Recommender System
Report on Data Mining Group Project

Danis Alukaev, Shamil Arslanov, Maxim Faleev, Rizvan Iskaliev, Lada Morozova

1 Business understanding

1.1 Business objectives

1.1.1 Background

Most modern companies rely on machine learning solutions to increase their competitiveness and consolidate or improve their position in the market. Companies that offer entertainment services like no other are focused on recommendations that satisfy each client. The company must find a way to satisfy that condition. A recommendation system was chosen as the proposed solution.

1.1.2 Business objective

The primary objective of the company is to consolidate or even improve their share in the global entertainment market. To do we should offer a personalized service for each client to increase their loyalty to the service.

1.1.3 Business success criteria

The project and integration with the service will be judged by increase in spent time of the customers per visit, customer loyalty and their satisfaction over the service.

1.2 Assessing the situation

1.2.1 Inventory of resources

The personnel available for the project is constrained by 5 people team: 1 business analyst, 1 project manager, 1 data scientist, 1 machine learning engineer, 1 business unit manager. The data available for the project is any fixed extracts of open-source data which relate to the purpose of the project and any data received through scrapping of related websites with open-source APIs. Computing resources are restricted to the use of Tesla V100 16GB. In terms of software usage, GitHub was chosen as base version control system, as the main team-management software Notion was chosen, PowerPoint, Tableau and LaTex were chosen as means for preparation of reports and presentation of the project.

1.2.2 Requirements, assumptions, and constraints

The project is required to be finished by 15th of May and each step of research and development should be well documented and organized. The results of the project should be indicative for the service, taking into account that the quality of the data may affect them, as well as data should be nonproprietary. For success of the project team should assume that there is no change in behavior among clients and that they do not provide self-contradictory feedback. The project is mostly constrained by time of completion, small personnel, and limited computing resources.

1.2.3 Risks and contingencies

Despite all precautions, risks for the project still exist. One of the main risks is poor data quality, which may lead to non-comprehensive analysis of the customers behavior and misleading results of the project. Customer attrition for reasons uncontrolled by the company may lead to a decrease in revenue and to the uselessness of this project. The major risk for the project is a drastic change in customers behavior to which the recommender system developed within the project will not be able to adapt.

1.2.4 Terminology

The glossary of terminology relevant to the project consists of two main parts: business terminology and data mining terminology.

Business terminology:

- Clientele loyalty – customer's likeliness to do repeat business with our company.
- Results indicative for the service - results which may be useful for increasing market share and clientele loyalty.
- Personalized service – providing customer experiences that are tailored to the consumer's individual needs and preferences

Data Mining terminology:

- API – application program interface, which provides a means by which programs written outside of the system can interface with the system to perform additional functions.
- Accuracy – refers to the percentage of suggested movies which the user rated 4 or 5 among suggested movies which user rated.
- Ground truth – information that is known to be true, provided by direct observation and measurement.
- Test data – dataset independent of the training dataset, used to compare the estimates of the model with ground truth results.

1.2.5 Costs and benefits

As costs, we should mainly consider indirect costs (e.g. electricity, etc.), and intangible costs such as change in customers' behavior. However, the company will receive the benefit of increased revenue and customer satisfaction.

1.3 Data mining goals

1.3.1 Data mining goals

Use historical information about previous views of user to generate a model that links "related" movies. When users look at a movie description, provide links to other movies in the related group (market basket analysis).

1.3.2 Data mining success criteria

The project will be considered as a successful in terms of data mining goals if 80% accuracy of recommendations will be reached on the test data.

1.4 Workflow

1.4.1 Project plan

1. Business understanding:
   - determine business requirements, evaluate possible solutions and the strategies for achieving goals
   - take into account costs and benefits for future assessment
2. Data understanding
   - describe data
   - find first insights to the data
   - assess data quality and applicability to the goal achievement
3. Data preparation
   - clean data

   - format data for the modeling step
   - add additional insights from data to modeling step

4. Modeling
   - select modeling technique based on the data
   - choose test and assess model's quality

5. Evaluation
   - evaluate the model in terms of correctness and applicability

6. Deployment
   - deploy the solution for the goal
   - collect the new data based on the solution and return to the evaluation step to assess the quality of solution

1.4.2 Initial assessment of tools and techniques

As the main tool for prototyping and development, the programming language Python was chosen; team management is performed with use of Notion.

2 Data understanding

2.1 Collection of initial data

The main data set was obtained from the GroupLens Research project [1]. Additional IMDB data was collected from IMDB scraping data and some publically available IMDB datasets. When collecting data, several problems arose related to unknown licenses for the use of data, incompatible data encodings and inconsistent data formats that required specific programs. All obtained datasets are located in data directory within project version control.

2.2 Data description

Obtained data from MovieLens dataset consist of

- 3883 distinct movies with following characteristics:
  1. Genres – one or more genres to which movie is related
  2. Movie ID – unique identifier of the movie
- 1000209 ratings of movies with following characteristics:
  1. User ID – unique identifier of user which rate the movie
  2. Movie ID – unique identifier of movie which was rated
  3. Rating – whole number from 1 to 5 represented a rating given by user to the movie
  4. Timestamp – moment of time at which movie was rated
- 6040 distinct users with following characteristics:
  1. User ID – unique identifier of user
  2. Gender – gender of the user (male or female)
  3. Age – category of age chosen from following ranges:
     - ∗ 1 – under 18
     - ∗ 18 – from 18 to 24
     - ∗ 25 – from 25 to 34
     - ∗ 35 – from 35 to 44

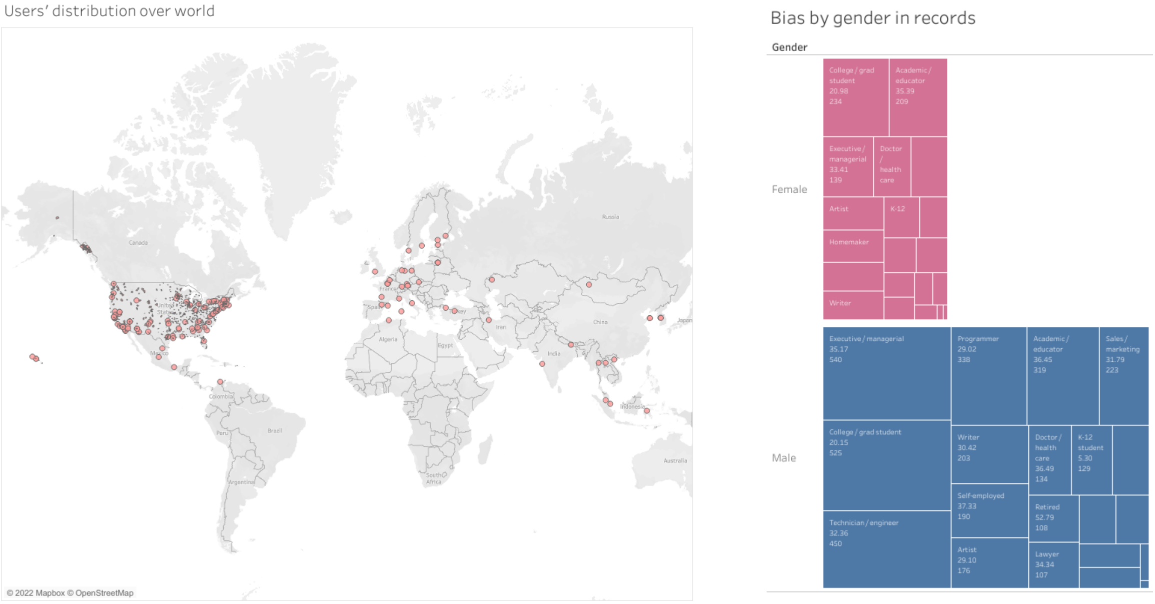
        - 45 – from 45 to 49
        - 50 – from 50 to 55
        - 56 – from 56
    4. Occupation – user’s line of work chosen from following categories:
        - 0 – not specified or other
        - 1 – academic or educator
        - 2 – artist
        - 3 – clerical or administrative
        - 4 – college or graduate student
        - 5 – customer service
        - 6 – doctor or health care
        - 7 – executive or managerial
        - 8 – farmer
        - 9 – homemaker
        - 10 – K-12 (from kindergarten to 12th grade) student
        - 11 – lawyer
        - 12 – programmer
        - 13 – retired
        - 14 – sales or marketing
        - 15 – scientist
        - 16 – self-employed
        - 17 – technician or engineer
        - 18 – tradesman or craftsman
        - 19 – unemployed
        - 20 – writer
    5. Zip code – postal code of the region of living

IMDB dataset consists of following features:

- 16342 distinct movies with following characteristics
    1. Movie ID – unique IMDB identifier of film
    2. Primary Title – title of movie
    3. Original Title – title of movie in native language
    4. Is Adult – whether movie is marked as adult
    5. Start Year – year of release
    6. Runtime Minutes – duration of movie in minutes
    7. Genres – one or more genres of movie
    8. Directors – identifiers of movie directors
    9. Writers – identifiers of movie writers
    10. Average Rating – decimal number from 1 to 10 that corresponds to average rating of the movie
    11. Number of Votes – natural number, describing how many people rated the movie
    12. Region – short identifier of movie country (TR, RU, etc.)
    13. Language – short identifier of movie language (ja, tr, etc.)
    14. Ordering – natural number, representing series number
- 14548 distinct persons that related to movie (actors, producers, writers, etc.)
    1. Person ID - unique identifier of person
    2. Primary Name – name and surname of person
    3. Birth Year
    4. Death Year
    5. Primary Profession – professions of person in cinematography
    6. Known For Titles – movie IDs, in which person participated in.

2.3 Exploration of the data

The MovieLens dataset can be considered as skewed toward the USA in terms of users' records. The following dataset is gender biased, as the number of male users outweighs the number of female users.

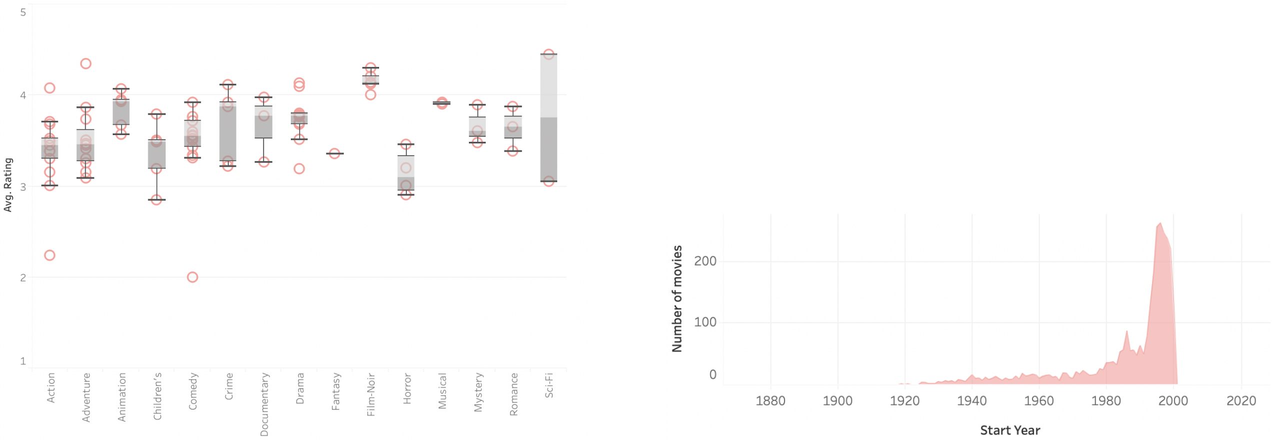


(a) Location bias

(b) Gender bias

Figure 1: Bias in Movielens data

Additionally, variety of movies is bounded by the combination of 18 genres. The distribution of ratings for each genre is not consistent and can have a fairly large number of outliers.



(a) Average rating in MovieLens dataset per genre

(b) Movies' distrinution over the years

Figure 2

The distribution of number of votes and average rating of movies with respect to user can be considered as normal.

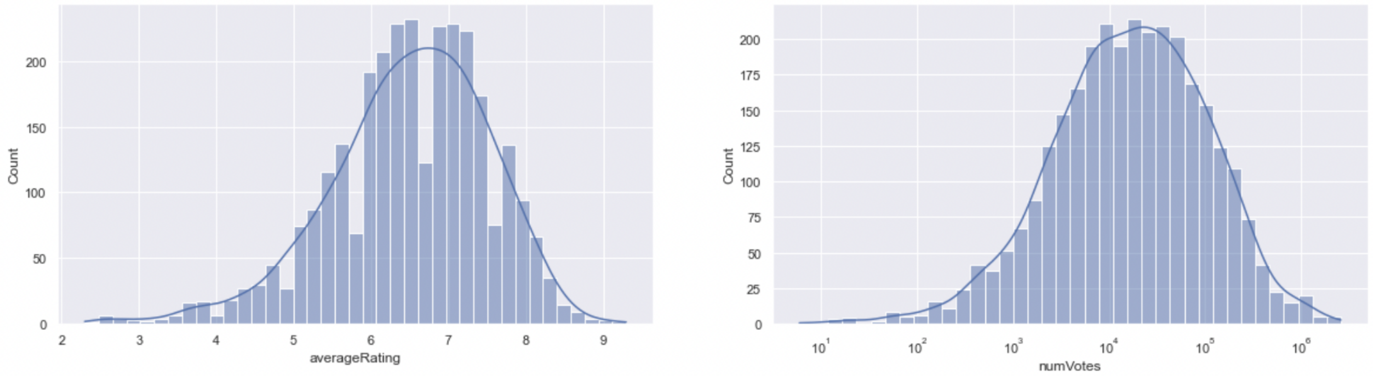


Figure 3: Average rating and number of votes distribution

Moreover, MovieLens dataset contains films released mainly from 1894 till 2000. Other visualizations are available at Tableau Online

2.4 Verify data quality

The data collected for the project can be considered of fairly high quality, since strong outliers and missing data were not detected during the data examination. However, inconsistency in titles of the movies among IMDB dataset and MovieLens dataset was detected, as well as an inconsistency in genres. Moreover, the MovieLens dataset can be considered as skewed toward the USA in terms of users' records. The following dataset is gender biased, as the number of male users outweighs the number of female users.

3 Data Preparation

3.1 Select data

3.1.1 Select data source

MovieLens 1M
The MovieLens 1M is our main dataset. It has three files: ratings.dat, users.dat, movies.dat. The first file contains the information about the films' ratings entered by users and corresponding timestamps. The second file provides the data about users' genders, ages, occupations, zip-codes. Third one gives info about movies: titles and genres. MovieLens 1M dataset is provided here.
IMDB
The IMDB datasets provide more information about different movies and series. We use it to augment existing model with movies features and get better quality of the recommender system. The first dataset `title.basics.tsv.gz` is the entrypoint dataset with various information. First, it matches titles to their inner IMDB identifiers, which are necessary to navigate other datasets. We use this to integrate IMDB data into MovieLens data. Second, it contains information about title genres, which is used further. The second dataset `title.crew.tsv.gz` contains information about title directors, which will further be used for feature construction. Finally, `title.ratings.tsv.gz` provides information about title average ratings and number of votes given to them. IMDB datasets are provided by IMDB itself here.

3.1.2 Select attributes and records

Movie attributes
Movie attributes was taken from IMDB dataset.

1. **Ratings**: We use ratings statistics from IMDB dataset. Ratings statistic includes average rating over different users, and number of votes for the film. Average movie rating by itself is a great approximator for each particular user-movie interaction, its applicability is obvious. Further, we've considered the number of votes

to be a representative statistics for the film popularity, which in turn can be a good feature for the recommender system.

2. **Genres**: We also use information about the film genre. Genre is a categorical value and its processing is explained further.

User attributes
Further we go to the user features. All user features come from the MovieLens dataset

1. **Age**: The MovieLens dataset provides this information about users, and obviously the user age can explain a substantial part of variance in user preferences. So we use it.

2. **Occupation**: What does the user do to not die from hunger. Occupation largely influences the interests, so we use it.

3. **Gender**: Also influences the interest, so we use it too.

3.1.3 Records

The IMDB contains an extensive amount of records for every at-least-slightly popular title, which are not presented in the MovieLens dataset. Only a small subset of titles, which are presented in the MovieLens dataset is used.

3.2 Clean data

Before starting the project, we have carefully investigated different sources of data and have chosen those that are the most exhaustive and exact. Thus, the data is clean and there is no need to clean it more.

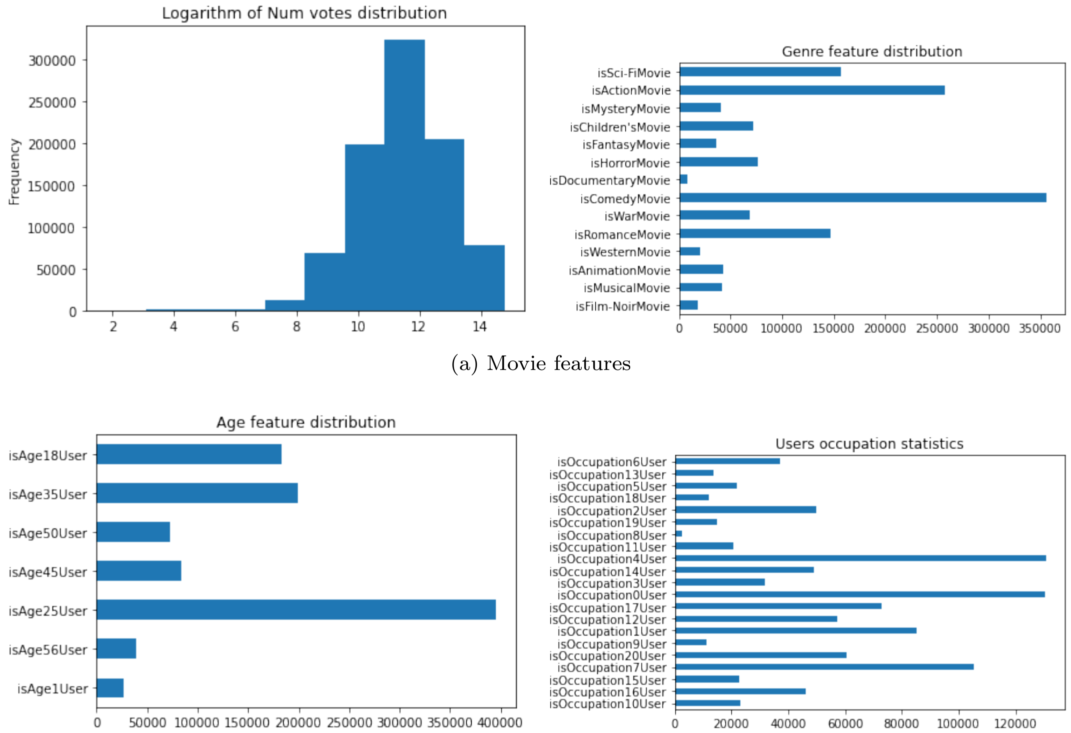
3.3 Construct data

During following step the following attributes were processed to be suited for modeling:

- Movie attributes
  - **Ratings**: The number of votes comes from a distribution with a very heavy right tail, that makes the feature hard to use for linear models (which our recommender system is). Thus, during the augmentation procedure, we apply log scaling for the feature.
- User attributes
  - **Age**: The MovieLens dataset has only a few unique values for the ages, they presumably were splitted to several bins. This decision is quite useful for our system, as the dependency between the user age and their preferences is not really linear, so we will one-hot encode these values. However, we also add the log value of the user age.

3.4 Integrate data

To integrate the data first some changes should be performed. As titles were written in a slightly different ways, some manual string processing was applyed. This includes removing the year from the title and moving articles ”A”, ”The” to the beginning of the string.



(a) Movie features

(b) User features

Figure 4: Some prepared features

3.5 Format data

3.5.1 Convert initial MovieLens' .dat files to .csv files and merge .csv files

We have decided to convert source .dat files to the .csv format since it is more convenient for us to work with Dataframes rather than with raw text files. Furthermore, .csv format allows us to do data analysis and preprocessing more easily.

3.5.2 Movie data

Genres: One movie may have several genres, and the genre is a categorial value, so we make a bitmask for each film, which has 1 if the movie belongs to the corresponding genre. The augmentation gives us 18 feature columns for the film.

3.5.3 User data

- **Age:** again just make a bitmask, encoding the user age. For this feature, one user may have only one age value, so it basically becomes one-hot encoding (OHE).
- **Occupation**: Next, we encode the occupation of users. Again, we create a bitmask which is basically OHE. Twenty-one new feature columns.
- **Gender**: Finally, the gender. We were brave enough to binarize this value.

4 Modeling

4.1 Select Modeling Technique

This section describes modelling techniques employed in this project. We have organized our development process in an iterative manner, i.e. on each iteration we came up with a novel architecture that potentially outperforms the current SoA method. Recent advances in the field were taken from ”A review on deep learning for recommender systems: challenges and remedies" by Batmaz Z. et al. In total, there are four different implemented models: item popularity model, multilayer perceptron, factorization machine, and behavior sequence transformer.

4.1.1 Item Popularity Model

Model grounds on assumption that most popular movies are likely to be interested in a user. We define popularity in terms of ratings number. Thus, for each user recommender system always return top-k movies with the highest number of ratings. Certainly, this is a naive approach and is not likely to satisfy our data mining goal. For this reason, we will assume that the company uses this model in its current setup, and benchmarking of all other (more sophisticated) models will be performed in comparison with performance of this "baseline".

Although the model is simple, the great thing about it is the lack of assumptions about the data. Essentially, we can apply this method to any dataset until it contains user ratings and names of movies.

4.1.2 Multilayer Perceptron

Another quite natural option is to use Multilayer Perceptron (MLP). As an input this model takes concatenated latent representation of user, movie, age, occupation, gender, as well as others manually derived on previous step features. This data passed through multiple dense layers with non-linear activations, e.g. rectified linear unit. The output layer is a single neuron with logit, that is activated using sigmoid function. The result can be interpreted as a probability that the user will enjoy the movie.

4.1.3 Factorization Machine

Until recently, de facto the silver bullet in the field of recommendation systems were factorization machines. They were firstly presented in "Factorization Machines" by Rendel S. in 2010. It was inspired by and proposed as a substitution for traditional matrix factorization. This approach allows to learn interactions between user and movies, and at the same time involve user metadata for final prediction. Thus, this approach resolves well-known problem of the cold-start.

4.1.4 Transformer

Finally, we decided to apply Behaviour Sequence Transformer (BST) architecture for our problem. The implementation is mainly inspired by manuscript "Behaviour Sequence Transformer for E-commerce Recommendation in Alibaba".[3] Long story short, authors expand MLP using sequence-to-sequence model with multi-head attention mechanism. Now the model not only takes into account metadata of user and movie, but also rating history. They claim to increase online Click-Through-Rate (CTR) gain by 7.57% compared to a control group using BST.

4.2 Generate Test Design

In order to slightly relax our problem, let's reformulate it from regression to binary classification. Recall that in the original dataset grades are from 1 to 5, thus the new target is 1 if the grade above 3, and 0 – otherwise.

The dataset will be splitted in train, test, and validations subsets with size of approximately 70, 20, and 10% respectively. The data is splitted according to the timestamp to make sure that it inherits its historical order. Iteratively a model is trained on train subset and tested on test subset (to check how a model generalizes data, detect over-fitting, etc.). Final performance of models is checked on validation subset.

Frankly speaking, the recommender system cannot be properly tested without external validation. Apparently, it seems that the best way is to design the A/B testing, and check how the new approach influenced certain metric, e.g. CTR. This project does not require deployment, so building infrastructure is out of the interest scope. **Metric** The performance of model is defined in terms of percentage of suggested movies which the user rated 4 or 5 among suggested movies which user rated (further referred as accuracy).

4.3 Build Model

4.3.1 Item Popularity Model

Although the model is simple, the great thing about it is the lack of assumptions about the data. Essentially, we can apply this method to any dataset until it contains user ratings and names of movies.

4.3.2 Multi-layer Perceptron

The developed model uses PyTorch embeddings to encode user, movie, age, occupation, and gender. Therefore, the input tensor should consist of indices that can be referred to these embeddings. Also, it is quite important that the input tensor is uniformly distributed, no missing values are allowed, and the input is numeric.

4.3.3 Factorization Machine

The developed model uses PyTorch embeddings to encode user, movie, age, occupation, and gender. Therefore, the input tensor should consist of indices that can be referred to these embeddings. Also, it is quite important that the input tensor is uniformly distributed, no missing values are allowed, and the input is numeric.

4.4 Assess Model

Models are assessed with the percentage of suggested movies which the user rated 4 or 5 among suggested movies which the user rated. More is better.

5 Evaluation

5.1 Evaluate Result

5.1.1 Data Mining Results with respect to Business Success Criteria

As a result of the development process, our team has created 3 models that outperform the baseline model in terms of accuracy. Data scientists performed data preparation and fancy feature engineering. Application of the SoA model yields an increase of 15.2% in the quality of recommendations.
The introduction of a new recommender system might increase the satisfaction of users with our service. This could result in increased time spent by users on the service, customer loyalty, and the flow of new users. In the long term, owners may expect an improvement in the share in the global market.

5.1.2 Approved Models

Multilayer Perceptron, Factorization Machine, and Transformer resolve the problem of cold start by exploiting users' metadata. All models meet the data mining success criterion of 80% of accuracy on the validation dataset.

5.2 Review Process

Although we have reached the data mining goal, some points could be improved:

1. Additional datasets or scraped data can be utilized.
2. For training, there were not used geodata, movie directors, and top actors.
3. Hyperparameter tuning (via Bayesian methods) could be performed for the larger number of trials.
4. As an additional modeling technique we could use graph neural networks and variational autoencoders.
5. Could be performed external validation and applied A/B testing.

5.3 Determine next steps

5.3.1 List of possible actions

1. Scrapped data from open movie databases – time-consuming, but impact is not guaranteed
2. Created new architectures, e.g. GNN, VAE – requires sophisticated data preparation
3. Inference in elastic search – ok!
4. Performed more sophisticated feature engineering – impact is not guaranteed

5.3.2 Decision

The project already seems to be overall success, but next step might be creating API with inferenced Behaviour Sequence Transformer and deployment in Elastic Search. It also makes sense to perform statistical testing to ensure that there is a difference between the baseline model and our SoA.

6 Links

1. Code
2. Tableau Report

References

[1] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4, Article 19 (December 2015), 19 pages. DOI=http://dx.doi.org/10.1145/2827872

[2] Rendle, S., 2010. Factorization Machines. Department of Reasoning for Intelligence, [online] Available at: https://www.csie.ntu.edu.tw/ b97053/paper/Rendle2010FM.pdf.

[3] Chen, Q., Zhao, H., Li, W., Huang, P. and Ou, W., 2022. Behavior Sequence Transformer for E-commerce Recommendation in Alibaba. [online] arXiv.org. Available at: https://arxiv.org/abs/1905.06874.