Minor Project Mid Term Progress Report

# (2022, group 22)

**BOOK RECOMMENDATION ENGINE**

Submitted in partial fulfilment of the requirements for the award of the degree of

Bachelor of Technology

In

Computer Science and Engineering



Submitted By ;-

Konark Lohat (1905801)

Gautam Jain (1905785)

Jaskamal Singh (1906006)

Guru Nanak Dev Engineering College Ludhiana - 141006

|  |  |  |
| --- | --- | --- |
| **Sr No** | **Title** | **Page** |
| 1 | Introduction | 3 |
| 2 | System Requirements | 5 |
| 3 | Software Requirement Analysis | 6 |
| 4 | Software Design | 7 |
| 5 | Literature | 8 |
| 6 | Coding / Core Module | 10 |
| 7 | Performance | 11 |
| 8 | Output | 12 |
| 9 | References | 13 |

1. INTRODUCTION

A recommendation engine based on dynamic content, further classified or categorized into content-based (milestone 1) or neighborhood-based collaborative learning (milestone 2) by using matrix factorization methods and other algorithms used in a recommendation system.

The content will be recommended on various factors and in a better / optimized way.

A Book Recommendation Engine Based on Collaborative Filtering Concept using K Nearest Neighbors algorithm to recommend book based on the books liked by other readers and their corresponding ratings (ranking system).

Based on the reading provided by all the readers books are recommended to the users based on the users they follow and their liked books.

The Collaborative Filtering System is established using the cosine similarity relation between ratings and the recommendations with maximum match are returned.

1.1 Objectives

* Implementing an optimized / scalable recommendation system.
* Creating a framework for recommendation engine/system.
* Implementing Content-based filtering.
* Implementing neighborhood based collaborative filtering.

1.2 Technology used

1. **Python** – Python is an interpreted general-purpose high-level language. Created by Guido van Rossum and first released in 1991, Python’s design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.
2. **MongoDB** – MongoDB is an open-source leading NoSQL and document-based database. MongoDB is written in C++. MongoDB handles large volumes of data at high speed with a scale-out architecture. Store unstructured, semi-structured, or structured data. Enable easy updates to schemas and fields. Developer-friendly. Take full advantage of the cloud to deliver zero downtime.

1. **ML/AI –** Machine learning is a subfield of artificial intelligence (AI). The goal of machine learning generally is to understand the structure of data and fit that data into models that can be understood and utilized by people.

1. **Git** – Git is an open-source version control system. It was designed and developed by Linus Torvalds (creator of the Linux kernel) and is the most popular version control system to date.

2. SYSTEM REQUIREMENTS

2.1 Hardware Requirements

* Windows 10+ / Linus / Mac Operating System
* i5 8 Gen+
* 4+ GB RAM
* Dual Core processor.
  1. Software Requirements
* Python 3.6+ installed
* Python Packages (like NumPy, pandas, matplotlib etc.)
* Seaborn

3. SOFTWARE REQUIREMENT ANALYSIS

Most of the recommendation systems used nowadays in big companies/products are not open source. Pushing these technologies like a recommendation engine to be an open-source product creates more horizons for beginners in ML/AI and also to make optimised and accurate recommendations.

This recommendation engine can be used by system architectures to make fast and reliable integration of AI in their application/system. To create a high open-source impact, the reliable scalability design of the recommendation engine has been given proper consideration.

Recommender systems help the users to get personalised recommendations, make correct decisions in their online transactions, increase sales and redefine the user's web browsing experience, retain the customers, enhance their shopping experience, etc. The information overload problem is solved by search engines, but they do not provide the personalization of data.

Recommendation engines provide personalization. There are different types of recommender systems such as content-based, collaborative filtering, hybrid recommender system, demographic, and keyword-based recommender systems. A variety of algorithms are used by various researchers in each type of recommendation system.

4. LITERATURE

1. **Google Developers Community [1]:** Recommendation System by Google Developers Platforms
   * Proposed Learnings:
   * Describe the purpose of recommendation systems.
   * Understand the components of a recommendation system including candidate generation, scoring, and re-ranking.
   * Develop a deeper technical understanding of common techniques used in candidate generation.
   * Use TensorFlow to develop models used for recommendation.

1. **Joeran Beel et al. [2]:** [Research-paper recommender systems: a literature survey](https://d-nb.info/1147681678/34)

* This paper has introduced recommender systems to new research. This paper has also identified key problems which need research in recommender systems.
* This paper can help Ph.D. and Master's students in choosing their area of research. The research gap is already presented in this paper to form different problems of recommender systems.
* The recommendation system finds its utility in major areas of web Applications. As these problems get solved more and more useful recommendation systems will become.

1. **Nitin Mishra et al. [3]:** [Research Problems in Recommender systems](https://iopscience.iop.org/article/10.1088/1742-6596/1717/1/012002/pdf)

* It is a system that helps users to choose items that they may need. A Good Recommender System saves user’s time and keeps the user engaged in the system resulting in higher revenue.
  + Collaborative Filtering
  + Content-based
  + Hybrid Recommendation System

5. CODING / CORE MODULE

6. PERFORMANCE / OUTPUT

7. REFRENCES

**[1]** Google Developers Community (2020). Recommendation System by Google developers platforms. Link - https://developers.google.com/machine-learning/recommendation

**[2]** Joeran Beel, Bela Gipp, Stefan Langar, Corinna, Breitinger et al., 2016. Research-paper recommender systems: a literature survey. Journal on Digital Libraries. Link - https://d-nb.info/1147681678/3442-6596/1717/1/012002/pdf

**[3]** Nitin Mishra, Saumya Chaturvedi, Aanchal Vij, Sunita Tripathi et al., 2020. Research Problems in Recommender systems - Journal of Physics: Conference Series. Link - https://iopscience.iop.org/article/10.1088/1742-6596/1717/1/012002/pdf