

Code ▾

Рекомендательная система

Команда 15: Горохова Анастасия, Чечёткин Макар, Писковский Роман, Мигунов Юрий, Гильмутдинова Алина

- Начало работы
- Анализ других рекомендательных систем
- Описание данных
- Эксплораторный анализ
 - Сетевой анализ
 - Регрессионный Анализ
 - Текстовый анализ
- Рекомендательные системы
 - Content-Based система
 - Коллаборативная система
 - Overview-based система
 - Общий вид
 - Примеры работы
 - Оценка работы систем
- Q&A секция
- Источники, которые мы использовали как примеры рекомендательных систем

Начало работы

Цель проекта: создать систему, способную рекомендовать фильмы:

- На основе преференций (content-based система)
- На основе ранее понравившихся фильмов (коллаборативная система)
- На основе описания фильма (overview-based система)

Входные данные:

Два датасета: один из них содержит все данные о фильме: язык, студия, каст, etc; второй датасет содержит в себе оценки фильмов пользователями.

Наш план работы выглядел таким образом:

0. Анализ других рекомендательных систем
1. Описание данных
2. Эксплораторный анализ данных, выбор переменных, на основе которого будут создаваться системы
3. Создание рекомендательных систем и их проверка
4. Оценка полученных рекомендаций
5. Создание симулятивной модели
6. Сбор всей работы в общий отчет.

Анализ других рекомендательных систем

Перед началом работы мы проанализировали рекомендательные системы популярных платформ

(kinopoisk, okko, ivi, etc.) чтобы выяснить, какие переменные используются в них.

Все системы имеют примерно одинаковое наполнение. Основные переменные: жанр, страна-производитель, рейтинг, длительность, ключевые слова, возрастной рейтинг (От сайта к сайту, конечно, наполнение может разниться).

Для разных типов систем необходимы разные данные. Мы выбрали следующие:

[Code](#)

[Code](#)

Name	System
Жанр	content-based
Страна	content-based
Длительность фильма	content-based
Возрастной рейтинг	content-based
Оценка фильма	content-based, overview-based
Оценки пользователей	коллаборативная система
Описание фильма	Overview-based

Описание данных

Перед непосредственным анализом, данные необходимо почистить и добавить новые, если необходимо.

Помимо дефолтных датасетов, мы решили добавить еще информацию о **возрастных ограничениях** фильмов. Для этого мы использовали два датасета, взятых с habr.com.

Далее, мы удалили N/A значения, а также удалили ненужные переменные. В итоговом датасете остались следующие переменные:

[Code](#)

[Code](#)

Variable_Name	Description	Variable_Type
genres	жанр фильма	Character
overview	описание фильма	Character
MPAA_rating	возрастной рейтинг (Система MPAA)	Nominal
runtime	длительность фильма	Interval
Rating	рейтинг фильма	Ratio

Variable_Name	Description	Variable_Type
popularity	популярность фильма	Ratio
production_countries	страны, которые сняли фильм	Character

Эксплораторный анализ

Эксплораторный анализ позволяет нам сделать brief overview полученной даты, выявить паттерны и зависимости, которые помогут в создании точной рекомендательной системы.

Кроме того, у нас есть несколько гипотез, которые поможет проверить эксплораторный анализ:

1. Существует связь между разными категориями переменных, что может влиять на точность рекомендательной системы (Например, при указании нескольких жанров авторами фильма)
2. Мы также предполагаем, что другие переменные (например, страна или жанр) могут повлиять на рейтинг. Причем, степени влияния отдельных переменных и переменных в совокупности могут различаться
3. В то же время, сеть по фильмам будет четко кластеризоваться и модулярность будет высокой – такую переменную можно смело включать в нашу content-based систему.

Эксплораторный анализ был проведен тремя методами:

1. Сетевой анализ
2. Регрессионный анализ
3. Текстовый анализ

Сетевой анализ

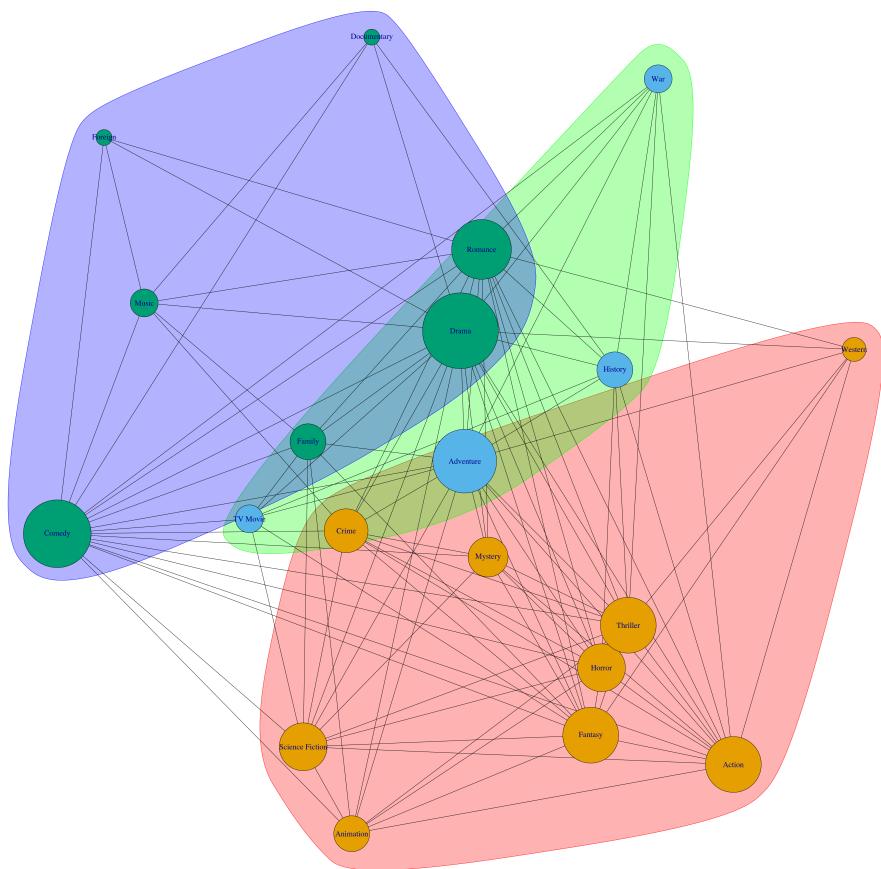
Первый этап отбора данных для content-based системы – сетевой анализ. Он позволит узнать, есть ли связь между категориями. Читабельные сети могут быть построены только по категориальным данным. Просмотрев рекомендательные системы на крупных сайтах и проанализировав наш датасет, мы пришли к выводу, что обязательно нужно проверить страны и жанры, так как они присутствовали в большинстве рекомендательных систем. Напомним, что мы будем смотреть на четкость получившихся кластеров и модулярность (чем выше модулярность и четче кластеры – тем лучше для нашей рекомендательной системы)

Начнем с жанра

Сначала рассмотрим, как связаны сами жанры между собой. Напоминаем, что наша гипотеза заключается в том, что если категории переменной тесно связаны между собой, то система не будет 100% точной. Почему? Например, если человек ищет фильм в жанре “Драма”, то может случиться такое, что ему выпадет фильм, где вторым жанром указана “Комедия”

Строим сеть по жанрам, где вершинами будут *жанры*, а связями - *фильмы*. Размер вершин задается *degree*

Code



Сеть показывает, что связь между жанрами существует, и довольно сильная, потому в рекомендации могут быть неточности по жанрам (когда просишь драму, а тебе выдают боевик). Так получается из-за особенностей датасета: на один фильм приходится два-три жанра. Потому неточности в рекомендациях зависят не от системы, а от датасета.

Но можно ли это как-то поправить?

Мы видим, что сеть разбилась на три кластера, причем, эти кластеры перекрывают друг друга, что, опять же, говорит о довольно тесных связях между вершинами. Возможно, можно попросить пользователя ввести несколько жанров, ведь они разбились на кластеры (в таком случае будет проще определить “сообщество” фильмов). Вопрос лишь в том, точна ли эта кластеризация

Посчитаем модулярность.

Code

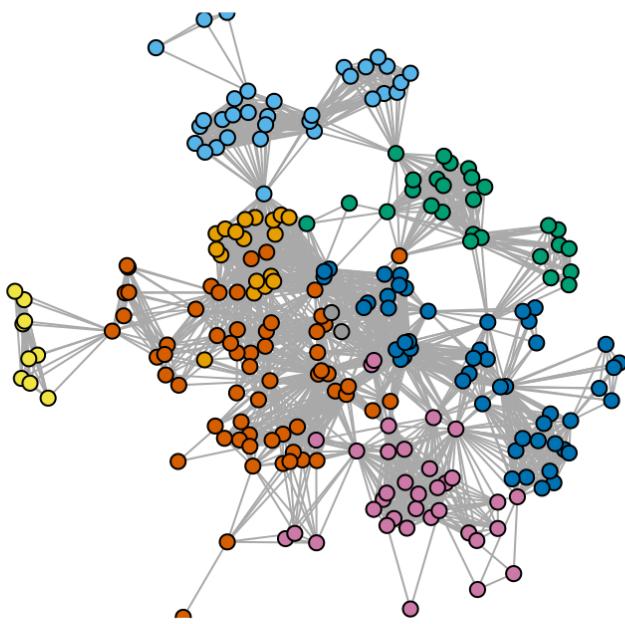
```
## [1] 0.09606481
```

модулярность очень низкая, что говорит о не очень высокой точности кластеризации. Таким образом, фильмы хоть и тесно связаны между собой, не имеют достаточной логики для кластеризации, то есть, выбирать несколько жанров для более точной рекомендации нет особого смысла

Но есть ли смысл вообще брать жанры как переменную для рекомендации?

Построим кластерную карту фильмов, где вершины будут фильмами, а связи - жанрами.

Code



Выделилось довольно много кластеров, визуально почти все они имеют довольно четкие границы.
Посмотрим на модулярность

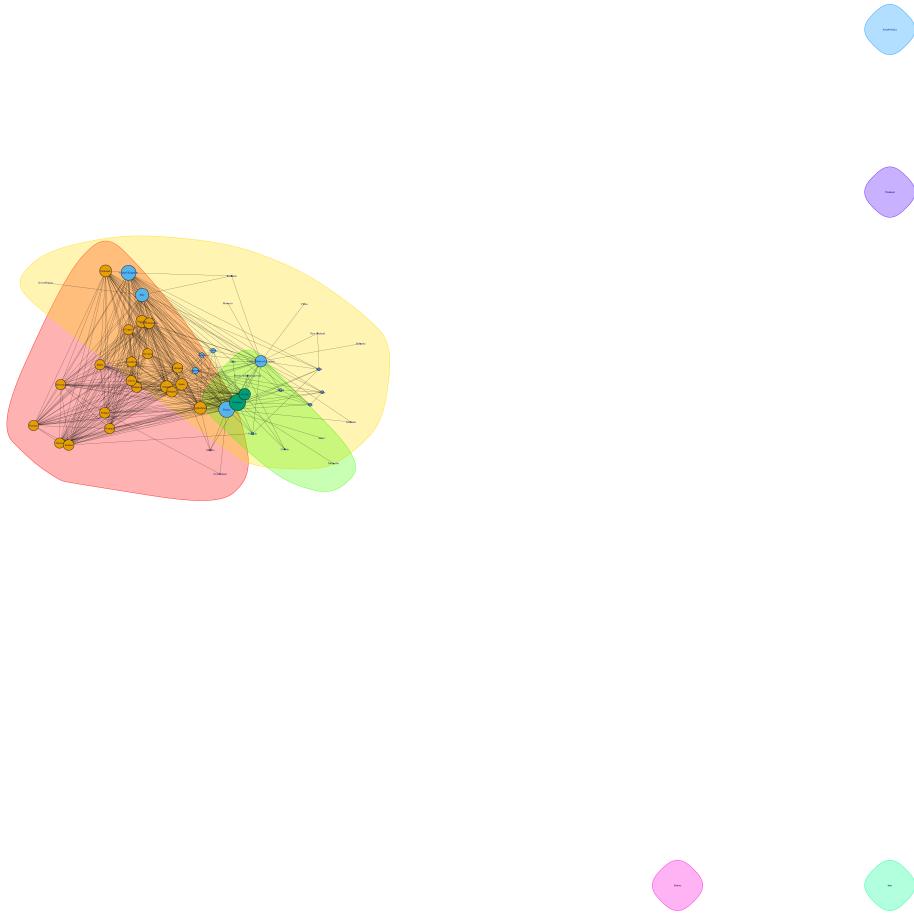
Code

```
## [1] 0.586539
```

В целом, значение намного выше, нежели в случае, когда связями были фильмы, а вершинами – жанры. К тому же, жанры – один из самых главных критериев при выборе фильма, поэтому мы оставляем жанры как критерий для нашей будущей content-based системы.

Очень часто в рекомендательных системах используют *страны*. Давайте проверим, выделяются ли кластеры по странам. Как и с жанрами, сначала построим кластерную карту, где вершины будут странами, а связи – фильмами.

Code



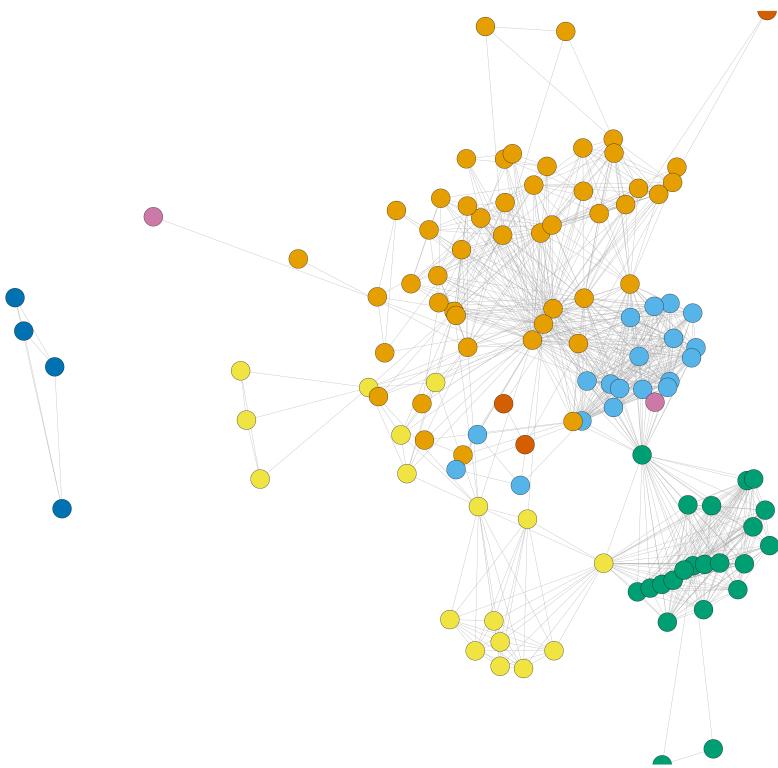
Code

```
## [1] 0.1567641
```

Ситуация та же, что и с жанрами: низкая модульность и кластеры, которые накладываются друг на друга. Связь есть, и довольно сильная (отсюда возможная неточность в рекомендации, и все еще по вине датасета), но сама кластеризация доверия не вызывает (смысла в вводе нескольких стран нет).

Построим сеть по фильмам, где страны станут линками:

Code



В целом, кластеры довольно четкие, это значит, что переменная нам подходит. Посмотрим на модулярность

Code

```
## [1] 0.4955914
```

Довольно высокое значение – разделению можно доверять.

Вывод

По результатам сетевого анализа и страны, и жанры подходят как переменные. Также в ходе анализа подтвердились гипотезы №1 (кластерные карты по жанрам и по картам показали, что есть связи существуют, к тому же, можно выделить паттерны и посмотреть, какие комбинации жанров используются чаще всего и какие страны чаще сотрудничают друг с другом) и №3 (как мы и ожидали, сети по фильмам в обоих случаях разбились на четкие кластеры, это значит, что рекомендация по похожим признакам будет релевантна в случае с жанрами и странами)

Регрессионный Анализ

Мы предположили, что если переменная влияет на рейтинг, то, значит, люди обращают внимание на нее при выборе фильма, а это значит, что переменную можно включить в content-based систему.

Code

Чтобы определить, влияют ли выбранные нами переменные на рейтинг, проведем статистические тесты (данные заранее почищены и приведены к нужному формату, все условия выполнены)

Code

```
##  
## Pearson's Chi-squared test  
##  
## data: df_reg$vote_average and df_reg$genres  
## X-squared = 13971, df = 9765, p-value < 2.2e-16
```

[Code](#)

```
##  
## Pearson's Chi-squared test  
##  
## data: df_reg$vote_average and df_reg$countries  
## X-squared = 2173.6, df = 2160, p-value = 0.4141
```

[Code](#)

```
##  
## Pearson's Chi-squared test  
##  
## data: df_reg$vote_average and df_reg$MPAA_Rating  
## X-squared = 540.64, df = 360, p-value = 2.102e-09
```

[Code](#)

```
##  
## Spearman's rank correlation rho  
##  
## data: df_reg$vote_average and df_reg$runtime  
## S = 42915000, p-value = 0.001304  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho  
## 0.1244226
```

[Code](#)

```
##  
## Spearman's rank correlation rho  
##  
## data: df_reg$vote_average and df_reg$popularity  
## S = 30919000, p-value < 2.2e-16  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho  
## 0.3691662
```

Смотрим на p-value в каждом teste. Если оно меньше 0.01, значит, существует статистически значимая

разница между наблюдениями, и, соответственно, влияние этой переменной на рейтинг. По результатам тестов нам не подходят только страны (они практически не влияют на рейтинг)

Но мы предположили, что в совокупности эти переменные влияют на рейтинг по-разному. Для того, чтобы проверить это мы построим 4 регрессии с разными предикторами, а после сравнить их с помощью статистического теста.

Построим следующие модели: 1. С длительностью (т.к. это числовая переменная и имеет высокий коэффициент корреляции с рейтингом) 2. С длительностью и жанром 3. С длительностью, жанром и возвратным рейтингом 4. С длительностью, жанром, возвратным рейтингом и популярностью

Перед тестом необходимо проверить все требования (проводить диагностику моделей), чтобы быть уверенными в точности нашего теста.

Требование 1: Мультиколлинеарность

[Code](#)

```
##           GVIF   Df GVIF^(1/(2*Df))
## runtime    2.23351     1      1.494493
## genres     2.23351    217     1.001853
```

[Code](#)

```
##           GVIF   Df GVIF^(1/(2*DF) )
## runtime    2.608339     1      1.615035
## genres     1714.451286  217     1.017307
## MPAA_Rating 867.314672    8      1.526286
```

[Code](#)

```
##           GVIF   Df GVIF^(1/(2*Df))
## runtime    2.659893     1      1.630918
## genres     4990.217657  217     1.019814
## MPAA_Rating 1015.544005    8      1.541412
## popularity  3.927917     1      1.981897
```

Во всех случаях показатель GVIF менее 10, значит требование выполняется

Требование 2: Аутлаеры

[Code](#)

```
## No Studentized residuals with Bonferonni p < 0.05
## Largest |rstudent|:
##          rstudent unadjusted p-value Bonferonni p
## 638 -3.775902        0.00017376      0.11555
```

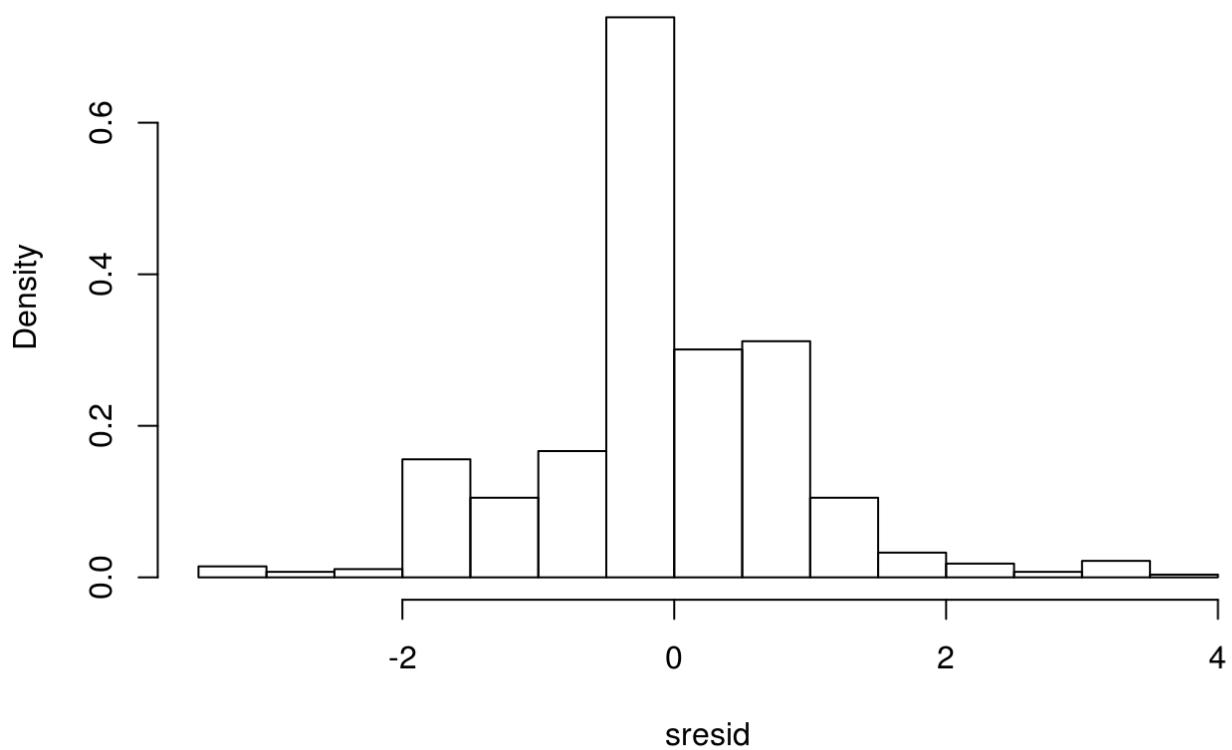
[Code](#)

В нашем случае выявились два аутлаера. Мы удалили их.

Требование 3: Studentized residuals

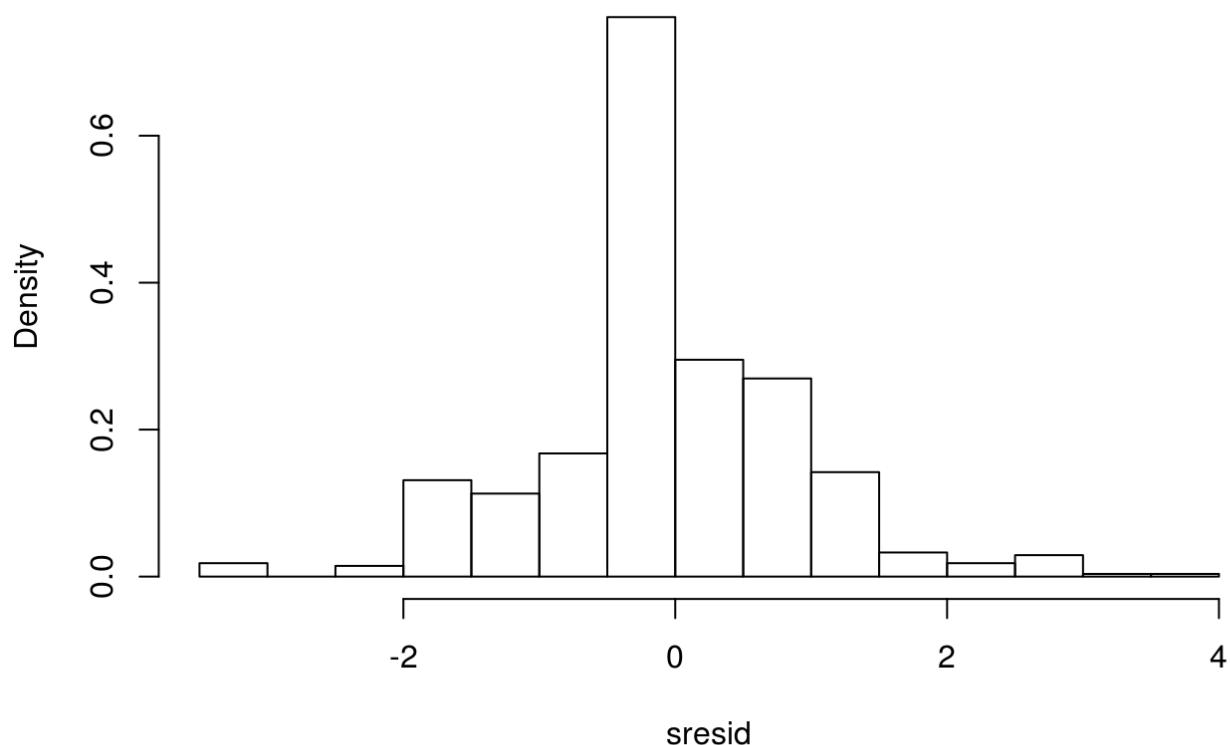
Code

Distribution of Studentized Residuals (model 2)

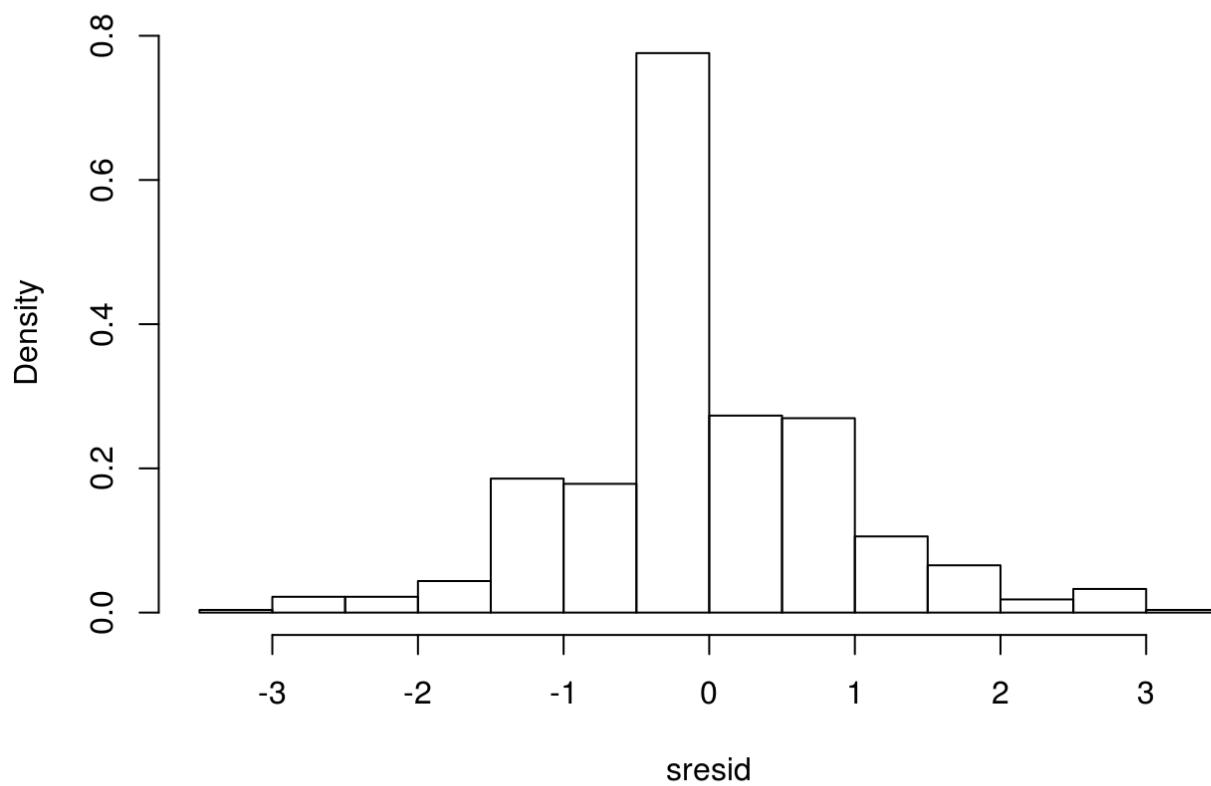


Code

Distribution of Studentized Residuals (model 3)

[Code](#)

Distribution of Studentized Residuals (model 4)



Распределение близко к нормальному. Учитывая, что в реальном мире практически нет нормально распределенных данных, нас устроит и так.

Требование 4: heteroscedasticity

[Code](#)

```
## Non-constant Variance Score Test  
## Variance formula: ~ fitted.values  
## Chisquare = 222.4272, Df = 1, p = < 2.22e-16
```

P-value is significant. Это плохо, попробуем исправить путем трансформации респонса.

После изменений снова проверяем:

[Code](#)

```
## Non-constant Variance Score Test  
## Variance formula: ~ fitted.values  
## Chisquare = 20.83333, Df = 1, p = 5.0103e-06
```

Стало получше, но p-value все еще значимое. Плохо. Тем не менее, 3 из 4 требований выполнены, потому тест сделать можно, просто во время интерпретации результатов помнить о том, что модели не очень точны

ANOVA Тест

Сравниваем модели с помощью ANOVA теста:

[Code](#)

```
## Analysis of Variance Table  
  
## Model 1: vote_average ~ runtime + genres  
## Model 2: vote_average ~ runtime + genres + MPAA_Rating  
## Model 3: vote_average ~ runtime + genres + MPAA_Rating + popularity  
##   Res.Df   RSS Df Sum of Sq    F    Pr(>F)  
## 1     439 305.45  
## 2     431 288.41  8    17.043  3.9908 0.0001399 ***  
## 3     430 229.54  1    58.873 110.2877 < 2.2e-16 ***  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Судя по ANOVA, наиболее точной является модель 2 (длительность, жанр, возрастной рейтинг), так как у нее меньше F-статистика и больше степеней свободы. По анове в принципе сложно выбрать лучшую модель, так как здесь работает принцип “Чем больше предикторов – тем лучше”. Судя по F-статистике, наши модели не имеют тенденции к улучшению, а модель 1 выбирать нецелесообразно, так как слишком мало предикторов, потому вторая модель выглядит наиболее подходящей.

Вывод

Основываясь на проведенном анализе, для content-based системы мы возьмем:

1. **Жанры** – они релевантны и с точки зрения анализа, и с точки зрения “жизни”. Жанр – основной критерий выбора фильмов
2. **MPAA рейтинг** – то же самое. Во-первых, очень часто возрастной рейтинг встречался в других рексистемах. Во-вторых, по рейтингу можно выбрать фильм под настроение или под компанию (хочешь боевик с насилием – ставь высокий рейтинг, хочешь фильм про любовь, дружбу и пони – ставь низкий рейтинг)
3. **Длительность фильма** – анализ доказал, что эта переменная релевантна. Также в других рексистемах пользователю предлагают выбрать длительность фильма. Мы считаем, что длительность – важный критерий, ведь не всегда людям хочется сидеть за фильмом долго.

Хоть сетевой анализ и показал, что *страны* могут быть релевантным критерием для системы, стат тест показал обратное. К тому же, мы столкнулись с проблемами, связанными с форматом данных: в фильмах указано несколько стран, что существенно снижает точность системы (как и в жанрах, но они, по нашему мнению, намного полезнее для рекомендательной системы). Мы попробуем добавить эту переменную в content-based систему, но как критерий она остается под вопросом.

К тому же, регрессионный анализ подтвердил нашу гипотезу №2: переменные действительно влияют на рейтинг как по отдельности, так и в совокупности (причем, влияние на рейтинг разное для разных комбинаций переменных)

Текстовый анализ

В наших планах – создание рекомендательной системы на основе схожести описаний. Такая система решила бы проблему холодного старта, ведь новый пользователь, который не имеет каких-то особых предпочтений, может ввести название любого понравившегося ранее фильма, и система выдаст список фильмов с похожим описанием.

Для того, чтобы воплотить нашу идею, необходимо провести текстовый анализ. Он покажет, насколько идея рекомендации фильмов по описанию релевантна, а также станет основой для нее.

Разобъем описание фильма (overview) на отдельные слова и удалим из них стоп-слова. В данном случае мы используем английский словарь стоп-слов. Посмотрим на топ-7 по частоте использования слов.

[Code](#)

words	Count
life	101
young	78
new	76
man	68
film	67
two	57
love	54

Далее, для того, чтобы предлагать фильмы на основе их описания, надо создать матрицу смежности. Также, чтобы система выдавала нам максимально схожие фильмы(а не те же фильмы, что вводим мы), заменим диагональные элементы с 1 на 0.

Посмотрим на матрицу смежности фильмов на разных отрезках. Как мы видим, уровень схожести заходит до 28%, поэтому можно использовать описания.

Code

```
## [1] 0.2407753 0.1174330 0.1516562 0.1865950 0.2799763 0.1568439 0.1231496
```

Рекомендательные системы

Content-Based система

Content-based система используется в том случае, если у пользователя, отсутствующего в системе есть *предпочтения* в выборе фильма. Предоставляется возможность указать предпочтения в *жанре, длительности и возрастном рейтинге*.

- **Жанр** мы считаем основополагающей характеристикой фильма, а любимые жанры у каждого пользователя разные, поэтому мы даем его указать.
- **Продолжительность фильма** было решено выбрать всвязи с тем, что на этапе эксплораторного анализа было выявлено, что фильмы продолжительностью около полутора часов имеют более высокий рейтинг, а в нашей системе происходит ранжирвоание в том числе и по рейтингу. Не давая выбор продолжительности, вероятно, рекомендация предлагала бы только фильмы длины около полутора часов.
- **Возрастной рейтинг** мы выбрали исходя из логических соображений о том, как пользователи могут выбирать фильмы. Например, человек желает посмотреть фильм с семьей, где есть маленькие дети, а значит он не хочет получить рекомендацию где были бы фильмы 18+, поэтому мы решили дать возможность выбирать возрастное ограничение.

Кроме того, если пользователь не определился, либо просто безразличен к какому-либо пункту для заполнения, он может ничего не вписывать - рекомендация будет выдана на основе **заполненных пунктов**, а пустые проигнорируются. (Примечание: несмотря на то, что страны были выявлены как хорошая переменная для включения в систему, их использование сильно сокращало итоговую выдачу в количестве фильмов. В комбинации с более приоритетными характеристиками, которые мы выбрали, в лучшем случае находилось 1-2 фильма. С большим датасетом этой проблемы бы не возникло, а страны имело бы смысл включить в фильтр.) —

Code

Code

Code

Code

```
##                               title      rating
## 1                      5 Card Stud 4.225000
## 2             The Dawn Patrol 3.576923
## 3        Hour of the Gun 3.461538
## 4            The Bunker 2.800000
## 5 Mad Max 2: The Road Warrior 2.384615
```

Коллаборативная система

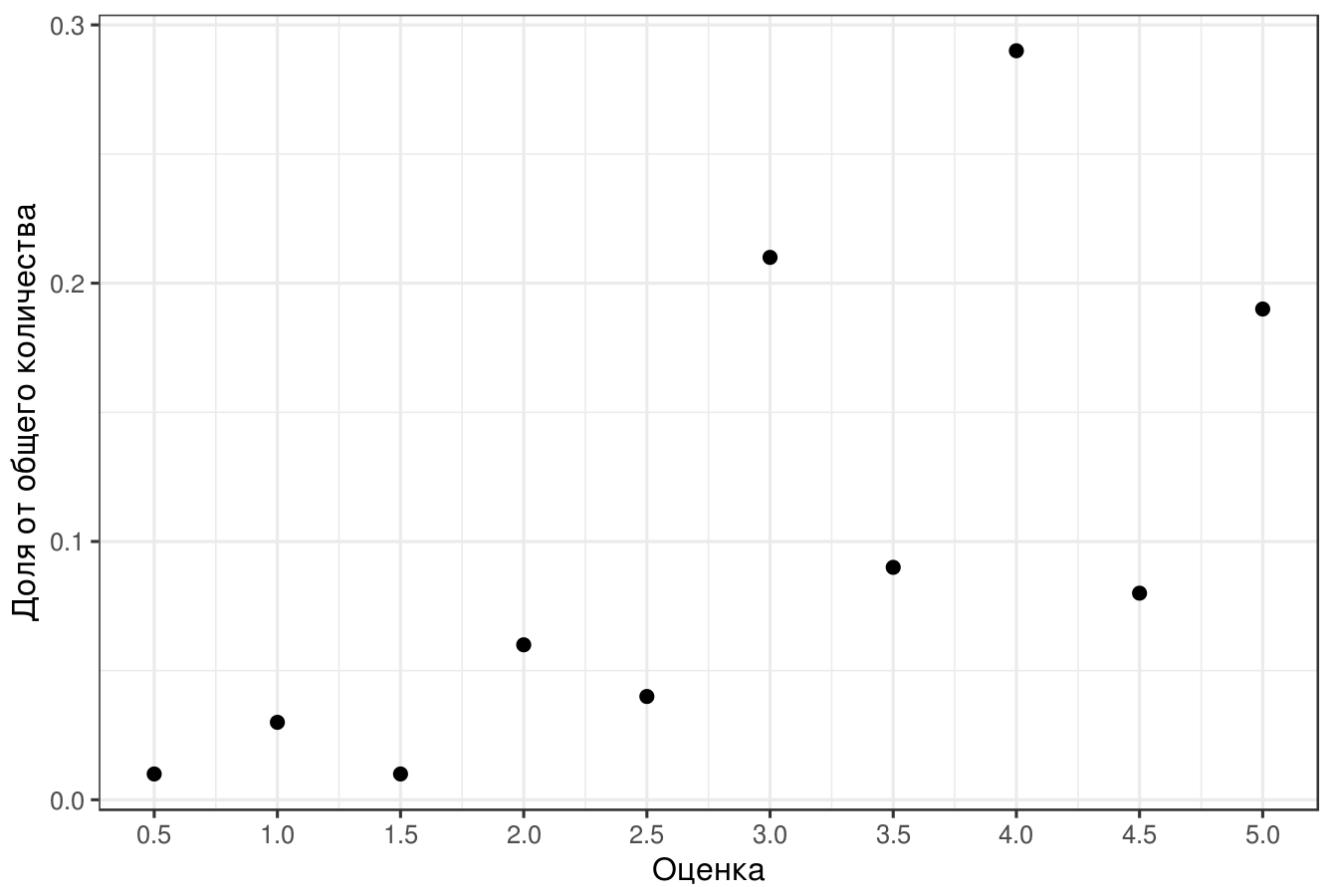
Данная система используется, если пользователь уже в системе и оценил некоторое количество фильмов.

Мы спрашиваем у пользователя его id и желаемое количество фильмов, после чего реализуем коллокративную систему по методу “IBCF”, так как данный метод направлен на схожесть элементов, а не пользователей, что является более точным подходом в данной ситуации. Кроме того, мы убрали фильмы с маленьким количеством оценок и пользователей, оценивших маленькое количество фильмов. При этом, если пользователь вводит айди, которого нет в базе данных, то он получит аутпут “Ваш профиль не найден в системе”.

Для начала создадим нового рандомного пользователя, чтобы реализовать систему на нем. Для этого определим соотношение оценок в исходном датасете, чтобы с таким же соотношением расставить оценки фильмов рандомному пользователю.

Code

Распределение оценок в исходных данных.



Как мы видим, положительные и нейтральные оценки (3.5 и выше) преобладают в исходных данных.

При этом больше всего фильмов оценены на 3 или 4 балла. Создадим *рандомного пользователя*, который оценит рандомное количество фильмов исходя из данного соотношения.

Создаем датафрейм с рейтингами пользователя, полученными рандомом. Соединяем его с общим датафреймом рейтингов. Посмотрим, какие фильмы оценил *рандомный пользователь* и какие оценки он им поставил(он оценил от 5 до 20 фильмов, но мы выведем его топ-5)

[Code](#)

movield	rating
2078	4.0
527	4.0
16	3.5
2115	3.5
2011	3.5

[Code](#)

Overview-based система

Система рекомендация по Overview используется в том случае, если у пользователя *нет предпочтений* для желаемого к просмотру фильма, из тех, которые предлагает выбрать Content Based система, но он хотел бы посмотреть что-то похожее на свой *любимый фильм*.

Пользователь может ввести свой *любимый фильм* и получить рекомендацию на основе схожести описаний. Как и в Content-Based системе ему будут предложены 5 фильмов проранжированные по рейтингу.

[Code](#)

[Code](#)

[Code](#)

```
## # A tibble: 5 x 2
##   title           rating
##   <chr>          <dbl>
## 1 Back to the Future Part II    3.69
## 2 Reign Over Me            3.54
## 3 Shiloh                  3.5 
## 4 Just a Question of Love   2.83
## 5 Back to the Future Part III  2.82
```

Общий вид

Мы попробовали представить, как бы все эти системы работали в совокупности.

Получилась следующая модель:

(system.jpg)

Принцип работы системы такой: Сначала пользователь указывает, есть ли у него аккаунт или он работает с нашим сайтом/системой/whatever впервые.

Если пользователь новый

Есть два пути:

1. У пользователя есть предпочтения для фильма. В таком случае он выбирает рекомендацию с помощью content-based системы, вводит свои предпочтения и получает список фильмов. Стоит отметить, что пользователь может заполнить не все поля, и система все равно будет работать
2. У пользователя нет предпочтений для фильма. В таком случае мы предлагаем ему выбрать фильм с помощью системы, основанной на описании фильма. Пользователь вводит название, например, любимого фильма, и система выдаст фильмы с похожим описанием. Таким образом мы решаем проблему холодного старта (пользователя нет в системе). Нам показалось, что просить пользователя оценить рандомные 10 фильмов не совсем целесообразно, ведь очень сложно угадать, какие фильмы пользователь смотрел, а какие – нет, к тому же, мы совершенно не знаем предпочтений пользователя (может, он не смотрит популярные фильмы, а предпочитает камерные авторские). Поэтому overview-based система показалась нам наиболее подходящим вариантом

Если пользователь есть в системе

Снова два пути:

1. У пользователя есть предпочтения для фильма. В таком случае он выбирает рекомендацию с помощью content-based системы, вводит свои предпочтения и получает список фильмов. Стоит отметить, что пользователь может заполнить не все поля, и система все равно будет работать.
2. У пользователя нет предпочтений для фильма. В таком случае он просто вводит свой айди, и система выдает рекомендацию основываясь на ранее оцененных фильмах.

Примеры работы

1) Я новый пользователь, мне нравится фильм Трудности Перевода, какой будет аутпут?

Code

```
## # A tibble: 5 x 2
##   title           rating
##   <chr>          <dbl>
## 1 Mothra vs. Godzilla    3.89
## 2 My Name Is Bruce     3.64
## 3 Nosferatu          3.48
## 4 Reclaim Your Brain  3.45
## 5 Amigo, Stay Away    3.33
```

2) Я новый пользователь, который отдает предпочтения комедиям, с рейтингом PG, длительность от часа до двух, какие фильмы будут мне рекомендованы?

[Code](#)

```
##                               title   rating
## 1           We're No Angels 4.357143
## 2           License to Wed 4.114130
## 3           Charlie's Angels 4.000000
## 4      Top of the Food Chain 3.625000
## 5 My Best Friend's Wedding 3.386364
```

3) Что если у пользователя любимый фильм это малоизвестный Титаник(movielid = 2699), что ему предложат? Будет ли известный Титаник в рекомендации?

[Code](#)

```
## # A tibble: 5 x 2
##   title             rating
##   <chr>            <dbl>
## 1 K-19: The Widowmaker 4.09
## 2 The Poseidon Adventure 3.96
## 3 Titanic            3.16
## 4 Titanic            3.16
## 5 My Tutor            3.16
```

4) Интересно посмотреть на работу системы, основанной на overview, поэтому пример для нее: если я выберу 11 друзей Оушена, будет ли мне рекомендовано что-то из этого: 48 часов, Крепкий орешек или Перевозчик

[Code](#)

```
## # A tibble: 5 x 2
##   title             rating
##   <chr>            <dbl>
## 1 Boat              4
## 2 Saw II            3.77
## 3 Casino            3.5
## 4 Taxi 4            3.37
## 5 Ocean's Twelve   3.21
```

5) Хотел бы воспользоваться content-based системой, в которой мне понравилась возможность хорошего фильтра для фильмов. Хотел бы узнать результат для "Drama", ">70", "R"

[Code](#)

```
##                               title   rating
## 1      A Woman, a Gun and a Noodle Shop 4.214286
## 2 Confession of a Child of the Century 4.154545
## 3                           Edward, My Son 4.142857
## 4           Wish You Were Here 3.750000
## 5             The Dawn Patrol 3.576923
```

6) Пример для системы по overview. Если пользователь предпочитает фильмы “Мстители”, “Человек-паук: Вдали от Дома” и “Доктор Стрэндж” будут ли ему рекомендованы подобные “супергеройские” фильмы или сиквелы этих фильмов (например, “Мстители: Финал” или “Капитан Америка”)?

Данных фильмов нет в базе, но теоретически должны рекомендоваться подобные фильмы. Единственное ограничение это если эти супергеройские фильмы не соответствуют “шаблону”. То есть являются совершенно новым направлением. Например сериал “The Boys”, он относится к супергеройской тематике, но описывает супергероев не с лучшей стороны где суперы злоупотребляют славой и безнаказанностью. Данный фильм/сериал возможно будет рекомендоваться но ближе к концу списка.

7) “Я новый пользователь и у меня есть следующие предпочтения: жанр - Drama, продолжительность - длинный, 16+”

Code

```
##                               title   rating
## 1                  Longitude 4.148649
## 2                   On Guard 3.825000
## 3                 The Mikado 3.807692
## 4 Mere Brother Ki Dulhan 3.421053
## 5        Visions of Europe 2.863636
```

8) Хочу проверить content-based систему, получу ли в рекомендации фильм Star Wars Return of Jedi, если введу жанр - Action, 150 минут и рейтинг - PG

Code

```
##                               title   rating
## 1 Mission: Impossible II 4.000000
## 2                      xXx 3.588235
## 3 Speed 2: Cruise Control 3.409091
## 4            Titanic 3.160000
## 5 The Day After Tomorrow 2.652174
```

Code

9) Меня нет в вашей системе (новый пользователь), мой любимый жанр - ужасы, длительность фильма ~ 1 час 40 минут, возрастное ограничение не указываю, какой будет ваша рекомендация?

Code

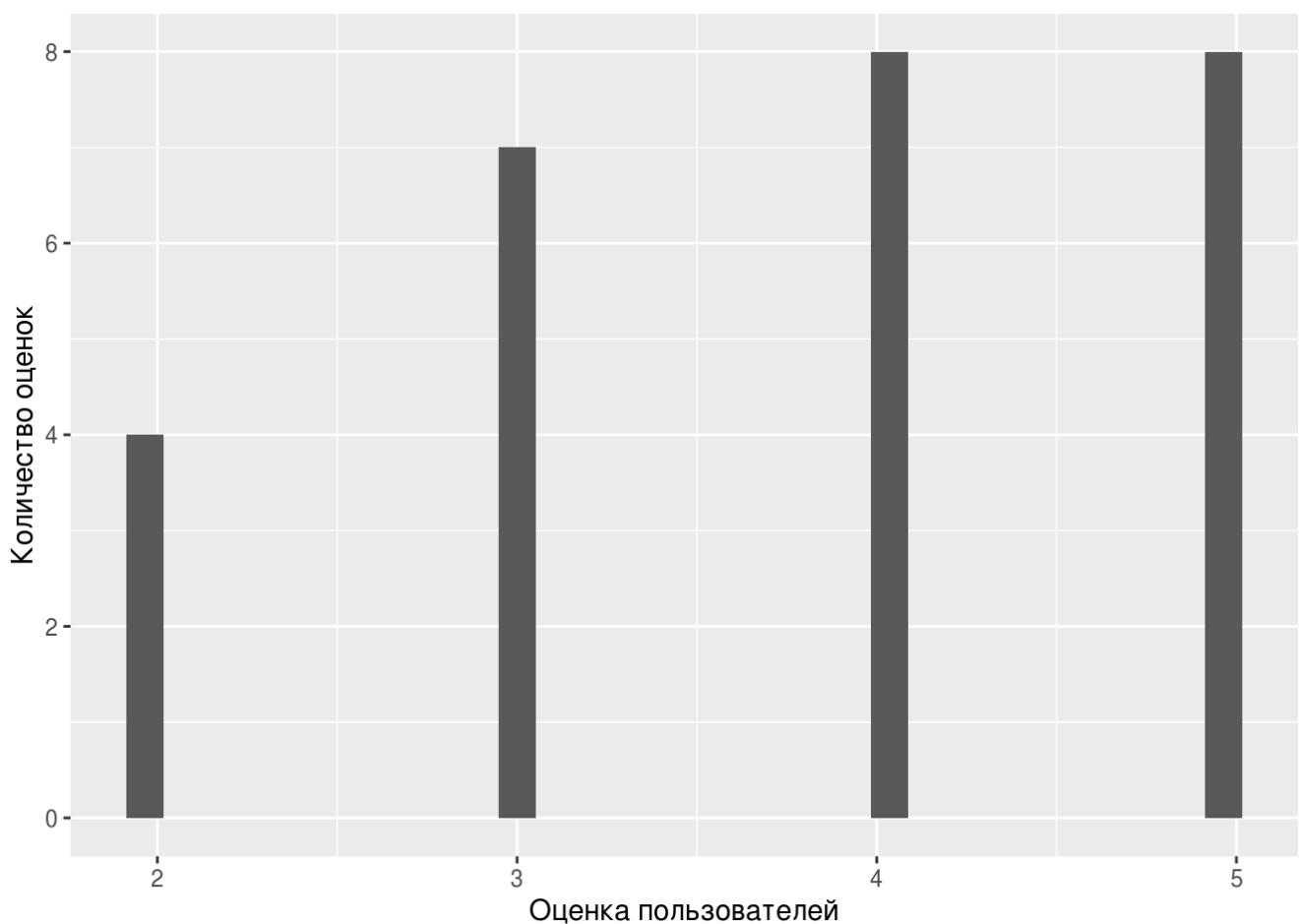
```
##           title      rating
## 1 Interview with the Vampire 3.875000
## 2 Dawn of the Dead 3.827586
## 3 Dawn of the Dead 3.744898
## 4 Silent Hill 3.636364
## 5 Rosemary's Baby 3.566667
```

Оценка работы системы

Оценим Content-Based систему Для ее оценки мы попросили у нескольких человек задать желаемые характеристики фильмов (жанр, длительность, возрастное ограничение) и выдали им рекомендацию используя нашу систему, затем попросили оценить полученный результат.

Как можно заметить, большинство опрошенных оценили рекомендацию положительно. Можно сделать вывод, что система работает неплохо.

[Code](#)



Теперь перейдем к коллаборативной системе и ее оценке

Сохраним только оценки пользователей и фильмы и найдем топ 5 фильмов какого-нибудь пользователя, например, 13937.

[Code](#)

```
## [1] 5 5 5 5 5
```

Проверим, есть ли у этого пользователя реальные оценки этих фильмов

Code

```
## # A tibble: 0 × 3
## # ... with 3 variables: userId <dbl>, movieId <dbl>, rating <dbl>
```

Реальных оценок нет. Поэтому выбираем фильмы, на которые оценки есть.

Берем топ-5 фильмов пользователя и смотрим на вероятность предсказания тех или иных жанров.

Code

```
## # A tibble: 15 × 3
##   genres_sep      n    prop
##   <fct>        <int>  <dbl>
## 1 Drama          8  0.216
## 2 Comedy         6  0.162
## 3 Animation      3  0.0811
## 4 Thriller       3  0.0811
## 5 Romance        3  0.0811
## 6 Science Fiction 2  0.0541
## 7 Family         2  0.0541
## 8 Adventure      2  0.0541
## 9 Fantasy        2  0.0541
## 10 Horror         1  0.0270
## 11 Action         1  0.0270
## 12 History        1  0.0270
## 13 Crime          1  0.0270
## 14 Documentary    1  0.0270
## 15 War            1  0.0270
```

Теперь посмотрим жанры предсказываемых фильмов от нашей системы. В целом, вероятность рекомендации фильмов тех жанров, что наиболее высоко оценил пользователь примерно сопоставима (Thriller и Drama находятся в топе в обоих случаях), а значит система работает неплохо, пусть и не выводит полный топ в соответствии с пользовательскими оценками.

Code

```
## # A tibble: 7 x 3
##   genres_sep     n    prop
##   <fct>      <int>  <dbl>
## 1 Thriller      3  0.231
## 2 Drama         3  0.231
## 3 Action        2  0.154
## 4 Crime         2  0.154
## 5 Romance       1  0.0769
## 6 Documentary   1  0.0769
## 7 Mystery       1  0.0769
```

Остается оценка системы по Overview

Оценка работы данной системы наиболее субъективна, но по итоговой выдаче рекомендаций можно сделать оценку ее работы.

Например, при запросе рекомендации по Back to the Future, система выдает 2 и 3 часть этого фильма, которые наверняка на него похожи.

Code

```
## # A tibble: 5 x 2
##   title           rating
##   <chr>          <dbl>
## 1 Back to the Future Part II  3.69
## 2 Reign Over Me            3.54
## 3 Shiloh                 3.5
## 4 Just a Question of Love  2.83
## 5 Back to the Future Part III 2.82
```

Если же попросить найти похожий фильм на Ocean's Eleven, то в рекомендации будет Ocean's Twelve - вторая часть этого фильма.

Code

```
## # A tibble: 5 x 2
##   title           rating
##   <chr>          <dbl>
## 1 Boat             4
## 2 Saw II          3.77
## 3 Casino           3.5
## 4 Taxi 4          3.37
## 5 Ocean's Twelve  3.21
```

Аналогично и со Star Wars, система выдает другую часть этого фильма.

Code

```
## # A tibble: 5 x 2
##   title                  rating
##   <chr>                 <dbl>
## 1 Asterix at the Olympic Games  4.07
## 2 Return of the Jedi           3.95
## 3 The Great Dictator         3.86
## 4 The Hidden Fortress        3.46
## 5 Die Hard 2                 3.37
```

Можно предположить, что система хорошо находит схожие описания фильмов.

Q&A секция

1) А какие дополнительные данные вы использовали?

Мы использовали датасет с возрастными рейтингами фильмов.

2) Что делать, если пользователю не важна продолжительность фильма в методе content-based?

Пользователь может не указывать продолжительность фильма, система все равно выдаст ему рекомендацию на основе двух других параметров.

3) “Не предлагаем рандомные фильмы, так как можно не попасть в интересы пользователя” - Велик ли шанс не попасть в его интересы, если предлагать топ фильмов по кол-ву высоких оценок?

Да, велик, потому что даже фильмы с самыми высокими оценками не покрывают все жанры. Например, в таком топе не будет хорроров. Кроме того, есть киноманы, которые наверняка видели большинство таких фильмов или же избирательные зрители, предпочитающие авторское кино.

4) Как именно с точки зрения кода реализуется коллаборативная система, основанная на overview? Каким образом там учитываются оценки уже существующих пользователей?

Данная система не является коллаборативной и предлагает фильмы исключительно на схожести описаний фильмов. Рейтинг в данном случае выводится исключительно для пользователя. Что бы он мог сам оценить стоит смотреть фильм или нет

5) Немного не понял, что значит в возникших проблемах “система не предполагает возвращения на шаг назад”. Для чего это нужно и как бы улучшило работу системы.

Возврат на шаг назад позволил бы пользователю поменять предпочтительные характеристики фильмов, а не прогонять всю систему с самого начала.

6) Можно ли в оценку по overview ввести несколько названий фильмов для получения более точной рекомендации? И можно ли ввести несколько жанров в content-based систему? Если нет, то добавление этого могло бы помочь улучшить качество рекомендаций, поскольку у пользователя может не быть четких предпочтений в жанре и ему тогда нужно дать возможность выбрать несколько вариантов.

Ввести несколько названий или жанров не представляется возможным, к тому же если попытаться увеличить кол-во фильмов для overview система может ничего не выдать так как пользователь(сам того не зная) введет абсолютно разные фильмы.

7) Какой метод построения коллаборативной фильтрации вы выбрали? IBCF или UBCF? И чем вы

руководствовались?

Мы выбрали метод IBCF, потому что данный метод выводит рекомендацию, основываясь на схожести фильмов, что в данной ситуации кажется более показательным, нежели схожесть пользователей. Так как у людей могут быть разные вкусы, несмотря на некоторые общие любимые фильмы.

8)Что произойдет, если пользователя нет в системе и он вводит только один из трех инпутов?

Рекомендация все равно сработает, только теперь сортировка идет по одному импути и значений фильмов может быть много.

9)В Content Based пользователь может указать только один жанр или есть возможность указать несколько?

Есть возможность указать только один жанр.

10)Как вы разделили длительность фильмов? Что будет видеть пользователь, когда ему нужно будет выбрать продолжительность фильма?

Мы разделили фильмы на 3 категории: менее 1.5 часов, менее 2 часов и более 2 часов. Пользователю предстоит указать желаемый интервал для рекомендации.

11) Мне показалось странным, что в рекомендациях, которые выпали для Star Wars появился фильм Asterix, при этом, он стоит на 1-м месте и выше, чем фильм Return of Jedi. С чем это может быть связано?

Нам тоже показалось это странным, но при детальном разборе и прочтении описания двух фильмов было выяснено что присутствуют такие слова как “Империя”/“Император”, В одном фильме это галактическая, а в другом - римская империя. Или другой пример: “Сопротивление”, в одном случае имеются в виду повстанцы, а в другом деревня Астерикса.

12) Почему вы взяли длительность, жанр, рейтинг и популярность в регрессионный анализ?

Проверялись ли остальные переменные? Если да, то почему их отсекли

Были взяты эти переменные исходя из субъективных суждений. Кроме того, другие возможные переменные имели неподходящий формат.

13) Проблема холодного старта, не вижу смысла брать в критерии продолжительность фильма, наверное имеет смысл только поделить фильмы на короткометражки и полнометражки, ведь обычно когда человек садится смотреть фильм он готов посидеть от полутора до 2 часов.

По нашему опыту не всегда человек готов сидеть от полутора до двух часов. Многие пользователи “заполняют” свободное время между работой и учебой или используют время в пути до работы/учебы /другого места назначения. Именно тогда возникает средний промежуток времени между короткометражками и полнометражками.

14) В системе content based - уверены ли вы во внедрении возрастного ограничения? Пользователь сам вводит его в инпуте?

Возрастное ограничение важно, чтобы пользователю не выпали фильмы, которые он не должен видеть исходя из своего возраста. Например, семья выбирает фильмы, который хочет посмотреть вместе с детьми. Для этого пользователь сам вводит данный параметр.

15) а если пользователь оценил все фильмы на сайте на 1-2 балла (нет хороших оценок), как система будет рекомендовать? если пользователь посмотрел совсем мало фильмов (1), для него актуальнее колаборативная или content-based рекомендация?

Такому пользователю предпочтительнее выбрать Content-Based систему, потому что порекомендовать фильм, который ему бы понравился на основе оцененных на 1-2 – невозможно. В Content-Based системе он уже сможет указать желаемые характеристики фильма.

Источники, которые мы использовали как примеры рекомендательных систем

1. <https://meduza.io/cards/ne-mogu-pridumat-kakoy-film-posmotret-chto-delat> (<https://meduza.io/cards/ne-mogu-pridumat-kakoy-film-posmotret-chto-delat>)
2. <https://www.netflix.com/ru/> (<https://www.netflix.com/ru/>)
3. <https://premier.one> (<https://premier.one>)
4. <https://www.ivi.ru> (<https://www.ivi.ru>)
5. <https://okko.tv> (<https://okko.tv>)
6. <https://www.kinopoisk.ru> (<https://www.kinopoisk.ru>)