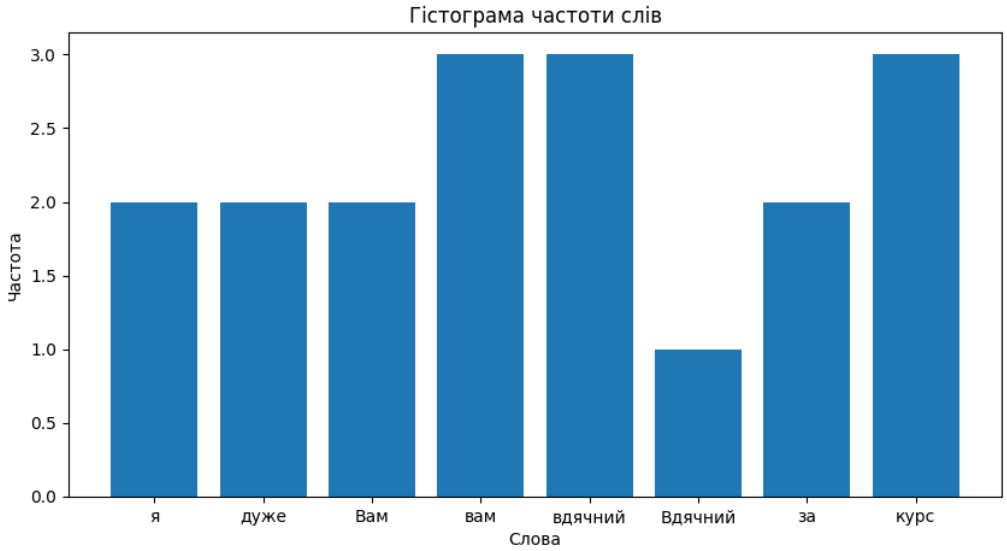
Та така гістограма:



Як можна побачити, усі слова правильно порахувались.

Тепер оберемо другий режим роботи:

```
Режим:2
я: 2 разів
дуже: 2 разів
Вам: 2 разів
вам: 3 разів
вдячний: 3 разів
Вдячний: 1 разів
за: 2 разів
курс: 3 разів
```



Програма тепер зважає на регістр слів, та також правильно рахує їх частоту.

3.3.4. Програмний код, що забезпечує отримання результату.

```
import matplotlib.pyplot as plt
import re


def read_text_from_file(file_path):
    with open(file_path) as file:
        return file.read()


def calculate_word_frequencies(text, case_sensitive=False):
    words = re.findall(r'\b\w+\b', text)

    word_freq = {}
    for word in words:
        key = word if case_sensitive else word.lower()
        if key in word_freq:
            word_freq[key] += 1
        else:
            word_freq[key] = 1

    return word_freq


def plot_word_histogram(word_freq):
    words = list(word_freq.keys())
    frequencies = list(word_freq.values())

    plt.figure(figsize=(10, 5))
    plt.bar(words, frequencies)
    plt.xlabel('Слова')
    plt.ylabel('Частота')
    plt.title('Гістограма частоти слів')
    plt.show()


if __name__ == "__main__":
    file_path = "input.txt"
```

```
    input_text = read_text_from_file(file_path)

    if input_text:

        print('Оберіть як визначити частоту слів:\n')
        print('1 - Не розрізняючи регістр')
        print('2 - Розрізняючи регістр\n')
        mode = int(input('Режим:'))

        if mode == 1:
            word_freq = calculate_word_frequencies(input_text)

            for word, freq in word_freq.items():
                print(f'{word}: {freq} разів')

            plot_word_histogram(word_freq)
        else:
            word_freq = calculate_word_frequencies(input_text,
case_sensitive=True)

            for word, freq in word_freq.items():
                print(f'{word}: {freq} разів')

            plot_word_histogram(word_freq)
```

Виконав студент: Тавлуй Дмитро Олександрович, група ІМ-13