Т. 1, № 1, 2019

УДК 004.942

Ю. Є. Кинаш[1], О. О. Петрушинський[1], В. М. Мищишин[2]
Національний університет "Львівська політехніка",
[1] кафедра інформаційних технологій видавничої справи,
[2] кафедра комп'ютеризованих систем автоматики

РОЗРОБЛЕННЯ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ ПІДБОРУ ФІЛЬМІВ

Розроблено рекомендаційну систему для пошуку фільмів на основі вмісту. Використано базу даних Mongo DB та утиліти з елементами машинного навчання для пришвидшення пошуку. Використання розробленої системи спрятиме економії часу при підбиранні фільмів за певними критеріями.

Ключові слова: рекомендаційна система, фільтрація на основі контенту, машинне навчання.

The recommendation system for content-based movie search has been developed. Used Mongo DB database and utilities with machine learning elements to speed up the search. Using the developed system will save time when selecting movies by certain criteria.

Keywords: recommendation system, content filtering, machine learning.

Вступ

Бажання переглядати фільми часто супроводжується довгими пошуками. Витрачений на пошуки час часто не виправдовує очікувань від перегляду, бо фільм виявляється не таким цікавим, як очікувалось спочатку. Згадана проблема проявляється у питаннях: де шукати фільм та за якими критеріями? Складність пошуку полягає у виборі якнайкращої бази даних фільмів. Для вибору можна скористатися набором даних – TMDB, де зібрано понад 5000 фільмів різних категорій та з високими рейтингами. Найчастіше фільми шукають за іменем, жанром, акторами, рейтингом і сюжетом. Саме реалізація пошукової системи за цими категоріями є актуальним завданням.

Аналіз даних для рекомендаційної системи

Система рекомендацій є підкласом системи фільтрації інформації для передбачення "рейтингу" або "уподобань" користувача. Такі системи використовують у різних областях і найчастіше визнають як генератори списків відтворення для відео та музичних послуг, таких як Netflix, YouTube і Spotify, рекомендації щодо продуктів для таких послуг, як Amazon, або рекомендації щодо вмісту для соціальних медіа-платформ, таких як Facebook і Twitter [1]. Існують також популярні платформи для конкретних тем, як ресторани, онлайн-знайомства, для вивчення наукових статей, фінансових послуг та страхування життя [2].

Рекомендаційні системи переважно використовують колаборацію фільтрацію або фільтрацію на основі вмісту. Колаборацію підходи до фільтрації створюють модель на основі колишньої поведінки користувача з раніше придбаних або вибраних елементів. Цю модель використовують для прогнозування чи оцінок для елементів, якими користувач може потенційно зацікавитися. Підходи фільтрації на основі вмісту використовують серію дискретних, попередньо позначених характеристик елемента, щоб рекомендувати додаткові елементи з подібними властивостями. Поточні системи рекомендацій зазвичай поєднують один або більше підходів у

гібридну систему [3–5]. Рекомендаційні системи можуть виступати альтернативою алгоритмам пошуку, оскільки допомагають користувачам виявляти елементи, що могли б бути не знайдені [6].

Як набір даних для виробленої рекомендаційної системи обрано найбільшу базу даних з фільмами Internet Movie Database, що містить інформацію про понад 4,7 мільйона фільмів і телесеріалів, майже 6,9 мільйонів акторів, режисерів та інших професіоналів кіно зі всього світу. Для пропонованої системи вибрано лише 5000 фільмів із найбільшими рейтингами для отримання найточніших та найактуальніших результатів.

На першому етапі зводились дані за допомогою бібліотек pandas і numpy до зручного перегляду у вигляді date frames. За допомогою бібліотеки matplotlib побудовано графік на основі входження кожного жанру у фрейм фільмів (рис. 1).

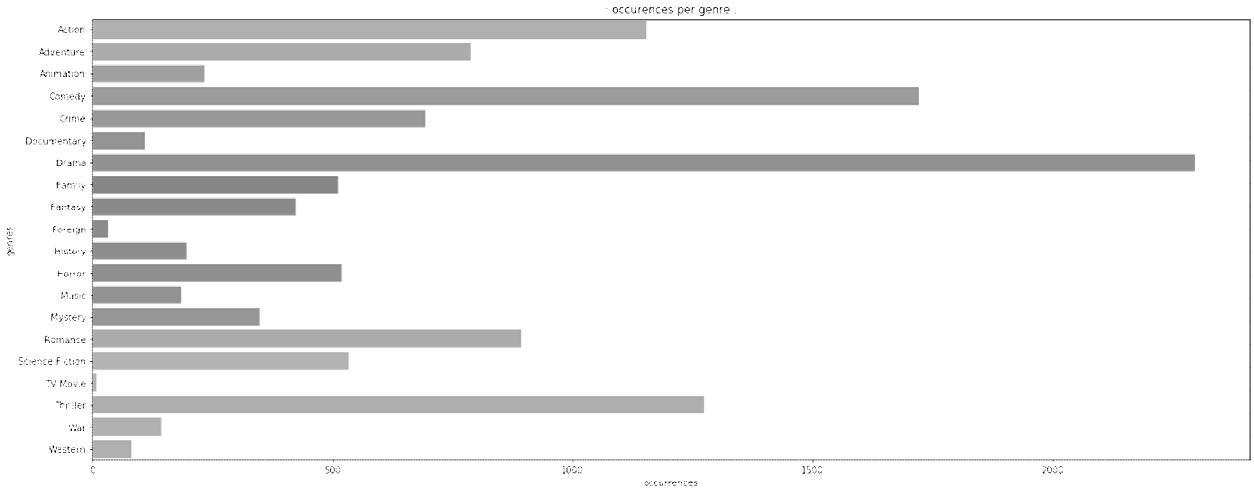

Рис. 1. Кількість входжень жанрів

На підставі аналізу набору даних фільмів оцінено залежність року, доходу та жанру (рис. 2).

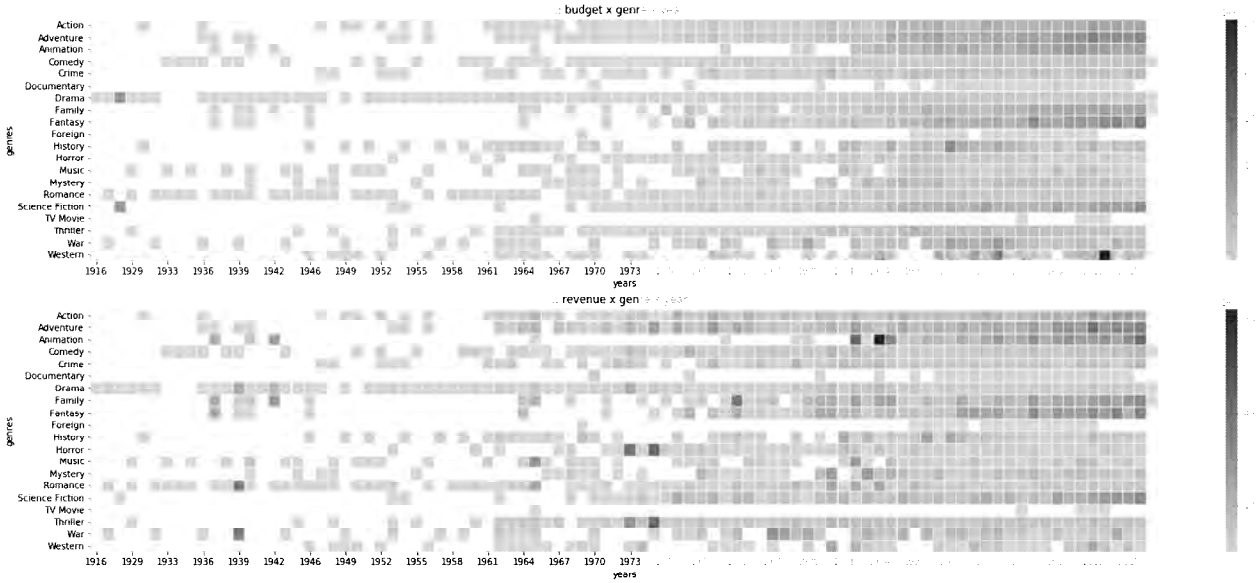

Рис. 2. Залежність між роком, доходом та жанром

Останнім етапом аналізу набору даних є оцінка входжень різних жанрів для одного фільму.

Фільтрація на основі контенту

Цей тип фільтрації використовує алгоритм для вибору елементів з подібним вмістом, щоб рекомендувати той чи інший фільм на підставі того, що використовувалось чи входило до кола уподобань. Логіка роботи моделі, що здійснює пошук фільмів на основі контенту, виглядає так, як показано на рис. 3.

Першим кроком потрібно зробити фільтрацію на основі рейтингу даних про фільми. Ми будемо використовувати зважений рейтинг WR за формулою:

$$\mathrm{WR} = \left(\frac{v}{v+m}\right) R + \left(\frac{m}{v+m}\right) C, \tag{1}$$

де v – кількість голосів за фільм; m – мінімальна кількість голосів, необхідних для переліку на графіку; R – середня оцінка фільму; C – середній голос.

Рис. 3. Логіка рекомендаційної моделі

Середній рейтинг для всіх фільмів становить близько 6 за десятибальною шкалою. Наступним кроком визначаємо відповідне значення для мінімальної кількості голосів, необхідних для відображення на графіку. Ми використовуємо 90 відсотковий поріг відсікання фільмів у списку. Після сортування за рейтингом ми отримали оцінку найпопулярніших фільмів (рис. 4).

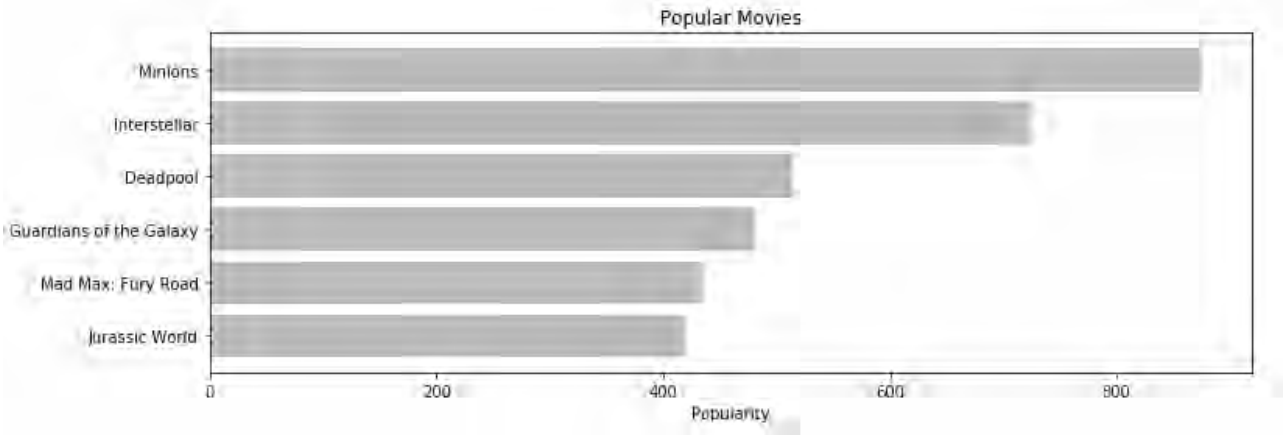

Рис. 4. Демонстрація найпопулярніших 6 фільмів

Далі обчислюємо попарні оцінки подібності для всіх фільмів на основі їхніх сюжетних описів для рекомендування фільмів за цим показником подібності. Для цього потрібно перетворити сюжети на вектори слів і обчислити вектори оберненої частоти (TF-IDF) для кожного огляду. Під частотою ми розуміємо відносну частоту слова. Зворотною частотою документів є відносна кількість документів, що містять визначений термін. Загальне значення кожного слова до документів, в яких вони з'являються, дорівнює TF*IDF. Ми отримуємо матрицю, де кожен стовпець представляє слово в оглядовому словнику і кожен стовпчик являє собою фільм. Це робиться для того, щоб зменшити важливість слів, які часто зустрічаються в сюжетах, а отже, і їх значення для обчислення остаточної оцінки подібності. Scikit-learn дає вбудований клас TfIdfVectorizer, який виробляє матрицю TF-IDF. За допомогою цієї матриці можна обчислити оцінку подібності з використанням підходів Евкліда і Пірсон та косинусної оцінки подібності. Немає однозначної відповіді, яка оцінка є найкращою. Для різних сценаріїв використовують відповідні підходи. Ми скористаємось подібністю косинуса для обчислення числової величини для позначення подібності між двома фільмами, оскільки вона не залежить від величини і є порівняно легкою та швидкою в обчисленні. Математично це можна записати так:

$$similary = \cos(\mathrm{J}) = \frac{A \cdot B}{\|A\| \cdot \|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}} \quad (2)$$

Оскільки ми використовуємо векторизатор TF-IDF, обчислення дасть нам показник подібності косинусу. Ми будемо використовувати linearar_ernel замість cosine_similarities. Потрібно створити функцію, яка приймає назву фільму як вхідний параметр і виводить список з 10 найбільш схожих фільмів. Ця функція повинна працювати за таким алгоритмом:

1. Отримати індекс фільму з назвою.
2. Отримати список оцінок подібності косинусу для цього фільму з усіма фільмами. Перетворити його на список кортежів, де першим елементом є його позиція, а другим – показник подібності.
3. Сортувати згаданий список кортежів на основі результатів подібності.
4. Отримати 10 найкращих елементів цього списку. Ігнорувати перший елемент, оскільки він відноситься до самого себе. Фільм найбільш схожий на конкретний фільм – це сам фільм.

Після побудови кінцевої матриці можна використовувати функції пошуку фільмів на основні контенту.

Оцінка та аналіз рекомендаційної моделі

Для оцінювання правильності виконання пошуку схожих фільмів потрібно ввести назву, за якою відбуватиметься фільтрація (рис. 5).

Рис. 5. Поле пошуку фільмів

Після вибору фільму появиться фрейм із трьома сторінками, де подано інформацію про сюжет, акторів та команду, яка розробляла цей фільм (рис. 6).

Рис. 6. Фрейми з даними фільму

Для отриманнях схожих фільмів потрібно натиснути на кнопку “GET SIMILAR”. Після цього у правій частині екрана з’явиться список рекомендованих фільмів (рис. 7).

Рис. 7. Список рекомендованихфільмів

Ми бачимо, що рекомендаційна система успішно захопила більше інформації через більшу кількість метаданих і надала кращі рекомендації. Очевидно, що шанувальники певного жанру любитимуть фільми того самого виробництва, тому в остаточному списку ми спостерігаємо, що більшість фільмів розробила компанія Marvel. Якщо, наприклад, користувач шукає фільм на основі першої частини, то логічно було б подивитись і наступні частини.

Висновок

Досліджено алгоритм фільтрації, що базується на аналізі контенту фільму, та розроблено оригінальне програмне рішення. Реалізовано завдання мовою програмування Python, використано бібліотеки для машинного навчання, інтерфейс користувача розроблено з використанням Angular, збереження даних – у Mongo DB.

Для швидкого пошуку очікуваного фільму використано машинне навчання, проведене за допомогою бібліотек мови Python. Досліджено програмну реалізацію рекомендаційної платформи та отримано швидкодіючий код для успішного виконання роботи сервісу.

Отримані результати засвідчують, що проект рекомендаційної системи, яка базується на фільтрації контенту фільму, доцільно використовувати з метою заощадження часу. Для отримання очікуваного результату кількість дій з користувацьким інтерфейсом зведено до мінімуму.

Список літератури

1. Чен Х., Гоу Л., Чжан Х., Джайлс К. Співробітник: пошукова система для відкриття співпраці в спільній конференції ACM / IEEE з цифрових бібліотек (JCDL) 2011.

2. "Facebook, Pandora Lead Rise of Recommendation Engines – TIME". TIME.com. 27 May 2010. Retrieved 1 June 2015.

3. Елахі, Мехді; Річчі, Франческо; Рубенс, Ніл (2016). "Огляд активного навчання в системах рекомендацій спільної фільтрації". Огляд комп'ютерних наук. 20: 29–50. doi: 10.1016 / j.cosrev.2016.05.002.

4. Herlocker, J. L.; Konstan, J. A.; Terveen, L. G.; Riedl, J. T. (січень 2004). "Оцінювання систем рекомендацій спільної фільтрації". ACM Trans. Inf. Сист. 22 (1): 5–53.

5. Джон С. Бріз; Девід Хеккерман і Карл Каді (1998). Емпіричний аналіз алгоритмів прогнозування для спільної фільтрації. В доповідях чотирнадцятої конференції «Невизначеність у штучному інтелекті» (UAI'98).

6. Mooney R. J. & L. Roy (1999). Content-based book recommendation using learning for text categorization. In Workshop Recom. Sys.: Algo. and Evaluation.

References

1. Kh. Chen, L. Hou, Kh. Chzhan, K. Dzhails Spivrobitnyk: poshukova systema dlia vidkryttia spivpratsi v spilnii konferentsii ACM / IEEE z tsyfrovykh bibliotek (JCDL) 2011.

2. "Facebook, Pandora Lead Rise of Recommendation Engines – TIME". TIME.com. 27 May 2010. Retrieved 1 June 2015.

3. Elakhi, Mekhdi; Richchi, Franchesko; Rubens, Nil (2016). "Ohliad aktyvnoho navchannia v systemakh rekomendatsii spilnoi filtratsii". Ohliad kompiuternykh nauk. 20: 29–50. doi: 10.1016 / j.cosrev.2016.05.002.

4. Herlocker, J. L.; Konstan, J. A.; Terveen, L. G.; Riedl, J. T. (sichen 2004). "Otsiniuvannia system rekomendatsii spilnoi filtratsii". ACM Trans. Inf. Syst. 22 (1): 5–53.

5. Dzhon S. Briz; Devid Khekkerman i Karl Kadi (1998). Empirychnyi analiz alhorytmiv prohnozuvannia dlia spilnoi filtratsii. V dopovidiakh chotyrnadtsiatoi konferentsii «Nevyznachenist u shtuchnomu intelekti» (UAI98).

6. R. J. Mooney & L. Roy (1999). Content-based book recommendation using learning for text categorization. In Workshop Recom. Sys.: Algo. and Evaluation.