APPLIED RESEARCH DOCUMENT

Mariela Gocheva

Class: S3-CB02

Student number: 455 2458

TABLE OF CONTENTS

INTRODUCTION

Purpose of the document

The recommendation engines are commonly used in nowadays applications and their popularity rapidly grows due to the customized delightful recreational user experience which they provide. The purpose of this research document is to determine which is the most suitable recommendation algorithm for a music streaming web application as the following aspects are taken into consideration: simple and not overly time-consuming establishment of the system, content-based model of recommendations, mechanism which doesn't hinder the application but still ensures good quality of the final results. For justifying the finest and most satisfactory outcomes of this research, during the process several of the DOT framework research methods will be implemented. The main focus of this research will be spread over all of the five research strategies: library, field, lab, showroom and workshop.

DOT Framework

Good structure and unyielding rationale are characteristics which have vital importance regarding the inditement of any successful research. Having mentioned that, I find the DOT Framework for a great research methodology which perfectly matches my needs. The framework provides an exhaustive overview of the available research strategies and methods thus it facilitates the first steps of the process. Another benefit of the framework is that it can be refracted through a wide range of needs including gaining knowledge about stakeholders, usefulness of tools and frameworks, quality levels of products. Besides providing inspiration and clear guideline, the DOT framework also helps to achieve complete certainty and unambiguous justification of the main research problem.

Scope definition

This research will be conducted in favor of my university individual track project which represents a music streaming web application. The software system contains of a back-end and a front-end application which use an additional library to access an external streaming service. This applied research aims to find the most suitable recommendation engine which can be established for searching through the data provided by the external system.

PROBLEM STATEMENT

Currently, there is no recommendation engine serving my music streaming web application therefore the quality of the user experience is significantly reduced. To provide seamless and engrossing comfort of use, the system needs an uncomplicated a tool which based on the user music preferences (favorite artists, music genres) will deliver precisely chosen music recommendations. Commonly used practices for solving issues from such aspect often include either the implementation of a sophisticated algorithm or a modified search engine. Taking into account the project scope and the most popular techniques, I believe that the insertion of an algorithm into the software system will contribute to a customized flawless user experience and the outcomes will completely satisfy the needs of my project. The fix of this particular problem would lead to significant improvement of the general product quality and the interaction between the users and the application.

MAIN QUESTION

Which type of a recommendation engine is most suitable for a music streaming web application and requires the least amount of resources, while it still provides both reliable and satisfying suggestions for the user without duplicating recommendations or slowing down the software system?

Sub-questions

- What are the commonly used recommendation systems and algorithms?
- Which algorithm is the most user-friendly?
- What are their pros and cons regarding costs, time-consuming criteria and results quality?
- Which recommendation system is the most suitable for a minor project with fewer resources?
- What does already exist in the field? Are there any suitable time-saving tools for the development of the recommendations system?
- Are there any crucial downsides of the system's arrangement? Are there any consistent problems within the algorithm? Can they be prevented?
- How can the reliability of the algorithm be reassured?

RESEARCH METHODS AND STRATEGIES

What are the commonly used recommendation systems and algorithms?

In order to determine the most suitable recommendation engine for a music streaming web application, a crucial role for the research has the performing of an available product analysis which will provide the necessary data regarding the already existing algorithms, based on which further comparison and elaboration for achieving the most relevant results will be fulfilled. For a vast overview of the components and their performance regarding the relation between the input and the output I consider for an ideal match of my needs the component test method which I will briefly perform in order to determine the main components of the algorithms and their purposes.

Library

Available product analysis

Lab

Component test

Which algorithm is the most user-friendly?

For this sub-question I decided to rely on the competitive analysis method due to the detail list of already-existing tools which it can provide. Hereupon, I will be able to observe the pros and cons of each tool and put additional attention to aspects such as quality of the application flow, level of user satisfaction etc. A nice way to get the most accurate and complete insight of the users' likings is to perform a survey. Implementing this research method from the field strategy is not only inexpensive but it's also the most optimal option for getting familiar with people's preferences without engaging a significant amount of the participants' time.

Library

Competitive analysis

Field

Survey

What are their pros and cons regarding costs, time-consuming criteria and results quality?

Investigating the best good and bad practices will create a solid base for the solution of the problem of this research as it will be able to gain experience from already proven sources. This method will give me an insight of what difficulties did other stakeholders faced regarding a variety of factors such as quality level, time expenses and resource costs. Additionally, a great supplement to the solution of this sub-question will be the implementation of the multi-criteria decision-making method. This method from the workshop strategy will give me the opportunity to prioritize the multiple aspects which I need to observe in order to find an answer to the problem. Having an initial general prioritization will simplify the comparison of the sub-aspects of the different criteria involved in the research.

Library

Best good and bad practices

Workshop

Multi-criteria decision making

Which recommendation system is the most suitable for a minor project with fewer resources?

To find the best-matching recommendation engine for a minor project the problem should be investigated in greater detail first in order to determine the importance of the different characteristics of the system. This will describe the significance of all of the qualities of the project and will provide an overview of which aspects with lower-importance can be left out or implemented with fewer resources. Moreover, the performing of a problem analysis is mandatory as it is going to help with creating a better understanding of the main question and it will assist me to stay in the right track in accordance to the scope limitations during the research process. For further research and proper justification of the problem analysis's results I will apply the available product analysis method which will again present the range of possible to use engines and will highlight their advantages and disadvantages.

Field

Problem analysis

Library

Available product analysis

What does already exist in the field? Are there any suitable time-saving tools for the development of the recommendations system?

To find the most relevant solution to the problem a clear line between the possible and the desired but unenforceable situations needs to be set. The gap analysis method provides the necessary comparison and presents the differences in a comprehensive overview as it creates an easily distinguishable plan which differentiates the current situation and the ideal solution. Before that the library strategy needs to take place and minimalize the range of the possible solution. I chose the best good and bad practices method for that task because it's going to point the already-provenly working tools which are used in the field, giving me the base on which later on I can examine the most suitable tool for my ideal solution in accordance with my current situation.

Workshop

Gap analysis

Library

Best good and bad practices

Are there any crucial downsides of the system's arrangement? Are there any consistent problems within the algorithm? Can they be prevented?

In the search of a problem's solvation a major pitfall is to focus only on the strengths of the possible solutions. To perform more completed research which also emphasizes on the outcome's quality I'm going to apply system tests to check what are the downsides of the available tools. Furthermore, I'm going to test again my desired system in order to revise its performance. Finally, a product review will take place in order to provide a finalized performance overview of the product and to justify the trouble-free execution of the solution.

Lab

System tests

Showroom

Product review

How can the reliability of the algorithm be reassured?

Testing the delicate and susceptible to problems parts of the system is not enough to guarantee its flawless performance after deployment. That the reason behind my decision to put my trust into the method of the usability test and the prototyping. With these different methods which use two research strategies I will be able to achieve a complete implementation of the triangulation principle, thus the satisfaction level of product performance will be guaranteed.

Lab

Usability tests
A/B testing

Workshop

Prototyping

RESULTS

1. What are the commonly used recommendation systems and algorithms?

Nowadays a rich variety of recommendation engines is available to respond to all kind of project's needs. After I performed the Available product analysis, I managed to define the following three most common subtypes of recommendation algorithms according to recent research: collaborative-filtering model, content-based model, hybrid model. (Minds, 2022) The collaborative-filtering models are producing their recommendations based on a dataset of user-item feedback and interactions, while the content-based systems are focused on the characteristic features an item has. The biggest and most successful algorithms are usually combining and layering different features of these two subsets, making them much more complex and sophisticated, but in return making them much more resource-demanding as well (e.g., the hybrid model).

During the initial brainstorming process when I faced the problem of this research for the first time, I decided that as a sub-question of the first one can be formulated the following inquiry:

What are the components of these recommendation systems and algorithms?

Inevitably, I decided to continue my research with the Component analysis which gave the following overview of a recommendation algorithm components and phases of functionality:

Components: user input (can be gathered in multiple ways – for example by checking the user's favorite songs or by taking a general look at what's the most liked content by other users), database (stores the user preferences), logic unit (performs the necessary filtering), front-end implementation (requires a separate section for displaying the recommendations).

Recommendation phases:

- Collection
- Storing
- Analysis
- Filtering

These definitions of a recommendation engine's components and phases will be used later on in the research as a criterion for engine compatibility with the project scope and needs.

Outcome overview

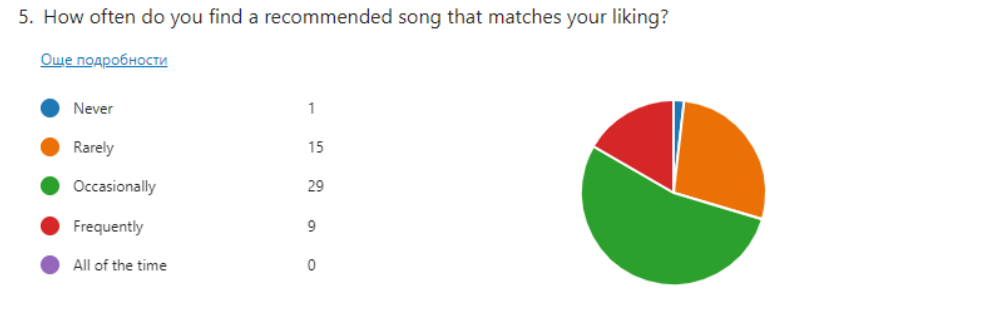
With these research methods was laid a good and informative foundation for the research. By gathering a rich overview of the available resources, I got familiar with the most important commonly used types of recommendation engines and their key components.

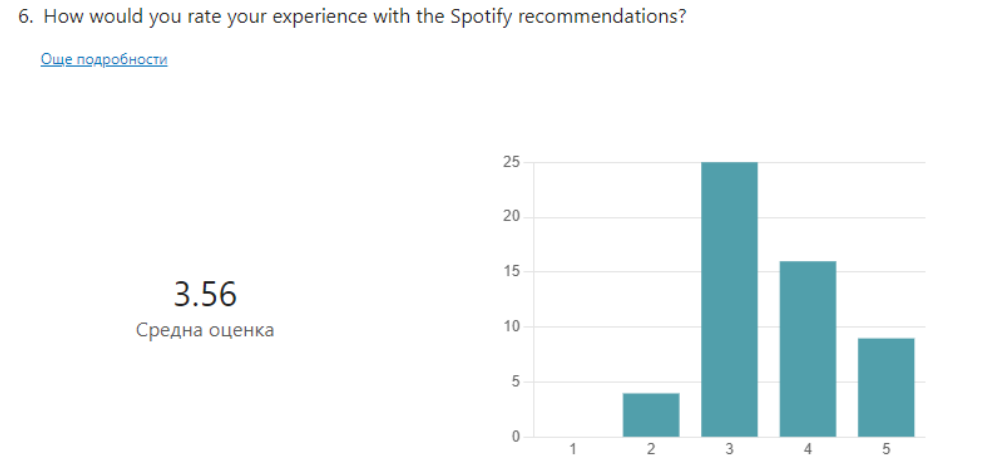
2. Which algorithm is the most user-friendly?

The results of the performed Competitive analysis point out four of the most famous recommendation engines around the social media:

- Spotify (music platform which uses advanced form of a collaborative filtering algorithm that compares multiple user-created playlists which have the songs that users have listened to. The algorithm then combs those playlists to look at other songs that appear in the playlists and recommends those songs.)
- Instagram (social media which provides recommendations that rely on machine learning based on each audience's past behavior and its aim is to create a personalized feed, based on how they interact with other accounts)
- YouTube (online streaming platform which uses collaborative filtering. It combines recommendations based on both the preferred content by the user indicated with a couple of assisting the algorithm functionalities and the generally popular content across the platform)
- TikTok (mobile application which makes advanced machine learning recommendations based on multi-criteria algorithm which not only analyses the users' preferences but also the content itself thus it manages to highlight and define perfectly what is the user appealing content)

Furthermore, I decided to research in greater detail the user experience with the different recommendation systems and I conducted a survey which provided the following extracts results:





8. How often do you find interesting the recommended by Instagram reels?

Още подробности

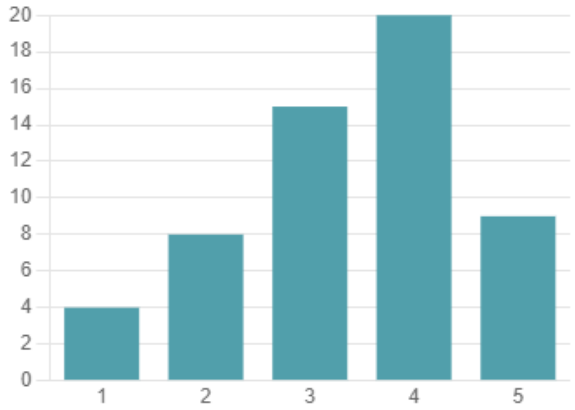
Never	4
Rarely	15
Occasionally	21
Very often	16
All of the time	0

9. How would you rate your experience with Instagram recommendations?

Още подробности

3.39

Средна оценка



Outcome overview

10. What would you prefer the recommendations on a music platform to be based on?

Още подробности

My favourite music genres	33
My favourite artists	25
My search history	9
The content that is most liked b...	9
The content that is liked by user...	11

To sum up, users are most satisfied by complicated machine learning algorithms (used by Instagram, TikTok) and advanced collaborative-filtering algorithms (used by Spotify, Amazon, YouTube, Netflix), however they exceed the scope of this project. Because of that I'll continue the rest of the research basing my decisions on the simplified foundational versions of them (collaborative filtering model, content-based algorithm, hybrid model). The survey gave me a clear idea of users' preferences regarding the content of the recommendations and their general opinions about the user interaction with the algorithms. The full survey and its results will be attached here and at the end of the document.

4. What are their pros and cons regarding costs, time-consuming criteria and results quality?

After defining the most commonly used types of algorithms for recommendations I made a multi-criteria table of factors that I believe are the most important, this way I can have a clear visual representation of which type is most suitable for my needs.

	Affordable	Time efficient	Easy to implement	Suitable for my needs	User friendly	Quality results
Collaborative Filtering	✓	✗	✗	✗	✓	✓
Content-based	✓	✓	✓	✓	✓	✓
Hybrid models	✗	✗	✗	✗	✓	✓

Explanation of criteria

- Affordable – The type of recommendation system doesn't require a lot of resources to be implemented.
- Time efficient – The integration of the recommendation algorithm is not overly time-consuming.
- Easy to implement – The implementation of the particular type of recommendation algorithm is suitable for my level of skills and knowledge and it is not too multi-component with stifling complexity.
- Suitable for my needs – The type of recommendation system does not require the implementation of additional content based on which it is going to perform its predictions (another way of rating artists, playlists etc.). It also does not perform consequential processes to improve its recommendations as this is out of the project scope.
- User friendly – Represents the needs and the likings of the users, considers their level of satisfaction with the recommendation outcomes.
- Quality results – The type of recommendation system presents accurate customized results for a specific user and successfully changes its proposals without duplicating the same content repeatedly.

Outcome overview

The conducted multi-criteria decision-making analysis provided the following results:

The collaborative filtering is affordable and does not require extra expenses for additional resources, however the time efficiency of this type of recommendation algorithm does not match my needs and exceeds the scope limitations. Its implementation requires more researching of the matters, furthermore the unfamiliarity with its integration into the project may cause time-consuming and harmful problems which can seriously affect the end results of the project. Moreover, the collaborative filtering makes indispensable the implementation of other assistant-functionalities based on which it is going to provide its predictions. Regarding user perception – the collaborative filtering satisfies the needs of the users and presents quality customized recommendations.

The content-based algorithms fulfil my expectations regarding affordability and time efficiency. They are easy to implement as they match my level of skills and I'm already familiar with their mechanism. They are suitable for the needs of the project due to their compatibility with the content of my software solution and implementing this type of systems is not going to require additional refactoring of the code and the already established data-relations. The content-based recommendation systems provide pleasing user experience even if they do not provide the remarkable functionalities of the more complex algorithms. They manage to present quality customized recommendations without going overboard with overcomplicated system intricacy.

The hybrid models provide the best user experience with flawless and unsurpassed quality of the results. They completely satisfy a user's needs and make the flow of the application's recommendations pleasantly intriguing and hooking. However, their downsides are that they require large amounts of resources and usually cannot be developed by a single developer especially in a time-tight period which makes them inefficient regarding costs and resources. That also explains why they are not suitable for the needs and the scope of the project. Finally, the level of complexity related to the hybrid models implementation also exceeds my skills and knowledge which may cause serious problems in the project elaboration.

5. Which recommendation system is the most suitable for a minor project with fewer resources?

By performing a detailed problem analysis, I managed to define and put spotlight on a number of criteria to help me determine whether a recommendation system is suitable for my project or not. The main problem was analyzed and via brainstorming broken into the following more comprehensive factors:

- The solution should match my level of skills
- The solution's implementation should be time-saving
- The solution should match the user preferences

Thanks to the previously conducted available product analysis, I applied the new criteria on the highlighted solutions and I came to the conclusion that it will be best for my project if I rely on a collaborative filtering open-source recommendation engine, which will not oblige me to create the whole structure and functionality of the system, thus it will both match my level of skills and save me time. Additionally, the results from the competitive analysis and the survey point out that the collaborative filtering is the most preferred by users type of recommendations, especially when it comes to music streaming.

To gain an overview of the available open-source recommendation engines I performed again the available product analysis which called my attention to the following systems:

- Raccoon Recommendation Engine
- EasyRec
- LensKit
- Crab
- PredictionIO

These systems are distinguished by their easier implementation and broad accessibility as they can be used by any developer regardless his resources and level of skills. They do not provide unnecessarily complex and tangled network of recommendation relations, yet they still give the users great quality of outcomes and pleasant quality of the product. With that said, I consider them as a good match for my needs and resources.

Outcome overview

There are numerous types of available recommendation systems but creating and designing one will be out of the project's scope. As a result, I decided to investigate further the open-source recommendation engines which may be more beneficial for my needs.

6. What does already exist in the field? Are there any suitable time-saving tools for the development of the recommendations system?

Because unexpectedly I already referred to the main concern of this question during researching it, the outcomes of question number 6 will partially overlap the results of question number 5.

In the previous question I found out about the existence of open-source recommendation engines. Further research proved to me that they are so far the best solution which my application can rely on because of their easier simplified implementation. Integration of such algorithm may require either only several adjustments of the setup of the system or the configuration of a connection for API calls (depends of the type of the system) which by all means will cost piles of time less than actually designing such engine from scratch.

Next, I decided to continue my research with a method from the workshop research strategy - the gap analysis. I composed a table which presents multiple criteria based on which I evaluate both the current situation and my ideal situation. The outcomes of that comparison will represent the gap which I will need to concentrate on in order to reach as close as possible to my ideal situation.

Criteria	Current Situation	Ideal Situation
Users can search for songs, playlists	✓	✓
Users can receive recommendation based on the number of playlists they have of a particular genre	✓	✓
Users can view recommendations based on what is popular at the moment	✗	✓
Users can view recommendations based on what their favorite artists release	✗	✓
Users can view an auto-generated playlist which displays the recommendations of songs that the user may like	✗	✓
Users can see a separate section dedicated to recommendations	✗	✓

Explanation of criteria

In the criteria section I included the most valuable features regarding recommendations performance of the final product. These are the goals which if achieved will provide flawless satisfactory user experience with a pleasing application flow.

Outcome overview

The gap analysis indicates that users can already have the most basic type of recommendations – the ones based on a particular genre tag. However, this criterion is very limiting and does not provide sufficient user experience. The missing functionalities which I still need to implement in order to achieve my ideal situation are the following features of the application:

- Users can view recommendations based on what is popular at the moment
- Users can view recommendations based on what their favorite artists release
- Users can view an auto-generated playlist which displays the recommendations of songs that the user may like
- Users can see a separate section dedicated to recommendations

Due to the deadline limitations of the time scope of the project I will not be able to implement all of the ideal case factors, however thanks to the gap analysis I can prioritize the missing components and decide what trade-offs can be done in order to achieve the most satisfying user experience for the shortest amount of time. With that said the gap analysis method helped me gain clear direction for my next steps and again compare my updated prioritized needs with the possibilities which the previous outcomes of these research provide. Taking it further, I can conclude once again that the integration of an open-source recommendation engine based on API communication with my application is the most suitable way to go.

RECOMMENDATION AND CONCLUSION

In conclusion, the conducted above research considers for most suitable in accordance with the project scope the implementation of an open-source recommendation engine. Thanks to a diversity of implemented research methods and strategies this outcome was justified multiple times and its rationale was based on a number of criteria. I believe that even if this solution is not the best for commercial purposes and is not intended for the mass media it is indeed the most relevant match for the project needs. Besides, this research provided a rich overview of the types of commonly used recommendation algorithms and it stated the importance of making a right choice about the type of engine as the study can conclude that there is not a universal solution to every application which will ever use recommendations rather than just factors and aspects which can guide you on the way of finding which one is most suitable for the current project situation.

The current study can be interpreted as a first step in the research on recommendation engines. However, the results of this study should be treated with caution due to the project scope limitations and the lack of details regarding some of the systems due to the accessible and yet insufficient number of sources.

Future research could further examine the differences in open-source recommendation engines or perhaps dive deeper into the machine learning topic, investigating the process and the required skills for implementing such system. Further contribution to the research may provide more complete understanding of recommendation algorithms and can result into improving both the knowledge and the skills of the researcher. Moreover, looking into the problem statement more precisely can lead to another layer of the topic which can examine in more detail the technicality of the systems. Additional researching could also contribute to putting together a generalized overview of the impact which such engines have on the society.

REFERENCES

Minds, C. (2022, September 8). *What are today's top recommendation engine algorithms?* Medium. https://itnext.io/what-are-the-top-recommendation-engine-algorithms-used-nowadays-646f588ce639

Tambekar, A. (2022, August 29). *How Spotify Uses Machine Learning Models to Recommend You The Music You Like*. Great Learning Blog: Free Resources What Matters to Shape Your Career! https://www.mygreatlearning.com/blog/how-spotify-uses-machine-learning-models/

Hu, Y. (2019, January 28). The Mystery Behind Instagram Recommendation System | CCTP-607: "Big Ideas": AI to the Cloud. https://blogs.commons.georgetown.edu/cctp-607-spring2019/2019/01/28/the-mystery-behind-instagram-recommendation-system/

George, T. (2022, November 29). *How to Write Recommendations in Research | Examples & Tips*. Scribbr. https://www.scribbr.com/dissertation/recommendations-in-research/

User Preferences Survey - Microsoft Forms - Easily create surveys, quizzes, and polls. (n.d.). https://tinyurl.com/recommendations-survey