



مدرس: رامتین خسروی

مهلت تحویل:

جمعه ۱۰ مرداد ۱۴۰۴، ساعت ۲۳:۵۹



چی ببینم؟

هدف این پروژه، طراحی و پیاده‌سازی یک موتور پیشنهاد دهنده فیلم است که بتواند با استفاده از اطلاعات در دسترس از کاربر، فیلم‌هایی را پیشنهاد دهد که با علایق او هم‌خوانی بیشتری دارند. این سیستم می‌تواند به کاربران کمک کند تا تجربه‌ای شخصی‌سازی‌شده‌تر و لذت‌بخش‌تر از تماشای فیلم داشته باشند و از میان انبوه محتوا، سریع‌تر به گزینه‌ی دلخواه خود برسند.

شرح پروژه

فایل‌های ورودی

در این پروژه فایل‌های ورودی به شکل CSV هستند که به جزئیات آنها خواهیم پرداخت. آدرس این فایل‌ها از طریق آرگومان خط فرمان به برنامه ورودی داده می‌شوند. آرگومان اول آدرس فایل کاربران و آرگومان دوم آدرس فایل فیلم‌ها است.

پیکربندی نحوه اجرای برنامه

```
./ChiBebinam </path/to/users/file.csv> </path/to/films/file.csv>
```

فایل کاربران

در این فایل اطلاعات کاربرانی قرار دارد که در سیستم وجود دارند و برای آنها سابقه ثبت شده است. ستون Username مقادیر یکتا دارد. در مقادیر ستون Username کاراکتر فاصله¹ وجود ندارد. تضمین می‌شود که نام فیلم‌هایی که در سوابق کاربران استفاده می‌شود، در فایل فیلم‌ها وجود دارد. ستون امتیاز شامل یکی از کلمه‌های Average، Perfect و Poor می‌باشد. به تعداد فیلم‌هایی که در ستون Watched وجود دارد، در ستون Rating به ترتیب امتیاز ثبت شده است. در ستون‌هایی که بیش از یک داده داریم، مقادیر با سمی کالن (;) از هم جدا می‌شوند. توجه داشته باشید تمام کاربرانی که در این فایل وجود دارند، حداقل یک فیلم را دیده‌اند و سابقه خالی ندارند.

نمونه فایل کاربران

```
Username,Watched,Rating
kourosh,La La Land;Liar Liar;The Godfather,Perfect;Average;Poor
parisa,Twilight;La La Land;The Conjuring,Average;Poor;Perfect
```

¹ Space

فایل فیلم‌ها

این فایل شامل اطلاعات فیلم‌هایی است که در سامانه وجود دارند. تضمین می‌شود که فیلم تکراری در این فایل وجود ندارد. در ستون بازیگر فقط نام یک بازیگر وجود دارد. ستون Genre شامل یکی از کلمه‌های Horror و Drama، Comedy، Action، Science Fiction است. اعدادی که در ستون IMDb قرار می‌گیرند، عددی صحیح از ۰ تا ۱۰ هستند.

| نمونه فایل فیلم‌ها | | | | |
|--------------------|----------------------|---------------------|-----------------|------|
| Name | Director | Cast | Genre | IMDb |
| La La Land | Damien Chazelle | Ryan Gosling | Drama | 8 |
| Interstellar | Christopher Nolan | Matthew McConaughey | Science Fiction | 9 |
| Twilight | Catherine Hardwicke | Kristen Stewart | Horror | 5 |
| Inception | Christopher Nolan | Leonardo DiCaprio | Action | 9 |
| Parasite | Bong Joon ho | Song Kang ho | Drama | 9 |
| The Godfather | Francis Ford Coppola | Marlon Brando | Drama | 9 |
| The Big Lebowski | Joel Coen | Jeff Bridges | Comedy | 8 |
| Superbad | Greg Mottola | Jonah Hill | Comedy | 7 |
| Groundhog Day | Harold Ramiss | Bill Murray | Comedy | 8 |
| Liar Liar | Tom Shadyac | Jim Carrey | Comedy | 7 |
| The Mask | Chuck Russell | Jim Carrey | Comedy | 7 |
| Endgame | Scarlett Johansson | Joe Russo | Action | 8 |
| Lucy | Scarlett Johansson | Luc Besson | Action | 6 |
| The Conjuring | James Wan | Vera Farmiga | Horror | 7 |

دستورات

در این پروژه می‌خواهیم بر اساس فیلترهای مختلف به کاربر مورد نظر فیلم پیشنهاد دهیم. به طور کلی دو نوع کاربر برای این سیستم وجود دارد:

- کاربری که در سیستم سابقه فیلم دارد. بنابراین به کمک فیلم‌هایی که قبلاً دیده و امتیازی که برای آنها ثبت کرده، فیلم/فیلم‌های جدید به او پیشنهاد داده می‌شود.
- کاربری که در سیستم وجود ندارد. بنابراین داده‌ای نداریم که از آن برای تشخیص سلیقه کاربر استفاده کنیم و مجبوریم از داده‌های عمومی کمک بگیریم.

پیشنهاد بر اساس ژانر

به کمک این دستور کاربر ۳ فیلم از ژانری که انتخاب کرده را به عنوان پیشنهاد دریافت می‌کند. آرگومان اجباری برای این دستور Genre است. در صورتی که نام کاربری وارد نشود، یعنی برای کاربری که در سیستم سابقه‌ای ندارد، باید فیلم پیشنهاد شود. اگر نام کاربری مقدار داشته باشد، تضمین می‌شود که در فایل کاربران وجود دارد. تضمین می‌شود آرگومان Genre یکی از کلمات مجاز است. همچنین تضمین می‌شود به تعداد کافی فیلم وجود دارد.

| قالب ورودی |
|---|
| genre_recommandation "<Username>" "<Genre>" |

| قالب خروجی |
|---|
| 1. <Name1>: <Director1> (<IMDb1>) 2. <Name2>: <Director2> (<IMDb2>) 3. <Name3>: <Director3> (<IMDb3>) |

الگوریتم پیدا کردن فیلم مناسب

کاربر دارای سابقه

در صورتی که در سابقه فرد، فیلم/فیلم‌هایی از این ژانر وجود دارد، هیچکدام از آن فیلم‌ها نباید به عنوان پیشنهاد خروجی داده شوند. برای سایر فیلم‌های این ژانر، به کمک فرمول زیر یک *Score* محاسبه می‌کنیم.

$$Score = (IMDb * 0.7) + (DirectorEffect * 0.5) + (CastEffect * 0.6)$$

پارامترهای استفاده شده:

• پارامتر *DirectorEffect*:

○ اگر در سابقه کاربر فیلمی از این ژانر وجود دارد که کارگردان آن با فیلمی که در حال محاسبه

عدد آن هستیم یکسان است، بر اساس امتیازی که کاربر ثبت کرده، مقدار داده می‌شود:

■ امتیاز Perfect: پارامتر *DirectorEffect* برابر ۱۰

■ امتیاز Average: پارامتر *DirectorEffect* برابر ۵

■ امتیاز Poor: پارامتر *DirectorEffect* برابر -۱

○ در غیر این صورت: پارامتر *DirectorEffect* برابر صفر خواهد بود.

• پارامتر *CastEffect*:

○ اگر در سابقه کاربر فیلمی از این ژانر وجود دارد که بازیگر آن با بازیگر فیلمی که در حال

محاسبه عدد آن هستیم یکسان است، بر اساس امتیازی که کاربر ثبت کرده، مقدار داده

می‌شود:

■ امتیاز Perfect: پارامتر $CastEffect$ برابر ۷

■ امتیاز Average: پارامتر $CastEffect$ برابر ۴

■ امتیاز Poor: پارامتر $CastEffect$ برابر ۲-

○ در غیر این صورت پارامتر $CastEffect$ برابر صفر خواهد بود.

پس از محاسبه، فیلم‌ها را بر اساس $Score$ به شکل نزولی مرتب می‌کنیم. در صورتی که $Score$ چند فیلم یکسان شد، بر اساس IMDb بیشتر و اگر باز هم یکسان داشتند بر اساس حروف الفبای نام فیلم (به صورت صعودی) مرتب‌سازی را انجام دهید. در نهایت ۳ فیلم اول را خروجی دهید.

کاربر بدون سابقه

برای هر فیلم یک $Score$ محاسبه می‌کنیم. برای این کار از سابقه کاربران موجود در سیستم کمک می‌گیریم. برای تمام فیلم‌های این ژانر از فرمول زیر استفاده می‌کنیم:

$$Score = (P * 0.7) + (A * 0.5) + (Z * (-0.2))$$

پارامترهای استفاده شده در فرمول:

- پارامتر P : تعداد کاربرانی که این فیلم را دیده‌اند و به آن امتیاز Perfect داده‌اند
- پارامتر A : تعداد کاربرانی که این فیلم را دیده‌اند و به آن امتیاز Average داده‌اند
- پارامتر Z : تعداد کاربرانی که این فیلم را دیده‌اند و به آن امتیاز Poor داده‌اند.

پس از محاسبه، فیلم‌ها را بر اساس $Score$ به شکل نزولی مرتب می‌کنیم. در صورتی که $Score$ چند فیلم یکسان شد، بر اساس IMDb بیشتر و اگر باز هم یکسان داشتند بر اساس حروف الفبای نام فیلم (به صورت صعودی) مرتب‌سازی را انجام دهید. در نهایت ۳ فیلم اول را خروجی دهید.

| نمونه ورودی ۱ |
|---|
| genre_recommandation "kourosh" "Comedy" |

| نمونه خروجی ۱ |
|---|
| 1. The Mask: Chuck Russell (7) 2. Groundhog Day: Harold Ramiss (8) 3. The Big Lebowski: Joel Coen (8) |

توضیح مثال

کروش از کاربرانی است که در سیستم سابقه دارد. او فیلم Liar Liar را قبلاً دیده است. بنابراین نباید score برای این فیلم محاسبه شود. برای سایر فیلم‌های کم‌دقیق مقدار score را محاسبه می‌کنیم:

- The Big Lebowski: $score = (8 * 0.7) + (0) + (0) = 5.6$
- Superbad: $score = (7 * 0.7) + (0) + (0) = 4.9$
- Groundhog Day: $score = (8 * 0.7) + (0) + (0) = 5.6$

- The Mask: $score = (7 * 0.7) + (0) + (4 * 0.6) = 7.3$

پس از مرتب‌سازی به ترتیب زیر می‌رسیم:

1. The Mask
2. Groundhog Day
3. The Big Lebowski
4. Superbad

| نمونه ورودی ۲ |
|------------------------------|
| genre_recommandation "Drama" |

| نمونه خروجی ۲ |
|---|
| <ol style="list-style-type: none"> 1. La La Land: Damien Chazelle (8) 2. Parasite: Bong Joon ho (9) 3. The Godfather: Francis Ford Coppola (9) |

توضیح مثال

چون در این دستور نام کاربری وارد نشده، باید از الگوریتم کاربر بدون سابقه استفاده کنیم. مقدار score را برای تمام فیلم‌های این ژانر حساب می‌کنیم:

- La La Land: $score = (1 * 0.7) + (0) + (1 * (-0.2)) = 0.5$
- Parasite: $score = (0) + (0) + (0) = 0$
- The Godfather: $score = (0) + (0) + (1 * (-0.2)) = -0.2$

پس از مرتب‌سازی به ترتیب زیر می‌رسیم:

1. La La Land
2. Parasite
3. The Godfather

پیشنهاد بر اساس بازیگر

به کمک این دستور کاربر ۲ فیلم به عنوان پیشنهاد دریافت می‌کند که بازیگری که انتخاب کرده، در آن فیلم‌ها حضور داشته است. آرگومان اجباری برای این دستور Cast است. در صورتی که نام کاربری وارد نشود، یعنی برای کاربری که در سیستم سابقه‌ای ندارد، باید فیلم پیشنهاد شود. اگر نام کاربری مقدار داشته باشد، تضمین می‌شود که در فایل کاربران وجود دارد. در صورتی که فقط یک فیلم پیدا شد، اطلاعات همان یک فیلم را خروجی دهید و در صورتی که فیلمی پیدا نشد، پیام "No suitable movies were found." چاپ کنید.

قالب ورودی

```
cast_recommandation "<Username>" "<Cast>"
```

قالب خروجی

```
1. <Name1>: <Director1> (<IMDb1>)
2. <Name2>: <Director2> (<IMDb2>)
| 1. <Name>: <Director> (<IMDb>)
| No suitable movies were found.
```

الگوریتم پیدا کردن فیلم مناسب

کاربر دارای سابقه

در صورتی که در سابقه فرد، فیلم/فیلم‌هایی از این بازیگر وجود دارد، هیچ کدام از آن فیلم‌ها نباید به عنوان پیشنهاد خروجی داده شوند. بر اساس سابقه کاربر ابتدا باید ژانر مورد علاقه کاربر را پیدا کنیم. ژانری که کاربر بیشترین تعداد فیلم را از آن دیده، ژانر مورد علاقه کاربر است. توجه داشته باشید تضمین می‌شود که تعداد فیلم‌هایی که کاربر از ژانرها دیده باهم برابر نیستند. سپس فیلم‌های ژانر مورد علاقه کاربر که این بازیگر در آن حضور دارد را بر اساس IMDb به شکل نزولی مرتب کنید و اگر IMDb یکسان داشتند بر اساس حروف الفبای نام فیلم (به صورت صعودی) مرتب‌سازی را انجام دهید. در نهایت ۲ فیلم اول را خروجی دهید (در صورتی که فقط فیلم پیدا شد، همان یک فیلم را خروجی دهید).

کاربر بدون سابقه

تمام فیلم‌های این بازیگر را بر اساس IMDb به شکل نزولی مرتب کنید و اگر IMDb یکسان داشتند بر اساس حروف الفبای نام فیلم (به صورت صعودی) مرتب‌سازی را انجام دهید. در نهایت ۲ فیلم اول را خروجی دهید (در صورتی که فقط فیلم پیدا شد، همان یک فیلم را خروجی دهید).

| نمونه ورودی ۱ |
|---|
| cast_recommandation "kourosh" "Tom Hanks" |

| نمونه خروجی ۱ |
|--------------------------------|
| No suitable movies were found. |

توضیح مثال

طبق فیلم‌هایی که کوروش قبلاً دیده است، ژانر مورد علاقه او Drama است. در بین فیلم‌های این ژانر، Tom Hanks در هیچ‌کدام حضور نداشته است.

| نمونه ورودی ۲ |
|--|
| cast_recommandation "Scarlett Johansson" |

| نمونه خروجی ۲ |
|--|
| <ol style="list-style-type: none"> 1. Endgame: Joe Russo (8) 2. Lucy: Luc Besson (6) |

توضیح مثال

تمام فیلم‌هایی که این بازیگر در آنها حضور داشته، شامل Lucy و Endgame می‌شود که پس از مرتب‌سازی به ترتیب زیر می‌رسیم:

1. Endgame
2. Lucy

نکات و نحوه تحویل

- برای تحویل این پروژه یک مخزن² خصوصی در گیت‌هاب ایجاد کنید و فایل‌های خود را در آن قرار دهید. اکانت @AP-UT را به مخزن خود اضافه کنید. پس از اتمام پیاده‌سازی هش³ آخرین کامیت⁴ خود به همراه آدرس مخزن را به ایمیل صورت پروژه ریپلای کنید.
 - توجه داشته باشید پس از ارسال ایمیل امکان تغییر آخرین کامیت وجود ندارد و پروژه شما با همان هش ایمیل شده سنجیده خواهد شد.
 - دقت کنید که فایل Makefile باید در صفحه اول مخزن باشد و در پوشه‌ای قرار نداشته باشد.
 - نام پرونده قابل اجرای شما باید ChiBebinam باشد و پس از ساخته شدن در کنار Makefile قرار بگیرد (داخل پوشه‌ای فایل خروجی ساخته شده را قرار ندهید).
 - توجه داشته باشید مواردی که در صورت پروژه ذکر نشده در آزمون‌های خودکار سنجیده نمی‌شوند.
 - هدف از این پروژه انتخاب دستیاران آموزشی جدید آموزشی است، لطفا پروژه را خودتان انجام دهید. در صورت کشف تقلب از فرایند انتخاب حذف خواهید شد.
 - در صورتی که ابهامی وجود داشت و فرضی انجام دادید، به صورت کامنت در کد وارد کنید؛ در صورتی که مشکل خاصی وجود داشت می‌توانید با ما در ارتباط باشید.
- kouroshalinaghi@gmail.com
 - parisa.yahyapour.fatideh@gmail.com

² Repository

³ Hash

⁴ Commit