IMDb Data Analysis

ПСА - 18/Б Софія Татош 17.12.18

Опис проблеми:

У ході роботи над проектом, була поставлена ціль визначити, як різні пари працівників кінематографу впливають на рейтинг фільмів. Це допоможе у майбутньому визначити, чи буде Ваш фільм успішним з такою командою, чи ні, відповідно до їхніх успіхів у попередніх зйомках.

Хід обробки:

Спочатку я проаналізувала доступні дані для скачування для того, щоб зрозуміти, як скомпонувати дані у вигляді, який дозволить проводити аналіз пар, обмежений параметрами користувача.
Всі необхідні для вирішення проблеми дані знаходяться у файлах:

- name.basics.tsv - містить дані про працівників кіноіндустрії
- title.basics.tsv - містить дані про кінопродукцію (фільми, серіали тощо)
- title.ratings.tsv - містить дані про рейтинги кінопродукції
- title.principals.tsv - містить дані про те, який працівник кіноіндустрії яку роль відігравав при створенні певного кінопродукту

Для того, щоб мати можливість проводити аналіз успішних пар, я розробила у вигляді словника Python наступну структуру даних:

```
{
  ((“nconst-1”, “actor”), (“nconst-2”, “director”)) : [“tconst-1”, “tconst-2”, “tconst-3”],
  ((“nconst-3”, “actor”), (“nconst-4”, “director”)) : [“tconst-5”, “tconst-6”],
}
```

(СТ-1)

Де ключем є пара, а значенням - список фільмів, які відповідають критеріям користувача. В цій структурі *nconst* - унікальний ідентифікатор працівника кіноіндустрії, *tconst* - унікальний ідентифікатор фільму, відповідно до структури файлів IMDB. Таким чином у мене є можливість виводити результат у порядку спадання за кількістю фільмів у списку.

Оскільки файли які надає IMDB містять надлишкову інформацію, це значно сповільнює роботу розробленого скрипта. Тому я прийняла рішення сформувати власні файли, які дозволять більш ефективно працювати скрипту.

Для початку я вирішила відібрати з файлу *title.basics.tsv* інформацію лише про фільми, об’єднати цю інформацію з рейтингами фільмів, що знаходиться у файлі *title.ratings.tsv*.

Обробляючи файл *title.basics.tsv* я отримала список tconst фільмів, який дозволив мені зменшити розмір файлу *title.principals.tsv.* Вихідні дані я зберегла у файл movies.ratings.tsv.

Обробляючи файл *title.principals.tsv* я отримала список nconst працівників, які були залучені до виробництва фільмів що містяться у файлі movies.ratings.tsv. Вихідні дані збережені у файлі *movies.principals.tsv*.

Файл name.basics.tsv я обробила згідно списку працівників, і вихідні зберегла у файл names.short.tsv. При цьому збережено було лише дві колонки *nconst* і primaryName, оскільки інші дані є надлишковими для моєї задачі.

Зменшення розміру файлів IMDB значно пришвидшило роботу скрипта. Наприклад файл *title.principals.tsv* займає на диску 1.4GB. Після фільтрування даних і збереження лише тих колонок, які потрібні для мого скрипта, файл *movies.principals.tsv* займає на диску 43,8MB. Тобто значно скоротився об’єм інформації яку потрібно обробляти для аналізу даних.

Таким чином я отримала файли з наступною структурою:

movies.ratings.tsv	*movies.principals.tsv*	*names.short.tsv*
• tconst • startYear • primaryTitle • averageRating • numVotes	• tconst • nconst • category	• nconst • primaryName

Для побудови структури *CT-1* скрипт запитує в користувача наступні параметри:

1. Тип пари - (“actor”, “actor”) чи (“actor”, “director”) і т.д.
2. Мінімальний та максимальний рік випуску фільмів
3. Мінімальний та максимальний рейтинг фільмів
4. Кількість результатів для виводу - кількість успішних чи неуспішних пар.

Далі скрипт:

1. фільтрує файл *movies.ratings.tsv* згідно введених параметрів і формує список *tconst* фільмів - *movies_ids_set*.
2. фільтрує дані з файлу *movies.principals.tsv згідно movies_ids_set* і формує структуру *CT-1*.
3. отримавши структуру, перевіряє чи вона не порожня (якщо порожня - виводить відповідне повідомлення)
4. Якщо CT-1 не порожня, сортує структуру за розміром списку фільмів
5. використовуючи файли *movies.ratings.tsv* і *names.short.tsv* замінює *nconst і tconst* в структурі CT-1 на ім’я людини і назву фільму відповідно і виводить цю інформацію на екран.

Проблеми:

Мова Python дозволяє стандартними засобами обробляти CSV файли. Але зважаючи на те що обробка таких файлів великих розмірів відбувається досить повільно, я прийняла рішення використати бібліотеку *pandas* (https://pandas.pydata.org/). *Pandas* дозволила мені маніпулювати даними набагато швидше завдяки розширенням, що написані мовою сі.

Для формування СТ-1 я використала стандартний словник Python. Проте стандартний словник не підтримує сортування. Для того щоб мати можливість відсортувати результати і вивести їх у порядку спадання, я використала структуру *OrderedDict* з бібліотеки *collections*.

Результати:

Виконуючи цю роботу я отримала скрипт який дозволяє визначити пари працівників кіноіндустрії, робота яких потенційно виводить фільм у топ за версією IMDB. Також, вказавши низький діапазон рейтингу, можемо визначити співпраця яких працівників кіноіндустрії навпаки, тягне рейтинги до низу.

Приклад виконання програми:

```
Please, choose which pairs you want to compare

1 - If you want to compare top films by actor and actress
2 - If you want to compare by actor and actor
3 - If you want to compare by actress and actress
4 - If you want to compare by actor and director
5 - If you want to compare by actress and director
6 - If you want to compare by producer and director
7 - If you want to compare by producer and actor
8 - If you want to compare by producer and actress

Your choice: 4
Please, enter the minimum year from 1900 to 2018
(Consider the fact that minimum year cannot be the same as maximum): 1995
Please, enter the maximum year from 1900 to 2018: 2018
Please, enter the minimum rating from 0.0 to 10.0
 (Consider the fact, that minimum and maximum cannot be the same): 7
Now, please, enter the maximum rating from 0.0 to 10.0: 10
Please, enter a number of pairs you want to get after analyzing: 2

We are trying to get some results for you. This may take a couple of minutes. Please relax.

It can take from 3 to 5 minutes. Do not worry, it is working!
\
Wow! We have found some interesting results for you. Now we are printing it.
```

```
William Winckler involved as director and Bradford Hill involved as actor in 13 films:
Gaiking I (2011) with rating 8.0
Starzinger II (2011) with rating 7.9
Starzinger III (2011) with rating 7.7
Gaiking II (2011) with rating 8.3
Danguard Ace 2 (2010) with rating 7.4
Space Pirate Captain Harlock 2 (2010) with rating 7.5
Gaiking III (2011) with rating 7.7
Starzinger (2011) with rating 7.9
Space Pirate Captain Harlock (2010) with rating 7.1
Kitaro's Graveyard Gang 2 (2011) with rating 7.3
Danguard Ace (2010) with rating 7.2
Kitaro's Graveyard Gang (2009) with rating 7.4
Danguard Ace 3 (2010) with rating 7.4


Donald F. Glut involved as actor and William Winckler involved as director in 13 films:
Gaiking I (2011) with rating 8.0
Starzinger II (2011) with rating 7.9
Starzinger III (2011) with rating 7.7
Gaiking II (2011) with rating 8.3
Danguard Ace 2 (2010) with rating 7.4
Space Pirate Captain Harlock 2 (2010) with rating 7.5
Gaiking III (2011) with rating 7.7
Starzinger (2011) with rating 7.9
Space Pirate Captain Harlock (2010) with rating 7.1
Kitaro's Graveyard Gang 2 (2011) with rating 7.3
Danguard Ace (2010) with rating 7.2
Kitaro's Graveyard Gang (2009) with rating 7.4
Danguard Ace 3 (2010) with rating 7.4


It took - 120.19 seconds
```