ATLANTIS PRESS

Advances in Economics, Business and Management Research, volume 138

2nd International Scientific and Practical Conference "Modern Management Trends and the Digital Economy: from Regional Development to Global Economic Growth" (MTDE 2020)

Designing a Digital Content Recommendation System for films

Zagranovskaia A.V.* Mitiura D.Yu. Makarchuk T.A.

St. Petersburg State University of Economics, St. Petersburg, Russia
**Corresponding author. Email: zagranet@rambler.ru*

ABSTRACT

Purpose of the study. The purpose of the study is to analyze the existing methods for constructing content recommendation systems and developing the most accurate and adequate content recommendation system for films. Materials and methods. The paper considers a variety of data analysis methods: vectorization of descriptions, latent semantic analysis (LSA) and its probabilistic form (pLSA), latent Dirichlet allocation (LDA), proximity metrics (cosine divergence, Jensen-Shannon divergence, Kullback-Leibler divergence), the most common factorization method is truncated SVD. Designed content recommendation systems are tested on MovieLens dataset open data [1]. This opens up opportunities for checking the results obtained and improving the proposed models. Results. As a result, several models of content recommendation systems for films were developed and reviews of potential users of the system were analyzed which allowed determining the best version of the content recommendation system. Conclusion. An analysis of the created recommendation systems made it possible to understand the user requirements for them. The most important criterion was the level of trust to the system. In other words, this is how much the user is sure that he or she will really like offered recommendations. The quality of the recommendations was evaluated using a survey of the system potential users. This made it possible to cover a number of criteria for evaluating recommendation systems, such as accuracy of rating prediction, novelty, surprise and diversity.

Keywords: *recommendation systems, content filtering, data analysis, vectorization of descriptions method, latent semantic analysis method, proximity metrics, factor analysis*

1. INTRODUCTION

Recommendation systems are widely used in foreign companies. They are used on Amazon, eBay, Google News, Yahoo!, YouTube, Google Music, Pandora, Apple's iTunes, Spotify, Twitter, LinkedIn, Xbox game console, Netflix, dating sites, etc. [2, 3]. As it shows, the scope of the recommendation systems is very diverse.

The article considers the field of films search. In conditions when the clients rarely know what they are looking for, a large number of films of different genres complicates the search for suitable content. In this regard, customer satisfaction from using the services of films search is low. The development of accurate and adequate recommendation systems will increase trust and loyalty, which means the number of repeated visits to films search sites.

The following types of machine learning algorithms for recommendation systems are distinguished: content filtering, collaborative filtering and hybrids [4].

In case of content filtering, users are advised on objects that are similar in features to those that they like [5].

Collaborative algorithms work with object-user interaction matrices. Such matrices usually have a large dimension and many gaps. The goal of machine learning is to obtain a function that will predict the usefulness of objects for the user [6].

Hybrid algorithms combine content and collaborative filtering. They are more complex and may be more accurate than the indicated methods [7].

We can also highlight the naive approach, which involves creating a chart and recommending users the most popular objects from the list [8].

Let us consider content filtering algorithms in detail.

2. CONTENT FILTRATION ALGORITHMS

2.1. Architecture of the recommendation system based on the content filtering principle.

Content filtering algorithms are based on the similarity of objects' attributes. Content recommendation systems analyze sets of objects description previously evaluated by the user and on the basis of this, either a user profile is constructed, consisting of his or her preferences

(coefficients for the parameters of the objects), or a recommendation is given immediately on the most similar objects to the ones the user likes. A recommendation system based on content filtering consists of three main modules (Figure 1).
Note to Figure 1:

- Content analyzer - when information does not have a structure (text), it needs to be structured and all descriptions should be brought to the same form. In case of films, text descriptions need to be vectorized.
- Profile learner - this module collects information about user preferences, analyzing descriptions of evaluated objects, and builds a user profile based on the results of the analysis.
- Filtering component - this module analyzes the descriptions of objects together with the user profile. The result of its work is a list of objects that are most likely to please the user.

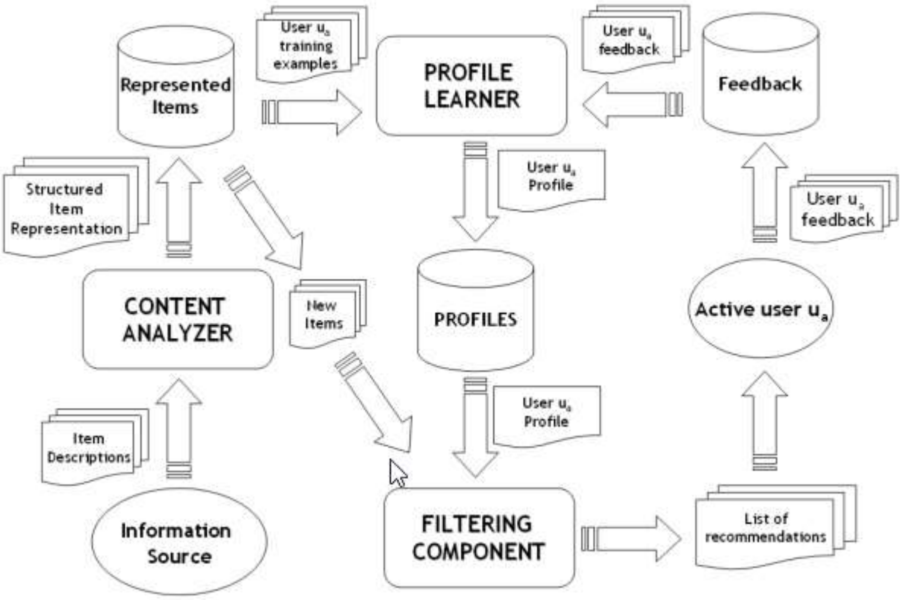


Figure 1 Architecture example of a recommendation system based on content filtering [2]

The main proximity metric in the content approach is the cosine divergence (1) [9].

$$d(x,y) = \frac{x\cdot y}{||x||\cdot||y||} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}, \qquad (1)$$

where n is the dimension of the vectors x, y and x and y vectors' x_i, y_i - $i$components, respectively.

2.2. Descriptions' vectorization

One way to get a vector for a film is Boolean frequency. The vector of the film will consist of 0 and 1, where 1 is the presence of a word in the description of the film. This approach works well when breaking descriptions into tags such as genre, names and surnames of the actors, year of film release, etc.
A more complicated vectorization way is to apply the TF-IDF measure to the movie description. TF (term frequency) - frequency of word occurrence of in a document, IDF (inverse document frequency) - reverse frequency of word occurrence in a document - inverse of the frequency with which a word occurs in documents. Let us denote the number of words t in document d as $f_{t,d}$, and the number of words in document d as n_d. Then the easiest way to calculate TF is to equate tf (t, d) to$f_{t,d}$(2):

$$tf(t,d) = f_{t,d} \qquad (2)$$

Other ways to calculate TF (3-6):

- Boolean frequency (3):

$$tf(t,d) = \{1, t \in d; 0, t \notin d) \qquad (3)$$

- Frequency scaled to the length of the document (4):

$$tf(t,d) = f_{t,d}/n_d \qquad (4)$$

- Logarithmically scaled frequency (5):

$$tf(t,d) = \log\left(1 + f_{t,d}\right) \qquad (5)$$

- Relative frequency (augmented frequency) (6):

$$tf(t,d) = 0.5 + 0.5 \cdot \frac{f_{t,d}}{\max\left\{f_{t',d}: t' \in d\right\}} \qquad (6)$$

IDF estimates the amount of information provided by the word (7):

$$idf(t,D) = \log\frac{N}{1+|\{d \in D: t \in d\}|}, \qquad (7)$$

Where$|\{d \in D: t \in d\}|$- the number of documents that contain the word t;
N - total number of documents.

After this, multiply $tf(t,d)$ by $idf(t,D)$ (8):

$$tfidf(t,d,D) = tf(t,d) \cdot idf(t,D) \qquad (8)$$

Next, the film vector is composed of tfidf values.

2.3. Text description issues.

To improve the accuracy of processing vectorized descriptions before applying vectorization, you can get rid of stop words - words that do not carry any description of the context, for example, the, is, are, in, to and so on.
Names and surnames should be combined in one word. For example, the presence of the word Matthew in the description will give much less information than MatthewPerry.
The vectorization methods described earlier do not take into account the word order, while due to the order the meaning of the statement can completely change. For example, ex1 = “something was cleaned by someone” and ex2 = “someone was cleaned by something”. In the vectorized representation (vex1, vex2), these statements are identical because consist of the same number of identical words: vex1 = vex2 = [1,1,1,1,1]. One way to solve this problem is to use N-grams. Using bigrams in our example will lead to expansions like vex12 and vex 22, voc = [“something was”, “was cleaned”, “cleaned by”, “by someone”, “someone was”, ”by something”]:

- vex12 = [1,1,1,1,0,0];
- vex21 = [0,1,1,0,1,1].

You can also use various combinations of N-grams. The size of N-grams is determined by cross-validation, as the result is highly dependent on the documents, in particular, on their language.
One of the simplest problems of textual descriptions is the difference between the same words, starting with a capital and low-case letters. This issue is resolved by single-case descriptions.
It is more difficult to deal with polysemy and synonymy. Polysemy is when the same word has several meanings, for example, a date. Synonymity is when several different words have the same meaning, for example, awful — dreadful. Due to synonymity, suitable descriptions can be regarded as bad, but due to polysemy, on the contrary, an unsuitable object may be recommended to the user. The impact of these problems can be reduced by latent-semantic analysis.
Latent-semantic analysis (LSA) is a natural language information processing method that analyzes the relationship between the document library and the terms used in them and identifies the characteristic factors (topics) inherent in all documents and terms. LSA consists of two steps:
Vectorization of descriptions. The most common method used for LSA is TF-IDF. Using it, the matrix A (mxn) is constructed, where m is the number of documents and n is the number of words in the dictionary.
Factorization of matrix A. The most common factorization method is truncated SVD (9):

$$A \approx U_t S_t V_t^T, \qquad (9)$$

where t is the number of factors (topics) that is selected to leave.
Probabilistic Latent Semantic Analysis (pLSA) uses a probabilistic approach instead of SVD (10):

$$P(D,W) = P(D)\sum\nolimits_Z P(Z|D)P(W|Z), \qquad (10)$$

Where$P(D)$ - document probability;
$P(Z|D)$ - the probability of the presence of Z topic in document D;
$P(W|Z)$ - the probability of meeting W word with Z topic.
$P(Z|D)$ and$P(W|Z)$are found using the EM algorithm (Expectation-maximization). Second way to set $P(D,W)$(11):

$$P(D,W) = \sum\nolimits_Z P(Z)P(D|Z)P(W|Z) \qquad (11)$$

This approach allows drawing a parallel with LSA (Figure 2).

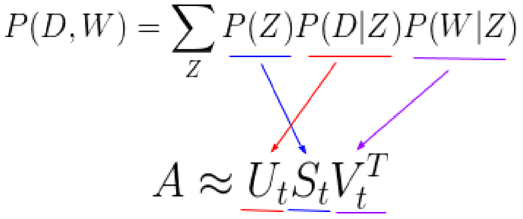


Figure 2 Connection of LSA and pLSA [10]

Cons of the pLSA approach [11]:

- inability to evaluate P(D) for a new document;
- a great tendency to retrain.

In this regard, instead of the pLSA model, the LDA is usually used. LDA - Latent Dirichlet Placement. The main idea of the algorithm is that the document is represented by a random set of topics, and each topic is characterized by a random distribution of words. The similarity of documents is determined by the similarity of their distribution of topics. Similarity can be estimated using the Jensen-Shannon divergence (12):

$$JSD(P||Q) = \frac{1}{2}D(P||M) + \frac{1}{2}D(Q||M), \quad (12)$$

where P and Q are the distribution of the topics of the two descriptions,$M = \frac{1}{2}(P + Q)$, D is the Kullback-Leibler divergence (13):

$$D(P||Q) = \sum_i P(i) \log\left(\frac{P(i)}{Q(i)}\right) \quad (13)$$

Then formula 12 takes the form (14):

$$JSD(P||Q) = \frac{1}{2}\sum_i \left[P(i) \log\left(\frac{P(i)}{\frac{1}{2}(P(i)+Q(i))}\right) + Q(i) \log\left(\frac{Q(i)}{\frac{1}{2}(P(i)+Q(i))}\right)\right] \quad (14)$$

The smaller the JSD, the closer are the distributions to each other. In the case of LDA, the problems of synonymy and polysemy are minimized, as there is no explicit comparison of word sets within documents.

2.4. The main pros and cons of the content approach

Pros:

- there is no problem with the new facility;
- content can be represented in a large number of ways, which opens up many possibilities;
- it is clear why the object is appeared in the list of recommendations;
- recommendations do not require ratings from other users.

Cons:

- availability of a description comparable in form to the rest is mandatory;
- it is not clear what to recommend to a new user;
- recommendation of very similar products (filter bubble).

The unexpectedness of recommendations can be increased by adding objects for which the difference between the probabilities of belonging to the positive and negative classes is minimal in Bayesian models.

These shortcomings of the content approach are partially eliminated in the approach to issuing recommendations based on collaborative filtering, but other problems appear in it.

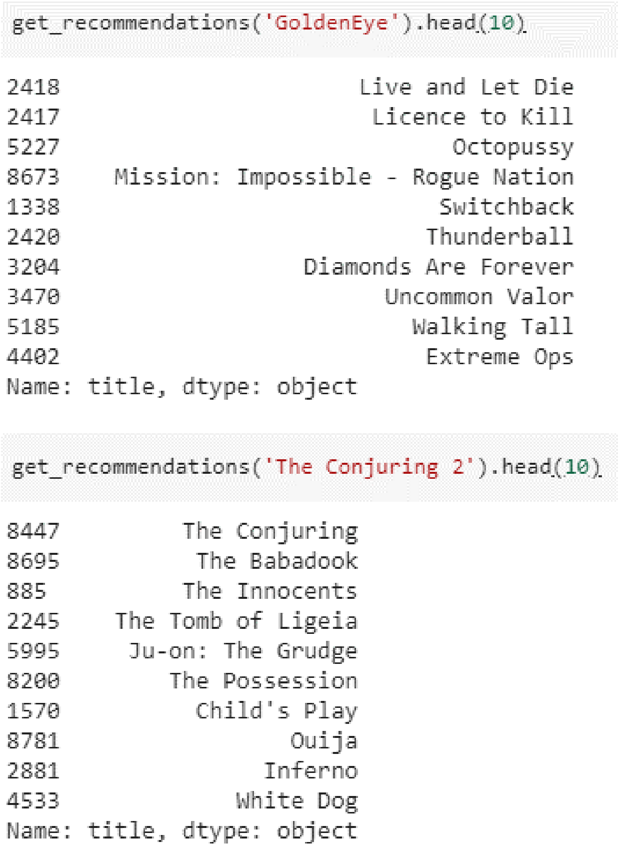
3. EXAMPLES OF RECOMMENDATION SYSTEMS BASED ON CONTENT APPROACH

Designed recommendation systems are tested on open MovieLens dataset data which includes 100,000 ratings, 700 users and 9,000 films. Let us build several models of content recommendation systems.

3.1 Model based on film descriptions

The first algorithm for constructing content recommendation systems is based on film descriptions. In order to save space, we present a graphical image of only the final result (Figure 3), giving a verbal description of the intermediate steps that had to be passed in order to achieve a satisfactory result:

1. Initially, we bring all the words in the descriptions to the same basis.
2. Next, we use the TF-IDF algorithm, not taking into account stop words in the calculation and not using n-grams to vectorize descriptions. To compare the proximity of vectors, the cosine divergence (1) is used. Lets make recommendations for GoldenEye film. The system was able to recognize GoldenEye as a representative of Bond.
3. Let us try to derive recommendations for another film - “The Conjuring 2”. As a result, the first three films are quite similar in spirit to the original, but the fourth and fifth recommendations are worthless, the fourth film is a sports comedy with an average rating of 3 out of 10, and the fifth one is a musical. “Ed” and “The Conjuring 2” have in common only the names of the main characters.
4. Let us try to get rid of names before vectorization using the ntlk.names library in Python programming environment [12]. Most of the names were removed from the descriptions, but the surnames were not touched. Let us try to make recommendations for the same two films. As a result, the quality of the recommendations has become even worse compared to the first method. It seems that the system lacks knowledge of genres.
5. Add the film genres of to their descriptions in triple size and build the corresponding recommendations (Figure 3).



```
get_recommendations('GoldenEye').head(10)

2418                          Live and Let Die
2417                          Licence to Kill
5227                                Octopussy
8673    Mission: Impossible - Rogue Nation
1338                               Switchback
2420                              Thunderball
3204                     Diamonds Are Forever
3470                           Uncommon Valor
5185                              Walking Tall
4402                              Extreme Ops
Name: title, dtype: object
```

```
get_recommendations('The Conjuring 2').head(10)

8447           The Conjuring
8695            The Babadook
885            The Innocents
2245     The Tomb of Ligeia
5995      Ju-on: The Grudge
8200          The Possession
1570            Child's Play
8781                   Ouija
2881                 Inferno
4533               White Dog
Name: title, dtype: object
```

Figure 3 Improved recommendations for “GoldenEye” and “The Conjuring 2” based on a content filtering model built on films descriptions

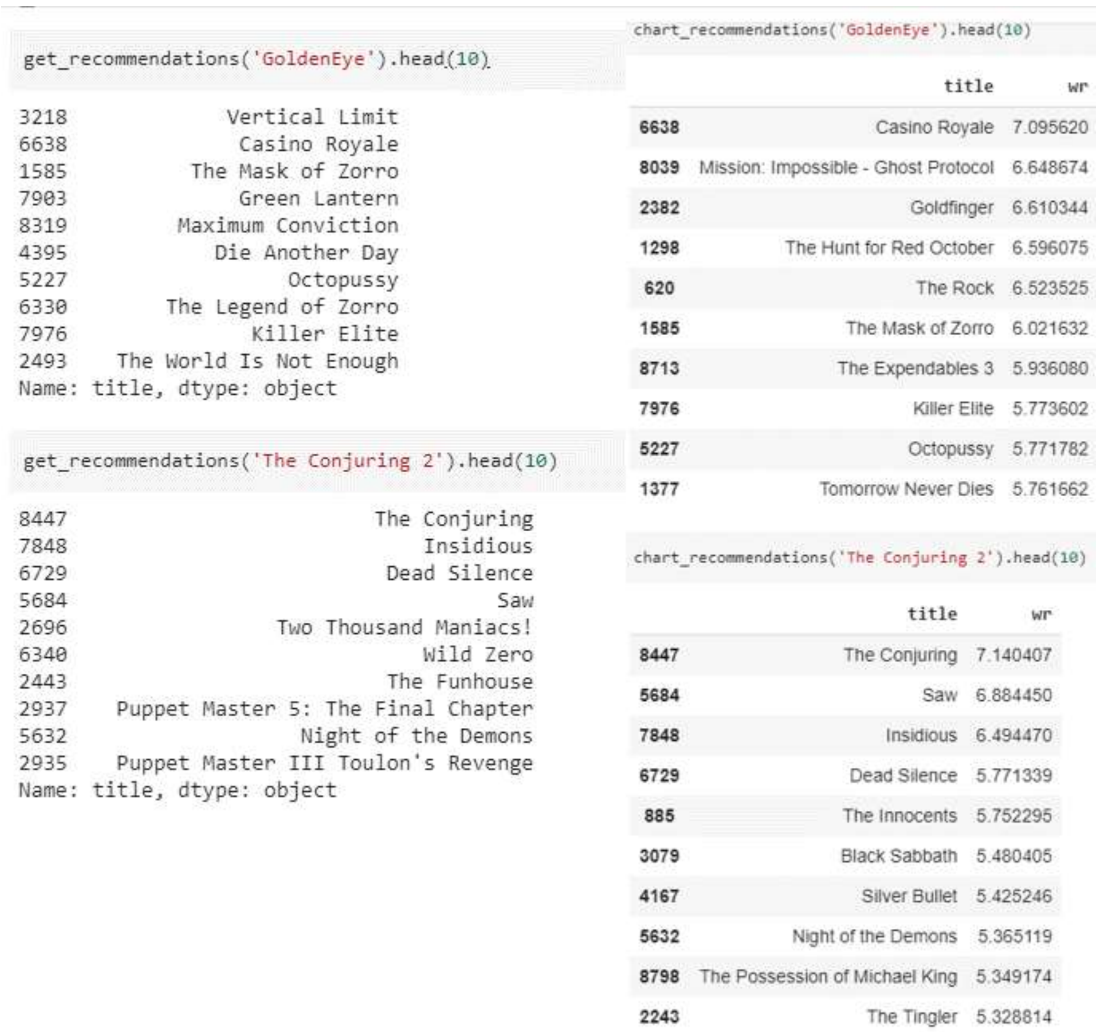
Recommendations have become much better (Figure 3). The film about Lara Croft disappeared from recommendation for “GoldenEye”, which is rather far from the James Bond films, and the film “Mission: Impossible - Rogue Nation” appeared. In the case of “The Conjuring 2”, all comedy and music films disappeared, only horror films remained. To this approach, films check on the chart to recommend films with the highest rating can be added.

3.2. Tag-based model

The second way to design recommendation systems based on the content approach is to build a tag-based model. Let us describe the main stages of this system modeling:

1. As a description of the film, we take the names and surnames of the three main actors, the director, the tags and genres of the film. After vectorization, new descriptions are compared and, similarly to the previous model, recommendations for the films “GoldenEye” and “The Conjuring 2” are obtained. The recommendations for “GoldenEye” turned out to be good, but “Furious 7” in the second list is a terrible recommendation.

2. Let us add genres to the descriptions twice to increase their weight in the sample and sort the recommendations by weighted rating to get rid of the “bad” films (Figure 4).



```
get_recommendations('GoldenEye').head(10)

3218            Vertical Limit
6638             Casino Royale
1585          The Mask of Zorro
7903             Green Lantern
8319         Maximum Conviction
4395            Die Another Day
5227                Octopussy
6330        The Legend of Zorro
7976               Killer Elite
2493     The World Is Not Enough
Name: title, dtype: object
```

```
get_recommendations('The Conjuring 2').head(10)

8447                              The Conjuring
7848                                  Insidious
6729                               Dead Silence
5684                                        Saw
2696                       Two Thousand Maniacs!
6340                                  Wild Zero
2443                               The Funhouse
2937     Puppet Master 5: The Final Chapter
5632                         Night of the Demons
2935      Puppet Master III Toulon's Revenge
Name: title, dtype: object
```

```
chart_recommendations('GoldenEye').head(10)

                                   title        wr
6638                       Casino Royale  7.095620
8039  Mission: Impossible - Ghost Protocol  6.648674
2382                          Goldfinger  6.610344
1298              The Hunt for Red October  6.596075
620                             The Rock  6.523525
1585                     The Mask of Zorro  6.021632
8713                   The Expendables 3  5.936080
7976                         Killer Elite  5.773602
5227                            Octopussy  5.771782
1377                 Tomorrow Never Dies  5.761662
```

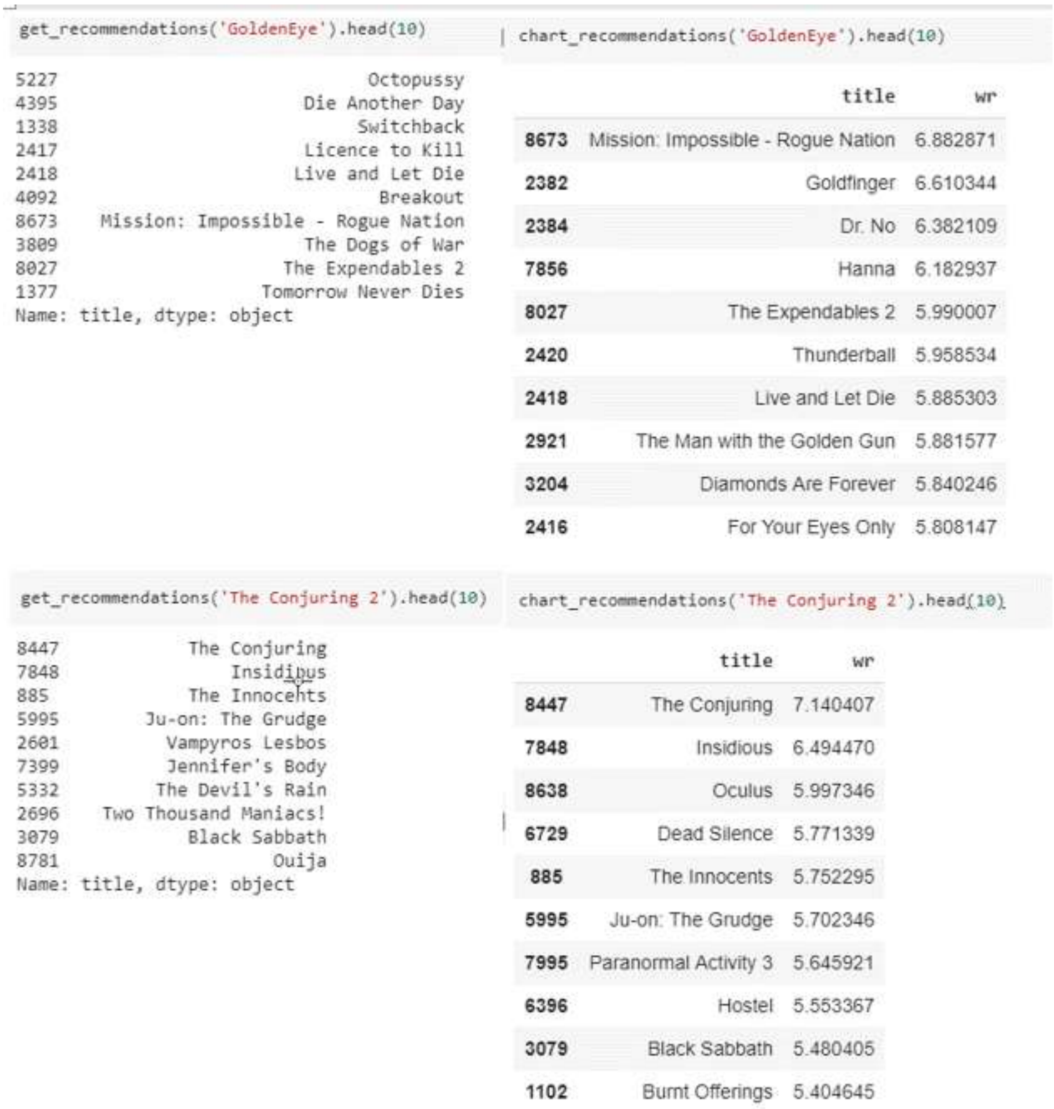
```
chart_recommendations('The Conjuring 2').head(10)

                                title        wr
8447                     The Conjuring  7.140407
5684                               Saw  6.884450
7848                         Insidious  6.494470
6729                      Dead Silence  5.771339
885                      The Innocents  5.752295
3079                      Black Sabbath  5.480405
4167                     Silver Bullet  5.425246
5632               Night of the Demons  5.365119
8798  The Possession of Michael King  5.349174
2243                        The Tingler  5.328814
```

Figure 4 “Weighted” recommendations for “GoldenEye” and “The Conjuring 2” films based on the content filtering model based on tags

The left half of Figure 4 contains recommendations for applying sorting by rating. They no longer have inappropriate films, but they differ from the recommendations of the first model, combining of them can give the best result among the three options.

3.3 Models combining based on film descriptions and tags.

Combine the descriptions of both models and build lists of recommendations based on them (Figure 5).
Figure 5 does not demonstrate any contradictions.



```
get_recommendations('GoldenEye').head(10)

5227                          Octopussy
4395                     Die Another Day
1338                          Switchback
2417                      Licence to Kill
2418                    Live and Let Die
4092                           Breakout
8673    Mission: Impossible - Rogue Nation
3809                     The Dogs of War
8027                   The Expendables 2
1377                 Tomorrow Never Dies
Name: title, dtype: object
```

```
chart_recommendations('GoldenEye').head(10)
```

	title	wr
8673	Mission: Impossible - Rogue Nation	6.882871
2382	Goldfinger	6.610344
2384	Dr. No	6.382109
7856	Hanna	6.182937
8027	The Expendables 2	5.990007
2420	Thunderball	5.958534
2418	Live and Let Die	5.885303
2921	The Man with the Golden Gun	5.881577
3204	Diamonds Are Forever	5.840246
2416	For Your Eyes Only	5.808147

```
get_recommendations('The Conjuring 2').head(10)

8447            The Conjuring
7848                 Insidious
885             The Innocents
5995        Ju-on: The Grudge
2601          Vampyros Lesbos
7399          Jennifer's Body
5332          The Devil's Rain
2696     Two Thousand Maniacs!
3079            Black Sabbath
8781                    Ouija
Name: title, dtype: object
```

```
chart_recommendations('The Conjuring 2').head(10)
```

	title	wr
8447	The Conjuring	7.140407
7848	Insidious	6.494470
8638	Oculus	5.997346
6729	Dead Silence	5.771339
885	The Innocents	5.752295
5995	Ju-on: The Grudge	5.702346
7995	Paranormal Activity 3	5.645921
6396	Hostel	5.553367
3079	Black Sabbath	5.480405
1102	Burnt Offerings	5.404645

Figure 5 Recommendations for “GoldenEye” and “The Conjuring 2” films based on a content filtering model built on film descriptions and tags

4. EVALUATION OF RECOMMENDATIONS

Due to the lack of objective ways to evaluate the lists of recommendations presented earlier, it was decided to create a survey using the Strawpoll service [13] and distribute it among friends, friends' friends, and so on. As a result, 126 people took part in the survey. According to the survey results (Figure 6), it is clear that people preferred recommendations with a rating (Figure 5, right side). Most likely, this is due to the fact that, in the absence of information about the recommended films, people tend to listen to the majority opinion, which represents the rating.

The second most popular option was the list of the first system (Figure 3). Its main advantage is that the films within the list were close to each other by their ideas. Its main disadvantage is that the recommendations for GoldenEye consisted mainly of James Bond films, which indicates a very low difference in recommendations.

5. CONCLUSION

Too much information makes the task of finding relevant content acute. However, the problem is that the recommendation system user finds important not only the accuracy of rating prediction, but also the novelty, surprise, and diversity of recommendations. Therefore, it is necessary to design such a system that would take into account conflicting criteria. In this regard, the following was done in the article:

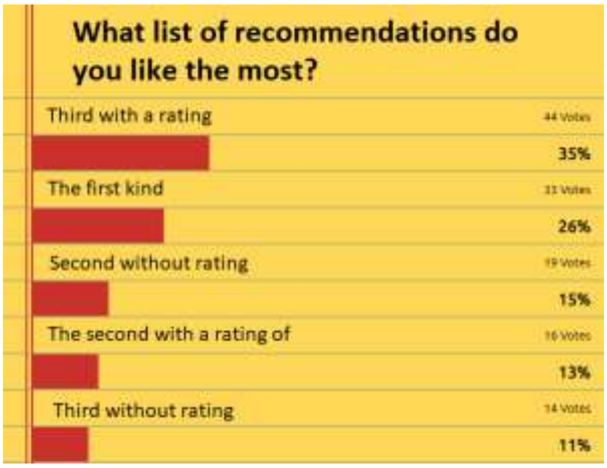


Figure 6 Survey results of potential users of recommendation systems

1. The analysis of the recommendation systems creation theory is carried out. In particular, methods of descriptions vectorization, the method of latent semantic analysis (LSA) and its probabilistic form (pLSA), latent Dirichlet allocation (LDA), proximity metrics (cosine divergence, Jensen-Shannon divergence, Kullback-Leibler divergence), most common factorization method is truncated SVD.

2. Several types of recommendation algorithms were modeled as part of the content approach, namely, a model was constructed that takes into account only films descriptions, a tag-based model and their combined version.

3. The reviews of the system potential users were collected and analyzed. This allowed to cover a number of criteria for evaluating recommendation systems.

An analysis of the created recommendation systems made it possible to understand the user requirements for them. The most important criterion was the level of trust to the system. In other words, this is how much the user is sure that he or she will really like the recommendations offered.

REFERENCES

[1] https://grouplens.org/datasets/movielens/ (data obrashcheniya: 02.06.2019)

[2] Richchi F., Rokach L., Shapira B. i Kantor P.B. Rukovodstvo po sistemam rekomendatsiy. N'yu-York .: Springer; 2015: 1003 p.

[3] TSze Lu, Dyan'shuan Vu, Minsong Mao, Vey Van, Guantsyuan' Chzhan. Rekomendatsii po razrabotke prilozheniy sistemy: opros. Sistemy podderzhki prinyatiya resheniy. Tom 74. Iyun' 2015. Stranitsy 12-32

[4] Dzh. Bobadil'ya, F. Ortega, A. Ernando, A. Gut'yerres. Rekomendatel'nyye sistemy obsledovaniya. Sistemy, osnovannyye na znaniyakh, tom 46, iyul' 2013 goda, stranitsy 109-132

[5] Bager Rakhimpur Kami, Khamid Khassanpur, Khoda Mashayekhi. Modelirovaniye pol'zovatel'skikh predpochteniy s ispol'zovaniyem modeli smesi protsessov Dirikhle dlya sistemy rekomendatsiy na osnove kontenta. Sistemy, osnovannyye na znaniyakh. Tom 163. 1 yanvarya 2019 goda. Stranitsy 644-655

[6] Alan Ekkhardt. Skhodstvo pol'zovatel'skikh (osnovannykh na kontente) modeley predpochteniy dlya sovmestnoy fil'tratsii v stsenarii s neskol'kimi reytingami. Ekspertnyye sistemy s prilozheniyami. Tom 39. Vypusk 14, 15. Oktyabr' 2012 goda. Stranitsy 11511-11516

[7] Isinkaye F.O., Foladzhimy YU.O., Odzhokokh B.A. Sistemy rekomendatsiy: printsipy, metody i otsenka. Yegipetskiy zhurnal informatiki. Tom 16. Vypusk 3. Noyabr' 2015. Stranitsy 261-273

[8] Gediminas Adomavichus, Aleksandr Tuzhilin. K sleduyushchemu pokoleniyu rekomendatel'nykh sistem: obzor sovremennogo sostoyaniya i vozmozhnykh rasshireniy. IEEE SDELKI PO ZNANIYU I DANNYM, VOL. 17, № 6 iyunya 2005, 734-748

[9] Pinata Vinoto, Tiffani YU. Tan. Rol' nastroyeniya pol'zovatelya v rekomendatsiyakh fil'ma. Ekspertnyye sistemy s prilozheniyami. 37 (2010) 6086–6092

[10] https://medium.com/nanonets/topic-modeling-with-lsa-psla-lda-and-lda2vec-555ff65b0b05 (data obrashcheniya: 21.04.2019)

[11] Bley Devid M., Endryu Y. Ng, Maykl I. Dzhordan. Skrytoye raspredeleniye Dirikhle. Zhurnal issledovaniy mashinnogo obucheniya. 2003. tom. 3. S. 993–1022

[12] https://www.nltk.org/ (data obrashcheniya: 24.02.2019)

[13] https://www.strawpoll.me (data obrashcheniya: 02.06.2019)