

Санкт-Петербургский государственный университет

Направление "Большие данные и распределенная цифровая платформа"

**Лабораторная работа по дисциплине
Функциональное программирование**

**"Создание программы для обнаружения и блокировки
подозрительного сетевого трафика."**

Выполнил:

Зайнуллин Мансур Альбертович

Группа: 23.Б16-пу

Руководитель:

Киямов Жасур Уткирович

Санкт-Петербург

2024

Оглавление

1	Цель работы	2
2	Описание задачи	3
3	Теоретическая часть	5
3.1	Рекомендательные системы	5
4	Описание программы	9
4.1	Структура проекта	9
4.2	Модули программы	9
4.2.1	config.json	9
4.2.2	books _{data} .csv	9
4.2.3	genres _{list} .json	10
4.2.4	recommendation_system.py	10
4.2.5	main.py	11
4.3	Основные шаги работы программы	11
4.4	Алгоритм работы программы	12
5	Рекомендации пользователю	13
5.1	Инструкция по установке программы	13
5.2	Инструкция по эксплуатации программы	14
5.2.1	Запуск приложения	14
5.2.2	Использование интерфейса	15
6	Контрольный пример	17
7	Вывод по работе	20
8	Полезные ссылки	21

1 Цель работы

Цель данной работы разработать программное приложение, реализующее функциональную рекомендательную систему для выбора книг. Приложение должно анализировать предпочтения пользователей и на основе введенных данных предлагать книги, которые могут быть интересны пользователю.

2 Описание задачи

Необходимо разработать программное приложение, реализующее рекомендательную систему для выбора книг на основе предпочтений пользователя. Приложение должно включать следующие функциональные компоненты:

1. Форма ввода данных пользователя:

- Возможность указания любимых жанров.
- Ввод имен любимых авторов.
- Ввод предпочитаемых ключевых слов.

2. База данных книг:

- Хранение predetermined набора книг с метаданными, включая название, автора, жанр, краткое описание и год публикации.
- Формат хранения данных: JSON или CSV.

3. Алгоритм рекомендаций:

- Сравнение введенных пользователем данных с базой книг.
- Присвоение каждой книге рейтинга соответствия на основе совпадений по жанрам, авторам и ключевым словам.
- Формирование списка рекомендованных книг, отсортированного по убыванию рейтинга.

4. Фильтры и сортировка:

- Возможность фильтрации рекомендаций по указанным жанрам и году публикации.
- Опции сортировки рекомендаций по рейтингу соответствия, алфавиту или году публикации.

5. Интерактивность:

- Возможность добавления выбранных книг в список “прочитать”.
- Сохранение списка рекомендаций в файл формата CSV или JSON.
- Веб-интерфейс для взаимодействия с пользователем, реализованный с использованием Tkinter.

6. Архитектура приложения:

- Модуль загрузки данных из файла.
- Модуль обработки предпочтений пользователя.
- Модуль генерации рекомендаций.
- Модуль пользовательского интерфейса.

3 Теоретическая часть

3.1 Рекомендательные системы

Рекомендательные системы предназначены для предсказания предпочтений пользователей и предоставления им персонализированных рекомендаций. Существуют два основных подхода к созданию таких систем:

1. Коллаборативная фильтрация:

- Основана на анализе поведения пользователей.
- Предсказывает предпочтения на основе сходства между пользователями или элементами.
- Не требует детальных метаданных о рекомендуемых объектах.

2. Контентная фильтрация:

- Использует характеристики объектов для формирования рекомендаций.
- Сравнивает свойства объектов с предпочтениями пользователя.
- Эффективна при наличии подробной информации о объектах.

В данном проекте применяется контентная фильтрация, так как рекомендации формируются на основе метаданных книг (жанры, авторы, ключевые слова) и предпочтений пользователя.

Методы обработки данных

Для реализации рекомендательной системы используются следующие методы обработки данных:

1. Загрузка и предобработка данных:

- Чтение данных из файлов формата JSON или CSV.
- Очистка и нормализация данных (приведение к нижнему регистру, удаление лишних пробелов).

2. Анализ предпочтений пользователя:

- Сбор и структурирование данных о предпочтениях (жанры, авторы, ключевые слова).
- Преобразование ввода пользователя в удобный для сравнения формат.

3. Вычисление рейтинга соответствия:

- Присвоение баллов книгам на основе совпадений с предпочтениями пользователя.
- Различные веса для совпадений по жанрам, авторам и ключевым словам.

4. Фильтрация и сортировка рекомендаций:

- Применение пользовательских фильтров (по жанрам, году публикации).
- Сортировка списка книг по выбранному критерию (рейтинг, алфавит, год).

Алгоритм рекомендаций

Алгоритм рекомендаций состоит из следующих шагов:

1. Сравнение жанров:

- Каждое совпадение жанра между предпочтениями пользователя и жанрами книги увеличивает рейтинг соответствия на 3 балла.

2. Сравнение авторов:

- Каждое совпадение автора между предпочтениями пользователя и авторами книги увеличивает рейтинг на 5 баллов.

3. Сравнение ключевых слов:

- Каждое совпадение ключевого слова в описании книги увеличивает рейтинг на 2 балла.

4. Формирование списка рекомендаций:

- Книги с рейтингом соответствия выше 0 включаются в список рекомендаций.
- Список сортируется по убыванию рейтинга соответствия.

Обработка ошибок и валидация

Для обеспечения надежности приложения реализованы механизмы обработки ошибок и валидации пользовательского ввода:

1. Проверка формата ввода:

- Проверка корректности введенных данных (например, проверка, что год публикации является числом).

2. Обработка отсутствующих данных:

- Замена отсутствующих описаний книг на пустые строки.
- Обработка случаев, когда пользователь не указал предпочтений.

3. Исключение ошибок при сохранении файлов:

- Обработка исключений при попытке сохранить рекомендации или список “прочитать” в файл.
- Вывод сообщений об ошибках пользователю в случае неудачных операций.

4 Описание программы

Программа представляет собой рекомендательную систему для выбора книг на основе предпочтений пользователя. Она состоит из нескольких модулей, каждый из которых выполняет определенные функции. Ниже приведено описание основных компонентов программы и их взаимодействия.

4.1 Структура проекта

```
Books_recommendations/  
  db/  
    books_data.csv  
    Books_rating.csv  
    genres_list.json  
  src/  
    config.json  
    get_possible_preferences.py  
    main.py  
    recommendation_system.py
```

4.2 Модули программы

4.2.1 config.json

```
{  
  "books_data_file": "/Users/mansurzainullin/MyCode/Python/  
  "top_genres_count": 500,  
  "genres_file": "db/genres_list.json"  
}
```

4.2.2 books_data.csv

Файл содержит базу данных книг с метаданными, такими как название, автор, жанр, описание и год публикации.

4.2.3 `genres_list.json`

Файл содержит список жанров, отсортированных по частоте появления в базе данных книг.

4.2.4 `recommendation_system.py`

Модуль отвечает за обработку данных и генерацию рекомендаций.

Функции:

- `load_books_data()`: Загружает данные о книгах из CSV файла, обрабатывает авторов, жанры и год публикации.
- `process_authors(authors_str)`: Преобразует строку авторов в список.
- `process_genres(genres_str)`: Обрабатывает строку жанров, разделяя их по символу ‘&’ и формируя список.
- `process_year(year)`: Извлекает год из строки даты публикации.
- `calculate_match_score(book, preferences)`: Вычисляет рейтинг соответствия книги предпочтениям пользователя.
- `generate_recommendations(books_df, preferences)`: Генерирует список рекомендованных книг на основе рейтингов.
- `apply_filters(books_df, filters, sort_option)`: Применяет фильтры и сортировку к списку книг.
- `prepare_genres_list(books_df)`: Подготавливает и сохраняет список жанров с указанием частоты.
- `get_all_genres()`: Загружает список жанров из файла.
- `get_all_authors(books_df)`: Возвращает список уникальных авторов.

4.2.5 main.py

Основной модуль, реализующий графический интерфейс пользователя с помощью Tkinter.

Функции:

- `get_user_preferences()`: Считывает предпочтения пользователя из формы ввода.
- `open_genres_window()`: Открывает окно для выбора жанров.
- `show_recommendations()`: Генерирует и отображает рекомендации на основе предпочтений.
- `display_books(books)`: Отображает список книг в таблице.
- `save_recommendations()`: Сохраняет выбранные рекомендации в файл.
- `save_read_list()`: Сохраняет список “прочитать” в файл.
- `add_to_read_list()`: Добавляет выбранные книги в список “прочитать”.
- `show_read_list()`: Отображает список “прочитать” в отдельном окне.

4.3 Основные шаги работы программы

1. **Загрузка данных:** При запуске программы загружается база данных книг из `books_data.csv` с помощью функции `load_books_data()`.
2. **Ввод предпочтений:** Пользователь вводит свои предпочтения: любимые жанры, авторов и ключевые слова.
3. **Генерация рекомендаций:** На основе введенных данных программа рассчитывает рейтинг соответствия каждой книги и формирует список рекомендаций.
4. **Фильтрация и сортировка:** Пользователь может применить дополнительные фильтры и выбрать критерий сортировки.

5. **Сохранение результатов:** Рекомендации можно сохранить в файл, а также добавить книги в список “прочитать”.

4.4 Алгоритм работы программы

1. Инициализация:

- Загрузка конфигурации из `config.json`.
- Загрузка базы данных книг из `books_data.csv`.
- Подготовка списка жанров из `genres_list.json`.

2. Сбор предпочтений пользователя:

- Пользователь выбирает жанры, вводит авторов и ключевые слова.

3. Расчет рейтинга соответствия:

- Для каждой книги проверяются совпадения по жанрам, авторам и ключевым словам.
- Присваиваются баллы за каждое совпадение.
- Итоговый рейтинг используется для сортировки рекомендаций.

4. Отображение рекомендаций:

- Рекомендованные книги отображаются в таблице.
- Пользователь может применять фильтры и менять порядок сортировки.

5. Дополнительные функции:

- Сохранение рекомендаций в файл.
- Добавление книг в список “прочитать” и сохранение этого списка.

5 Рекомендации пользователю

5.1 Инструкция по установке программы

Шаги установки:

1. Установите Python 3.9.

2. Установка Git:

- В терминале выполните команду:

```
sudo apt install git
```

- Проверьте установку:

```
git --version
```

3. Клонирование репозитория проекта:

- В терминале перейдите в желаемую директорию для хранения проекта, например:

```
cd ~/ \
```

- Клонировать репозиторий:

```
git clone https://github.com/MansurYa/books-recommen
```

- Перейдите в директорию проекта:

```
cd books-recommendations/src
```

4. Настройка виртуального окружения (опционально):

- Создайте виртуальное окружение:

```
python3.9 -m venv venv
```

- Активируйте виртуальное окружение:

```
source venv/bin/activate
```

5. Установка необходимых библиотек:

- В данном проекте используются стандартные библиотеки Python 3.9, дополнительных установок не требуется.

6. Запуск программы:

- В терминале выполните команду:

```
python3.9 main.py
```

5.2 Инструкция по эксплуатации программы

5.2.1 Запуск приложения

1. Убедитесь, что вы находитесь в директории `src` проекта:

```
cd ~/ \ /books-recommendations/src
```

2. Активируйте виртуальное окружение, если оно было создано:

```
source venv/bin/activate
```

3. Запустите программу:

```
python3.9 main.py
```

5.2.2 Использование интерфейса

1. Ввод предпочтений:

- **Любимые жанры:**

- Нажмите кнопку “Выбрать жанры”.
- В открывшемся окне выберите нужные жанры, используя поле поиска.
- Подтвердите выбор.

- **Любимые авторы:**

- Введите имена авторов через запятую в соответствующее поле.

- **Ключевые слова:**

- Введите ключевые слова через запятую в соответствующее поле.

2. Генерация рекомендаций:

- Нажмите кнопку “Показать рекомендации”.
- Программа отобразит список книг, соответствующих введенным предпочтениям.

3. Применение фильтров и сортировки:

- **Фильтр по жанрам:**

- Введите жанры через запятую в поле “Фильтр по жанрам”.

- **Фильтр по году:**

- Введите год, начиная с которого вы хотите видеть книги.

- **Сортировка:**

- Выберите критерий сортировки из выпадающего меню (Рейтинг, Алфавит, Год).
- После ввода фильтров и выбора сортировки рекомендации будут обновлены автоматически.

4. Сохранение рекомендаций:

- Выберите одну или несколько книг из списка рекомендаций.
- Нажмите кнопку “Сохранить рекомендации”.
- Выберите формат файла (CSV или JSON) и укажите путь для сохранения.

5. Работа со списком “прочитать”:

- **Добавление книг в список:**
 - Выберите книги из списка рекомендаций.
 - Нажмите кнопку “Добавить в список прочитать”.
- **Просмотр списка:**
 - Нажмите кнопку “Просмотр списка прочитать”.
 - Откроется окно с перечнем добавленных книг.
- **Сохранение списка:**
 - Нажмите кнопку “Сохранить список прочитать”.
 - Выберите формат файла (CSV или JSON) и укажите путь для сохранения.

6. Закрытие программы:

- Закройте окно приложения или нажмите сочетание клавиш для завершения работы.

6 Контрольный пример

Рекомендательная система книг

Предпочтения пользователя

Любимые жанры: Выбрать жанры

Любимые авторы:

Ключевые слова:

Фильтры и сортировка

Фильтр по жанрам (через запятую):

Книги после года:

Сортировка: Рейтинг

Показать рекомендации Сохранить рекомендации Добавить в список прочитать Просмотр списка прочитать Сохранить список прочитать

Название	Авторы	Жанры	Год	Рейтинг	Описание
----------	--------	-------	-----	---------	----------

рис. 1 Интерфейс

Рекомендательная система книг

Предпочтения пользователя

Любимые жанры: Выбрать жанры

Любимые авторы:

Ключевые слова:

Фильтры и сортировка

Фильтр по жанрам (через запятую):

Книги после года:

Сортировка:

Выберите жанры

Поиск жанра:

- ☐ ['Fiction']
- ☐ ['Religion']
- ☒ ['History']
- ☐ ['Juvenile Fiction']
- ☒ ['Biography']
- ☒ ['Autobiography']
- ☐ ['Business']
- ☐ ['Economics']
- ☐ ['Computers']
- ☐ ['Social Science']
- ☐ ['Juvenile Nonfiction']
- ☐ ['Science']
- ☐ ['Education']
- ☐ ['Cooking']
- ☐ ['Sports']

Показать рекомендации Сохранить рекомендации Добавить в список прочитать Просмотр списка

рис. 2 Выбор жанров

Список прочитать					
Название	Авторы	Жанры	Год	Рейтинг	
Leopardi: A Study in So	iris origo	['biography, autobiogra	2017	8	An extr
The Fall and Rise of the	pamela horn	['history']	2013	3	Forget
Dagger and the Cross, '	joseph hatton	['history']	2012	3	Unlike s

рис. 5 Список ”прочитать”

title	authors	genres	year	match_score	description
Leopardi: A Study in Solitude	iris origo	['biography, autobiography']	2017	8	An extremely moving account of the lonely life of the unloved and tragic genius - described as "the greatest moder
The Fall and Rise of the Stately Home	pamela horn	['history']	2013	3	Forget glossy period dramas, here is the real story of Britain's super-rich from the First World War to the end of the
Dagger and the Cross, The	joseph hatton	['history']	2012	3	Unlike some other reproductions of classic texts (1) We have not used OCR(Optical Character Recognition), as this

рис. 6 Сохранили выбранные рекомендации в csv файл

7 Вывод по работе

В ходе работы была разработана рекомендательная система для выбора книг на основе предпочтений пользователя. Программа позволяет пользователю вводить свои предпочтения, генерировать персонализированные рекомендации, применять фильтры и сохранять результаты. Применение принципов функционального программирования обеспечило модульность и расширяемость кода. Программа успешно функционирует на различных операционных системах, обеспечивая кроссплатформенность.

8 Полезные ссылки

Ссылка на репозиторий проекта: <https://github.com/MansurYa/books-recommendations.git>