|  |  |  |
| --- | --- | --- |
| UNIVERSITY OF COLOMBO SCHOOL OF COMPUTING | | |
| Draft Introduction and Literature Review | | |
| 3204 - Individual Project | | |
| **Project Name** | **:** | **Content and Collaborative based Sinhala Book**  **Recommendation System** |
| **Supervisor** | **:** | Dr. M. G. N. A. S. Fernando |
| **Student Name** | **:** | P. N. C. Perera |
| **Index No** | **:** | 19440677 |
| **Registration No** | **:** | 2019/MCS/067 |

Table of Contents

[1. Introduction 7](#_Toc149513782)

[1.1. Chapter Overview 7](#_Toc149513783)

[1.2. Background 7](#_Toc149513784)

[1.3. Motivation 7](#_Toc149513785)

[1.4. Problem Domain 8](#_Toc149513786)

[1.5. Aim 9](#_Toc149513787)

[1.6. Objective 9](#_Toc149513788)

[1.7. Scope 10](#_Toc149513789)

[1.8. Resource Requirement 11](#_Toc149513790)

[1.8.1. Hardware requirement 11](#_Toc149513791)

[1.8.2. Software requirement 11](#_Toc149513792)

[1.9. Chapter Walkthrough 11](#_Toc149513793)

[1.9.1. Chapter 02: Literature Survey 11](#_Toc149513794)

[1.9.2. Chapter 03: Methodology 11](#_Toc149513795)

[1.9.3. Chapter 04: Implementation 11](#_Toc149513796)

[1.9.4. Chapter 05: Evaluation and Results 11](#_Toc149513797)

[1.9.5. Chapter 06: Conclusion and Future work 12](#_Toc149513798)

[1.10. Chapter Summary 12](#_Toc149513799)

[2. Literature Review 12](#_Toc149513800)

[2.1 Chapter Overview 12](#_Toc149513801)

[2.2 Existing systems 12](#_Toc149513802)

[2.3 Chapter Summary 14](#_Toc149513803)

[3. Methodology 15](#_Toc149513804)

[3.1 Chapter Overview 15](#_Toc149513805)

[3.2 Data Set 15](#_Toc149513806)

[3.3 Architectural Diagram 17](#_Toc149513807)

[3.4 Flow Chart 18](#_Toc149513808)

[3.5 Technology Selection 19](#_Toc149513809)

[3.6 Chapter Summary 19](#_Toc149513810)

[4. Implementation 20](#_Toc149513811)

[4.1 Chapter Overview 20](#_Toc149513812)

[4.2 Preprocessing 20](#_Toc149513813)

[4.2.1 Language Translation 20](#_Toc149513814)

[4.2.2 Sentimental analysis 20](#_Toc149513815)

[4.2.2.1 Using libraries 21](#_Toc149513816)

[4.2.2.1.1 VADER - Valence Aware Dictionary and sEntiment Reasoner 21](#_Toc149513817)

[4.2.2.1.2 TextBlob 21](#_Toc149513818)

[4.2.2.1.3 Compare VADER and Textblob 22](#_Toc149513819)

[4.2.2.2 Using own mechanism 22](#_Toc149513820)

[4.2.2.2.1 Convert Uppercase to Lowercase 22](#_Toc149513821)

[4.2.2.2.2 Remove Links 23](#_Toc149513822)

[4.2.2.2.3 Remove Punctuations 23](#_Toc149513823)

[4.2.2.2.4 Remove Numbers 23](#_Toc149513824)

[4.2.2.2.5 Remove Stop words 23](#_Toc149513825)

[4.2.2.2.6 Apply Stemming 24](#_Toc149513826)

[4.2.2.2.7 Build Vocabulary 25](#_Toc149513827)

[4.2.2.2.8 Vectorization 25](#_Toc149513828)

[4.2.2.2.9 Model training and Evaluation 28](#_Toc149513829)

[4.2.2.2.10 Logistic Regression 28](#_Toc149513830)

[4.2.3 Get Sentiment analysis rate 29](#_Toc149513831)

[4.3 Collaboration based Filter 30](#_Toc149513832)

[4.4 Content based Filter 30](#_Toc149513833)

[4.5 Web Application 30](#_Toc149513834)

[4.5.1 User Interface 30](#_Toc149513835)

[4.5.2 Authenticate 35](#_Toc149513836)

[4.6 Database 35](#_Toc149513837)

[4.7 Chapter Summary 36](#_Toc149513838)

[5. Evaluation and Results 36](#_Toc149513839)

[5.1 Chapter Overview 36](#_Toc149513840)

[5.2 Evaluation 36](#_Toc149513841)

[5.3 Chapter Summary 37](#_Toc149513842)

[6. Conclusion and Future work 37](#_Toc149513843)

[6.1 Chapter Overview 37](#_Toc149513844)

[6.2 Conclusion 37](#_Toc149513845)

[6.3 Limitations 38](#_Toc149513846)

[6.4 Future Enhancement 38](#_Toc149513847)

[6.5 Chapter Summary 38](#_Toc149513848)

[7. References 39](#_Toc149513849)

**List of Figures**

[Figure 1: Questions Users ask from groups 9](#_Toc149576812)

[Figure 2: Architecture Diagram 18](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576813)

[Figure 3: Review after converting to English 21](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576814)

[Figure 4: VADER result 22](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576815)

[Figure 5: TextBlob result 23](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576816)

[Figure 6: Kindle Review Data sample 23](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576817)

[Figure 7: Code to convert review to upper case 24](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576818)

[Figure 8: Code to remove links in reviews 24](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576819)

[Figure 9: Code to remove punctuation in reviews 24](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576820)

[Figure 10: Code to remove numbers in reviews 25](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576821)

[Figure 11: download stopwords from nltk library 25](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576822)

[Figure 12: Read stopwords and store 25](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576823)

[Figure 13: Code to remove stopwords in reviews 25](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576824)

[Figure 14: Read the stem and store 26](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576825)

[Figure 15: Code to apply stemming 26](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576826)

[Figure 16: The way how the stemming is applied 26](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576827)

[Figure 17: Build the vocabulary 26](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576828)

[Figure 18: Vectorization 01 27](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576829)

[Figure 19: Vectorization 02 27](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576830)

[Figure 20: Vocabulary size 27](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576831)

[Figure 21: Vocabulary refactored size 27](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576832)

[Figure 22: Save vocabulary 28](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576833)

[Figure 23: Divide train and Test data set 28](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576834)

[Figure 24: Vectorization function 28](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576835)

[Figure 25: Apply vectorization function for both train and test data 29](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576836)

[Figure 26: Balanced dataset 29](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576837)

[Figure 27: Functions defined to check the accuracy 30](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576838)

[Figure 28: Apply Logistic regression 30](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576839)

[Figure 29: Save the model 30](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576840)

[Figure 30: Get the review rate 31](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576841)

[Figure 31: Find the books that are selected by more than 5 users 31](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576842)

[Figure 32:Filter the book list with above selected books 31](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576843)

[Figure 33:User vs Rate matrix 32](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576844)

[Figure 34: Code to implement the matrix 32](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576845)

[Figure 35: User vs book rate matrix 32](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576846)

[Figure 36:Library for cosine similarity 33](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576847)

[Figure 37: Function to recommend similar books 33](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576848)

[Figure 38: Collaborative filter result 33](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576849)

[Figure 39: replace and with comma 34](#_Toc149576850)

[Figure 40: extract authors 34](#_Toc149576851)

[Figure 41: Convert description to English 35](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576852)

[Figure 42: Combine all together 35](#_Toc149576853)

[Figure 43: Apply Cosine similarity 35](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576854)

[Figure 44: Function to recommend books based on content 35](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576855)

[Figure 45: Calling the function and get recommended books 36](#_Toc149576856)

[Figure 46: Login Page 38](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576857)

[Figure 47: Top Rated books 38](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576858)

[Figure 48: Popular Book List 39](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576859)

[Figure 49: Recommend book list 39](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576860)

[Figure 50: Selected book with reviews 40](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576861)

[Figure 51: Recommended books 40](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576862)

[Figure 52: User stored data 41](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576863)

[Figure 53: Saved Book Data 41](file:///C:\MCS\reports\git_Docs\19440677_MCS3204_Interim_Report.docx#_Toc149576864)

**List of Figures**

[Table 1: Google form with fields details 17](#_Toc149576865)

[Table 2: Technology Stack 20](#_Toc149576866)

# Introduction

# Chapter Overview

This chapter provides a foreword for the project in terms of background study, problem domain, the main aim, objectives, scope and activities that will be carried out towards the completion of the research. Finally, the chapter concludes with an overview on how the other chapters of the document fit into the project context.

# Background

From our childhood, everyone has heard that “Reading makes a man perfect”. People acquire the knowledge by reading a variety of materials. These materials could be a book, an internet article, a newspaper, a magazine, or even a piece of paper, and the gain knowledge by reading these materials is intense. People who read a lot tend to know more about life and are smarter when making decisions and handling difficult situations. (Marappan, 2022) It may not be possible for the reader to “know it all,” but a lot of reading brings man close to perfection. Most of them like to read books as a hobby because it imagines readers' own movie in their mind rather than watching a movie directed by someone.

In today’s world, time has more value and the researchers have no much time to spend on searching for the right articles according to their research domain. (Murali et al., 2019).

Book readers normally select books by reading some random pages or recommended by someone. When reading that book, if he finds that the book is not interesting, he will not read any book after that. therefore, it is better to suggest books that he is interested in. With the increase in library collections, it is diﬃcult for readers to quickly ﬁnd the books they want when choosing books. It is diﬃcult for readers to ﬁnd Sinhala books of interest in a short period of time in the face of various bibliographies. Therefore, the user experience of the traditional library borrowing method is poor (Dhanda and Verma, 2016). Due to the Covid-19 pandemic situation and the geographical barriers also it becomes a tremendous challenge for them (Sarma et al., 2021) to find a relevant book as they do not like to go out and spend time searching books of their preference.

# Motivation

When we navigate through social media specially in Facebook there are so many groups available for almost everything. If you are living in an area, there is a group for that area, if you have an aqua car, there is a group created for aqua car owners. The benefits of such group are you can learn many things and if you have any doubt you can ask from the group and get it clarified. For Sinhala book readers also, there are so many groups available in the Facebook. You can share what your thought for a book or you can see what are latest books released through the groups if you follow those groups. One thing I have noticed is many people ask I have read ‘Adaraniya Victoria’ any one can suggest similar type of books. And some of members ask what are the books related to Sri Lankan history. Some other have asked whether the book is good to read by uploading an image of the book. When considering these three scenarios I thought like it is better to have a system that displays what other users’ thoughts about a book and how much of rate could be given to the book and what are other similar books. It gives the motivation to initiate kind of such system for Sinhala book readers.



Figure : Questions Users ask from groups

# Problem Domain

Most organizations have implemented their recommendation system when users buy products online. But almost all the websites are not developed for the buyer's interest; the organizations force add-on sales to buyers by recommending unnecessary and irrelevant products (Sarma et al., 2021) For instance, if a user has read a book named ‘Madol Duwa’, he would like to read similar books and there is no Sinhala book recommendation system to address this problem. Additionally, some members of readers groups on Facebook have problems like, is this book good or I have read this book and are there any similar kind of books like the mentioned book? Many personal book recommendation systems have emerged to conduct eﬀective search based on user rating and interest.

This paper proposed an effective Sinhala book recommendation system for online users that rated a book list using the content and collaboration (hybrid) method. The solution could be used by all Sinhala book readers to find interesting or suitable books without wasting time or money. Authors also could use the system to have an idea of what kind of books readers rate and are interested more and write books accordingly.

# Aim

The main aim of the research is to analyze, design, implement and evaluate an accurate Sinhala book recommendation system using content and collaboration algorithms with attractive user-friendly interface which display the searched book details along with reviews and recommended books. The main aim can be further divided to three sub aims as;

* Input Data will be the dataset collected from readers.
* Preprocess by removing null values and unwanted data then apply the content and collaborative algorithm
* Final output data will be displayed in the system.

# Objective

The final outcome of this project is helping Sinhala book readers to find correct and recommend books based on their preferences using content and collaboration algorithm. There is no exact formula for determining how much data would be enough for a recommendation system. Even though capturing many data sets, ends up with manageable data sets after preprocessing and removing null values from the collection list. The most online dataset contains parsed data.

The main goal of this research will be achieved by targeting the following objectives.

1. To Collect selected Sinhala book details like title, author, publisher, description, image url and keywords. Online book stores will be used for collecting the details of books.
2. To produce a data set containing user details, selected Sinhala books and rates given by users for those books.
3. To implement web application along with login and registration features.
4. To recommend ten books based on specific field of interest using Content and Collaborative (Hybrid) methodology.
5. To determine the categories preferred by readers so that it motivates authors to write books as per the user’s preference.
6. To Increase the number of book readers by recommending books according to their preferences.

# Scope

The scope of the project can be defined as bellow.

1. Sharing a google form containing selected books along with authors and collecting user rates and reviews for those books based on previous readers experience.
2. Applying sentimental analysis for the reviews collected from above and assigning a new rate.
3. Provide facilities for readers to login or register to the system with their email id which is unique.
4. Provide facilities for readers to search a specific book in the repository and it will display the details such as author, publisher and image along with the reviews and rates given by users.
5. Additionally, the system will display a list of recommended Sinhala books based on users’ rates and reviews of specific interested field. Content based and Collaborative based (Hybrid) approach will be taken place in order to recommend books.
6. Only Sinhala books are recommended and it is based on users’ rates and reviews as well as keywords provided for selected books

We can use library to collect book details, but we will not be able to collect user reviews and rates for selected books. I am using online book store to collect book details but I preferred to get user reviews and rates from users themselves so that they are aware that they have provided the information for the application rather than just coping from the online without their awareness. Until now, around 3750 records have been collected and targeting to collect around 7000 records for the research.

# Resource Requirement

In order to implement and execute the application, following hardware and software requirements should be satisfied.

# Hardware requirement

* A Laptop or desktop with core i3 or above processor
* At least 4GB Ram
* At least 30GB

# Software requirement

* Python latest version – 3.11.5
* VS code as IDE for implementation and execute the application
* MS Excel and Notepad ++ for viewing and manipulating data
* Stable internet connection for downloading relevant libraries.
* GitHub for storing images and implemented code.

# Chapter Walkthrough

The outline of the chapters are as follows.

# Chapter 02: Literature Survey

This chapter will discuss about the review, conducted on the proposed project. It will extensively describe on the stakeholders, the problem, existing solutions, methodologies, and approaches along with their benefits and limitations.

# Chapter 03: Methodology

This chapter will discuss about the methodology to be used to implement the solution. The stakeholders, main technology, libraries, prioritized items, how the collected data is analyzed and the how the architecture of the system will be organized will be in detailed discussed. Furthermore, why the selected technology is more suitable than other existing technologies will be clarified.

# Chapter 04: Implementation

This chapter covers the implementation stage of the project. Algorithms used and challenges faced and how they are resolved will be discussed in this phrase. Screen shots and code segments for some selected functionalities are also provided to facilitate easier understanding and manipulating over the project implementation.

# Chapter 05: Evaluation and Results

The evaluation chapter provides how the results are evaluated based on the feedback collected from Domain experts in this projects Authors. The project will be shown to them and get the feedback for the evaluation. Other than that validation methods will be used to further evaluate the accuracy of the system.

# Chapter 06: Conclusion and Future work

The objectives that were able to be successfully achieved will be discussed in conclusion chapter. The challengers and the limitation of implemented system will be highlighted in order for someone to enhance the system.

# Chapter Summary

The chapter began with explanation on background and problem domain of the system. Although many applications have been developed for book recommending systems, most of them are related to English books. Proper applications that satisfy all the requirements with user satisfaction were limited. The main approach is to make an application that help all Sinhala book readers to recommend Sinhala books based on their preference. A goal followed by objectives was defined to make the effort to be success.

# Literature Review

# Chapter Overview

This main aim of the chapter is, study the existing systems implement for Book recommendation system and find out the limitations. As per the study there are two main models that can be used for the system named as Collaboration filtering and Content based filtering. The chapter concludes by explaining the hybrid model which is the combinations of Content and collaborate filter.

# Existing systems

Several research papers have been published related to book recommendation system. All of them related to English books and no research paper was found related to Sinhala Book Recommendation.

According to the research (Sarma et al., 2021) they proposed an effective system for recommending books to online users that used the clustering approach to rate books and then found book's similarity to suggest new books. The data set were collected from Good readers book repository of Kaggle for the research. To measure distance and determine similarity between book groups, the suggested system uses the K-means Cosine Distance function and the Cosine Similarity function. This study presented a clustering-based book suggestion framework that utilizes various methodologies including collaborative filtering, hybrid, content-based, knowledge based, and utility-based filtering in order to achieve the highest accuracy. Since the accuracy is a crucial aspect of evaluation, they calculate precision, sensitivity, specificity and F1 score and according to the value they have evaluate the system.

The research (Wadikar et al., 2020) proposes a platform that employs a Convolutional Neural Network (CNN) to recommend books based on two approaches. First approach is, using text processing and the second one is using image classification. In text processing approach, it takes the input from the user as a text and process it. The required data set were taken by performing web scrapped from websites like Amazon and Flipkart and processed separately then converted to csv files. In image classification, a book cover image need to be uploaded and the results are displayed accordingly. The book cover images data set were taken also using web scrapped. They use cosine similarity measure to find the similar books related to the subject or image from the sites. The researches have tried to improvised and modified the traditional recommending system and filtering techniques like content based or collaborative based have not been used in the system. They conducted the experiment to list the similar books from Amazon and Flipkart. But the evaluation and validation process have not been presented in the research.

The study (Ijaz, n.d.) propose how to use machine learning algorithms K-Nearest Neighbor and matrix factorization for the recommendation system. It first gathers the rankings or a preference of books provided by multiple users and then suggests books to different individuals based on various previous tastes and preferences. K-Means Multipathing together with K-Nearest Neighbor is applied on the BX dataset which are collected from the Kaggle official website to achieve the greatest-optimized outcome. To calculate the accuracy of the system predictions it used an ordinary statistical metric named root mean square error (RMSQ). RMSE is a measurement of the variation between the user’s real books ratings and the predicted rating for the same books. If the lower the RMSE, the more acceptable the model. An RMSE of zero means the model is absolutely guess the user ratings.

The research (Tian et al., 2019) designs a personalized recommendation system for college library based on hybrid recommendation algorithm which combines both collaborative filtering and content based filtering. According to the algorithm it first classified the readers, then establish user-item scoring matrix, then construct vector space model and finally calculate the similarity among users. The experimental data were collected from Library of Inner Mongolia University of Technology. Since the sparsity is a common problem in Collaborative filtering, the research use clustering to alleviate it. In order to verify the effectiveness of the system it performs the calculation of precision for single algorithm and hybrid algorithm respectively and compared.

The research (Shah, 2019) explain the algorithm collaborative filtering with memory based and model based. A user can either enter rate or sentence which ultimately calculate rate by determining the nature of the sentence using natural language processing. Even the paper discussed various methods that can be used to build a recommendation system, item-based collaboration approach was used for the implementation. It uses “goodbooks10k” dataset in Kaggle for training and test the accuracy and Mean Absolute Error (MEA) is used to determine the quality of the system.

According to (Mercy Milcah Y et al., 2020) they demonstrate a recommendation model that involves Metrix Factorization as a collaborative filtering solution and with further application of artificial intelligence over the previously obtained results from collaborative filtering. The paper presents six types of recommending systems that can be used by user friendly resources or websites or personalized recommending systems. They are collaborative, content based, demographic based, Utility based, Knowledge based and hybrid recommender systems. They also addressed the collaborative problems such as spacity and cold start by combining the system with content based and make it as hybrid.

(Wang et al., 2018) implemented Content based recommend system which gets the information about the scientific article and suggest most appropriate conferences or journals. After desiding the mode of feature acquisition, the content based filtering approach was used to predict through softmax regression which is more generic approach of logistic regression. It provides two kind of recommendation results. The first method is ‘One class’ and it recommends only one journal or conference. The other method is ‘Three class’ and it recommends three candidate journals or conferences. For the evaluation Chi-square, MI and IG are implemented to make comparisons for feature selection.

Collaborative algorithm is the most desired and widely implemented as well as one of most matured algorithms that are available in the industry. It is mainly based on the assumption that users who liked items in the past will like in the future. And also, users would like similar kind of items as they wanted in the past. The approach builds the model based on rating given by other users for a particular book and users past behavior towards the system. One of the drawbacks of this algorithm is that it needs a tremendous amount of real time user data. Other than that sparsity, cold start and scalability are some of limitation of the approach. But user-item scoring matrix and clustering can be used to alleviate the sparsity problem as it allows re grouping all the books based on the rating and user preference datasets.

Content based algorithm is based on description of the item and the profile of the user’s preference. It compares various candidate items with the books previously borrowed or rated by the user and the best matching books will be recommended. The method can be used when a new user login to the system and search for a particular book. The according to the category of the book, a recommended list can be displayed. Some of the draw backs are, it filters the entire set of books from the data set based on the content thus it hinder the performance and it does not help to find out the content quality of the book and it has low accuracy.

Combining any of two types of recommending systems is known as Hybrid recommender system. This is the most demanded method used by many industries as it combines the strength of more than two types of recommending systems while eliminate weaknesses that were there when only one recommended system is used. Since Collaborative based and content-based filtering algorithm having limitations when they used respectively, Hybrid algorithm will be used in proposed system in order to produce efficient and effective book recommendation

# Chapter Summary

The literature review chapter contains what are the existing system available along with tools and technologies used in these systems. As explained, most of the system have implemented mainly either collaboration based or content based specially for English books. While implementing these systems, it is highlighted the limitations of those systems so that the limitations can be addressed in proposed system.

# Methodology

# Chapter Overview

Earlier Literature review chapter helped to identified what are similar systems available in the world and what are the limitations in those applications. This chapter is mainly focused on how the problem is analyzed and identify the methodologies that can be used to implement the system. Further, architectural diagram will be explained along with why hybrid-based application is focused rather than one particular model will be discussed in details.

# Data Set

For any machine learning application, a dataset plays a significant role. While going through some initial process I was trying to find a data set from popular dataset providers like Kaggle. All those data providers have lot of datasets related to book recommendation system. But the limit is all of them related to English books. There were no data set for Sinhala books. Therefore, a google form is created to collect the dataset. The google form is shared in all book readers groups in Facebook and able to collect fair amount of data which can be used to build a model.

Following fields are listed in the form to be provided by the book readers.

|  |  |  |
| --- | --- | --- |
| Input Field | Usage | Google form |
| Email | Unique id to differentiate the user |  |
| Age Range | Drop down list with age range |  |
| Gender | Radio button with two fields of male and female |  |
| Book | This will be a dropdown list which contains 304 books |  |
| Rate | This is a range field from 1 to 10 |  |
| Reason | The input will be considered as a review and taken for sentiment analysis |  |

Table : Google form with fields details

Initial stage of the implementation, an input text filed is given to provide books. But readers entered different data for same value for example, ‘Madol duwa’, ‘Madol Duuwa’. Therefore, a list is created for readers to be selected. In additionaly, if any preferred books are not listed, those books can be included at the end of the form so that the books can be added in to the list by admins in future. high level architecture diagram will be depicted as below.

# Architectural Diagram

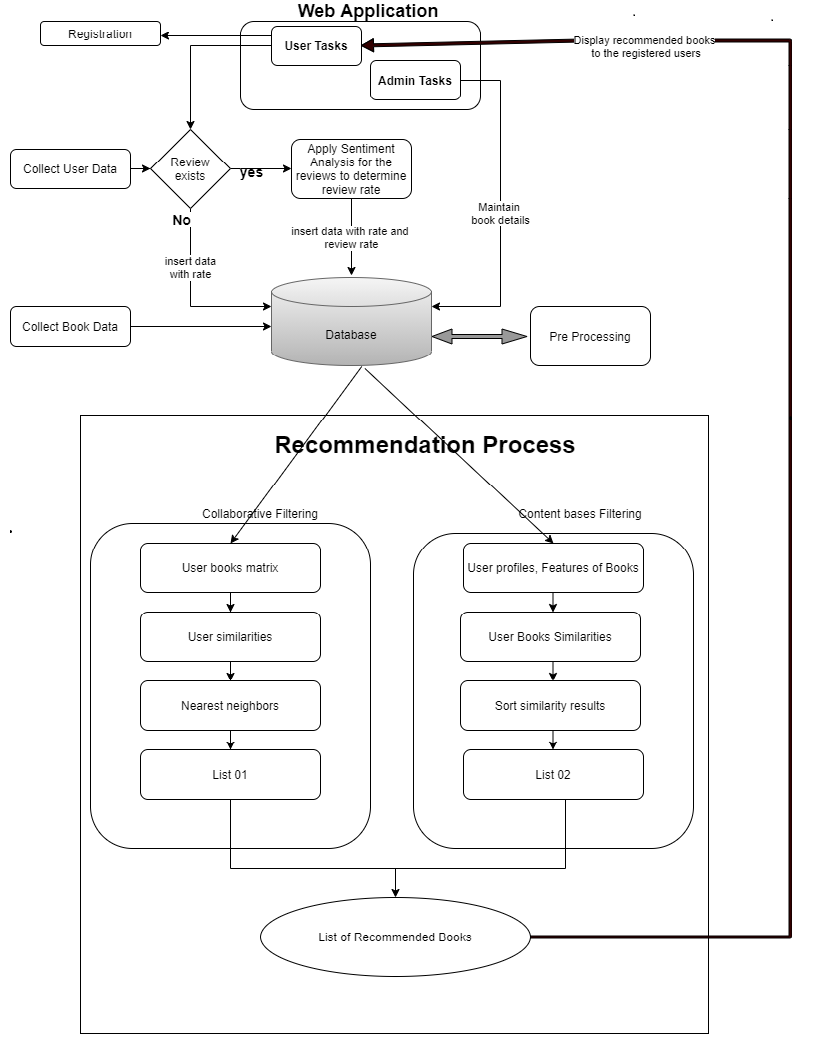


Figure : Architecture Diagram

According to the above architectural diagram, initially data is collected and then it needs to be categorized and store in separate 3 tables as Book\_Details, User\_Details and Rate\_Details. All user related information will be stored in User\_Details table. Additional book data like ISBN, publisher, year should be added in Book\_Details table to display the details of the book when a user search for a particular book. Ratings given for a book by users are stored in Rate\_Details table. If a review exists for a book, a separate review rate will be assigned for the review after completing of the sentimental analysis and include it in Rate\_Details table as a separate column. For the final calculation, mean value of review rate and normal rate will be taken. If there is no review given, normal rate will be taken for the calculation.

Before starting the recommendation process, collected data need to be preprocessed in order to remove unwanted data like null rate values. And data like giving maximum rate for just one and only book should be eliminated as those kinds of books should not be recommended.

Then the process of applying the algorithms will be taken place. As per the paper (Murali et al., 2019) More than 250 research paper recommender systems were published and the quantity of research papers published every day is increasing rapidly. Thus, it needs an efficient searching and filtering mechanism to choose the quality research papers, so that the effort and time of researchers can be saved.

One of the main two methods used to implement the system is Collaborate filtering. The method is used to recommend a user with best books in their domain according to the queries and preferences based on the similarities found from other users. (Murali et al., 2019). It will find the adjacent neighbors of a customer based on the ratings given by the other users. In user based collaborative algorithm, first we need to build a matrix upon users and books with the rating given. Then cosine similarity, which is one of the techniques of K Nearest Neighbor (KNN) will be computed for each user in the matrix. The KNN is a machine learning algorithm to find clusters of similar users based on common book ratings. The cosine similarity first collects books in which is evaluated by all the users in the nearest neighbors, and then the candidate list which the target user has rated or reviewed is removed. Finally, a list of recommended books will be generated based on the similarity.

In content-based approach, it is based on the description of the book and a user’s preference. The algorithm tries to recommend books which are similar to those that a user rated in the past. Initially it abstracts the features such as title, author, genre of books in the system. Then information such as books user read and the rates given will be considered to create user preference vector. Finally various candidate items are compared with the books previously rated by the user and the best matching books are recommended.

According to the researches most of them use hybrid method which combines both collaborative filtering and content-based filtering. Even though there are multiple strategies to apply the hybrid method, applying content-based and collaborative filtering separately and then combining them together will be adopted as it is more effective in book recommendation system. Therefore, the common book list which are generated from collaborative filtering and content-based filtering will be displayed as the final recommended book list.

# Technology Selection

The main technology along with other related technologies and libraries which will be used for the system is as follows.

|  |  |  |
| --- | --- | --- |
| **Front End** | **Logic** | **Data Persistence** |
| CSS - WikipediaInstalling Python 3 and Flask on GoDaddy | by Jordan Ireland | Towards Data  ScienceHTML5 - WikipediaLearn Bootstrap Tutorial - JavaTpoint | Natural Language Processing using NLTK (Python)Google Translator in Python scikit-learn - WikipediaIT12A01: FUNDAMENTALS OF PYTHON PROGRAMMING (SF) - NTUC LearningHub | MySQL Workbench | GPL, Oracle | MySQL is an open source relational databas…  | Relational database management system, Database management system,  Relational database Hosted MySQL - Amazon RDS for MySQL - AWS |

Table : Technology Stack

# Chapter Summary

The chapter explains the main architecture of the system along with the technology which will be used to implement the system. The clarification between the methodologies and why the hybrid method is used is also described in this chapter.

# Implementation

# Chapter Overview

After describing the methodology of the system, the next task is to convert the methodology into a functional prototype. The prototype of the proposed system should address the main objectives that were identified in the first chapter. This chapter will discuss the implementation details individually for identified modules. At the end of the chapter the decisions taken on the low-level implementation would be discussed.

# Preprocessing

When considering the dataset, it is noted that the dataset contains reviews that were written in Sinhala language. Since we are planning to apply sentiment analysis for the reviews, it is compulsory to have the review data in English language. There for the very first task would be to convert the reviews written in Sinhala to English language.

# Language Translation

In order to convert the language, Google Translator library could be used as below. Then all the reviews written in Sinhala will be converted to English reviews which ultimately could be applied sentiment analysis on top of the reviews.

from googletrans import Translator

translator = Translator()



Figure : Review after converting to English

# Sentimental analysis

Sentiment analysis is the process of analyzing a given text and determine if the text means to positive, negative or neutral. It basically helps to understand the human feelings via text. It is one of the Natural Language Processing (NLP) technique used to analyze the text. There are several ways to apply sentiment analysis for a text. Following are some of them

# Using libraries

# VADER - Valence Aware Dictionary and sEntiment Reasoner

It is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media.  It is available in the NLTK package and can be directly applied to a text and gives both polarity(positive/negative) and intensity or strength. The feature depends on a dictionary which maps lexical features with sentiment score.

from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

sent\_analyzer = SentimentIntensityAnalyzer()

text = "the greatest story"

sentiment = sent\_analyzer.polarity\_scores(text);

print("Analyser ----", sentiment)



Figure : VADER result

# TextBlob

It is another lexicon-based python library which can be used to process a text and gives two main values polarity and subjectivity. Other than sentiment analysis, the library contains lot of features like noun phrase extraction, tokenization, lemmatization, spelling correction. As per the below example the text contains the word ‘greatest’ which textblob consider as the sentiment analyser and return positive value 1.0.

Polarity has the value between -1 to 1 where -1 represents the most negative words like ‘worst’, ‘aweful’, ‘disgusting’ while 1 represents most positive words like ‘the best’, ‘excellent’. Subjectivity lies between 0 to 1 where 0 represent factual information while 1 represent more personal opinion. (Barai, 2021)

from textblob import TextBlob

text = "the greatest story"

testimonial = TextBlob(text)

print("textblob -- ", testimonial.sentiment)



Figure : TextBlob result

# Compare VADER and Textblob

When we check some value in Textblob, it is noted that some text which have more negative values like not and slow, it multiplies -0.5 and -0.3 and gives the polarity of the sentence as a positive value. Another issue of Textblob is, if it finds any negative word in between in a sentence, it gives some polarity other than 0. Due to these issues Textblog could not be considered as one of the best sentiment analyzers.

When the same above sentences check with VADER, it gives better result than Textblob. As per the (Barai, 2021) it compares both analyzers and came to a conclusion that Textblob struggled with negative sentences. The discussion further explained that It is not that VADER is better than Textblob in sentiment analysis. But it works better for negative sentences.

As conclusion, there are drawbacks for both of the analyzers. Therefore, it would be more convenient to implement own mechanism for sentiment analyze and predict the value for a given sentence.

# Using own mechanism

There are some limitations when use any library in our system like not able to customized and not able to understand the logic behind the functionality. There for own mechanism of implementing sentimental analysis would be used. Following are the list of main steps for building the model then implement the pipeline for the model built. (“Machine Learning Project | Classification | Sentiment Analysis | Sinhala - YouTube,” n.d.)

In order to train the model for sentiment analysis, ‘Kindle reviews’ dataset was taken from Kaggle. The dataset looks like as below



Figure : Kindle Review Data sample

# Convert Uppercase to Lowercase

The review text contains uppercase as well as lowercase. As a first step all the characters to be converted to lowercase so the case sensitiveness can be ignored when comparing values.



Figure : Code to convert review to upper case

# Remove Links

Since the links do not have any meaning for sentiment analyzer, those links to be removed with below code.



Figure : Code to remove links in reviews

# Remove Punctuations

Since the punctuations also do not have any meaning for sentiment analyzer, all the punctuations to be removed with below code. Punctuation list can be found in string library. A function is defined to remove the punctuations and it is invoked in all the review text.



Figure : Code to remove punctuation in reviews

# Remove Numbers

There were numbers also added in the review test and those were also to be removed as they do not have any meaning for sentiment analyzer process. Removing numbers in a text can be achieved by below code.



Figure : Code to remove numbers in reviews

# Remove Stop words

There were numbers also added in the review text and those were also to be removed as they do not have any meaning for sentiment analyzer process. The list of stop words can be download from nltk library to a folder specified.



Figure : download stopwords from nltk library

A variable is defined to store the list of stop words as below.



Figure : Read stopwords and store

Finally, the stop words are removed from the review text with following code



Figure : Code to remove stopwords in reviews

# Apply Stemming

After removing all unnecessary values in the text, the next phrase is converting different verb formats to a common pattern. As an example, write, wrote, written, writing are converted to base form write. This process is called as stemming and the following code snippet will do the conversion.



Figure : Read the stem and store



Figure : Code to apply stemming

After completing the preprocessing part for the reviews, the text came up without uppercases, links, punctuations, numbers and stop words. Finally stemming has been applied to convert all the text to their base form. Following depict shows how the conversion has been done up to now

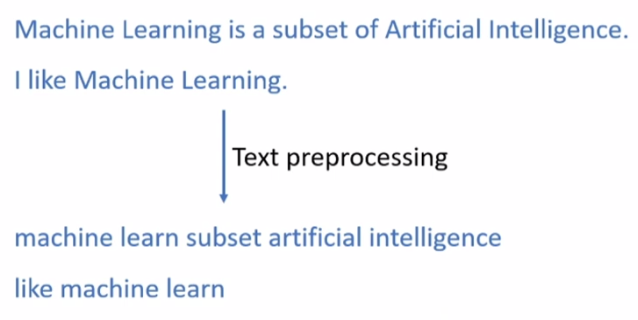


Figure : The way how the stemming is applied

# Build Vocabulary

In order to build a model, the machine is not able to read and understand the text or string and they need to converted to numerical values. The building the vocabulary is the process of converting the text to appropriate numerical values. As the first step a unique vocabulary set to be created from the converted text. In the above example, following is the list.



Figure : Build the vocabulary

# Vectorization

The next step of converting the text to numerical values, is vectorization process. As per the above example, all the sentence could be converted to numerical value which has the length of six (06). The value is same as the length of the vocabulary list. The list contains values which are called as features. In this example there are six features in the vocabulary.



Figure : Vectorization 01

Figure : Vectorization 02

After the process, all the reviews will be converted to a numeric value which have the same length. After that the output can be fed to machine learning model.

Following code will check the size of the vocabulary list simply a number of features. The list contains a unique text and the number of times the text is used in the review.



Figure : Vocabulary size

As per the above result, there are 33,599 features found in the reviews. In the scene, all the reviews will be represented as numeric value which has the length of 33,599. But the Kindle review data set contains around 12,000 records. If this much of features are used, the model will be over fit. The number of features should be less than the number of records in order for model to be a good one.

To overcome the issue, the feature selection will be used to reduce the feature count as bellow. Then the feature count is reduced to 3645.



Figure : Vocabulary refactored size

The final output of vocabulary list will be saved as bellow



Figure : Save vocabulary

When an own method of creating a model for sentiment analysis, accuracy plays a significant role. It gives how the model is accurate as percentage.

Before vectorization, the review dataset to be divided to two main parts as training data and test data. The training data will be used to train the model and the test data will be used to test and get the accuracy of the model.

x = data['reviewText']

y = data['rating']



Figure : Divide train and Test data set

The vectorization can be done via bellow function.



Figure : Vectorization function

Then the function will be invoked for both train and test data as follows. Then all train data reviews and test data reviews will be converted to numeric data set.



Figure : Apply vectorization function for both train and test data

Note how the values of data set are fairly shared (balanced dataset) for each rate as below



Figure : Balanced dataset

# Model training and Evaluation

The next stage of the sentiment analysis process is building a model and evaluation.

from sklearn.linear\_model import LogisticRegression



Figure : Functions defined to check the accuracy

# Logistic Regression

In order to train the model, we use logistic regression as it has the highest accuracy rate among other classification algorithms like decision Tree, Random Forest, Naïve bayes.



Figure : Apply Logistic regression

# 

 After the model is trained properly, it is saved to a location as below

Figure : Save the model

# Get Sentiment analysis rate

After the model is build, the sentiment analysis value to be calculate for a given text. There for the given test to be preprocessed, vectorized before get the prediction. Following code will invoke the appropriate function and return the predicted value.



Figure : Get the review rate

As per the above result, it gives positive value 4 out of 5 for the given text. Since the rate calculated for the application is out of 10, it needs to be multiplied by 2 as the review rate.

# Collaboration based Filter

Since Content based filtering cannot discover the quality of an item, collaborative filtering system is used to overcome this problem.

As per the first step, we need to find books that are selected by at least more than 5 users. If no one is selected a book, they should be ignored and should not recommend those books for the users. Below code will remove such books.



Figure : Find the books that are selected by more than 5 users



Figure :Filter the book list with above selected books

The process of applying the Metrix and how the collaboration filter works is depicted below.



Figure :User vs Rate matrix

The code snippet to implement the matrix as below



Figure : Code to implement the matrix

The result of the matrix can be found as below.



Figure : User vs book rate matrix

Finally, applying cosine similarity for the matrix and finding the similar books as below.



Figure :Library for cosine similarity

With below function, we can recommend the books.



Figure : Function to recommend similar books

Calling the function and getting the book list

Figure : Collaborative filter result



<https://medium.com/mlearning-ai/recommendation-systems-content-based-filtering-e19e3b0a309e>

# Content based Filter

The Collaboration filter is totally based on the previous data collected by other users. If the user does not have previous data, similar books are not be able to recommended. Even a new book added to the system and it has not been rated, it can not be recommended. Content based filtering introduced to overcome these problems, Even previos data is not found, books can be recommended based on the contents.

For an instance a book contains keywords like ‘sherlock Holmes’, ‘detective’ books which have similar keywords will be recommended. The fist task of the process is replacing ‘and’ with a comma (,). It could be done with bellow code snippet.



Figure : replace and with comma

With below code, all authors can be extracted and save in a different field.



Figure : extract authors

Below code will convert Sinhala description to English as the description will be added to the tags



Figure : Convert description to English

Combine all together as tags as below



Figure : Combine all together

Apply cosine similarity as below



Figure : Apply Cosine similarity

Define a function to recommend content similarity.



Figure : Function to recommend books based on content

Calling the function and getting the book list



Figure : Calling the function and get recommended books

# Web Application

The web application is developed using Flask in python and it contains user interface and the authentication.

# User Interface

The graphical user interface (GUI) of the web application is implemented using python flask, HTML, CSS and Bootstrap. The main file which defines the routes of the application is as follows.

from flask import Flask, render\_template, request, redirect, session

import pickle

import numpy as np

import pandas as pd

import mysql.connector

import os

app = Flask(\_\_name\_\_)

app.secret\_key=os.urandom(24)

@app.route('/')

def login():

    if 'user\_id' in session:

       return redirect('/home')

    else:

       return render\_template('login.html')

@app.route('/register')

def register\_ui():

    return render\_template('register.html')

@app.route('/logout')

def logout():

    session.pop('user\_id')

    return redirect('/')

@app.route('/login')

def login\_ui():

    return render\_template('login.html')

@app.route('/register\_user', methods=['post'])

def register\_user():

    loginName = request.form.get('login\_name')

    password = request.form.get('password')

    query = "INSERT INTO users VALUES ('" + loginName + "', '" + password + "');"

    try:

        connection = mysql.connector.connect(host="localhost", database="sinhala\_book\_recommendation", user="root", password="admin");

        cursor = connection.cursor();

        cursor.execute(query);

        connection.commit();

    except Exception as e:

        print("Something went wrong", e);

    finally:

        if connection.is\_connected:

            connection.close();

    return render\_template('login.html')

@app.route('/validate\_user', methods=['post'])

def validate\_User():

    loginName = request.form.get('login\_name')

    password = request.form.get('password')

    query = "SELECT \* FROM users WHERE login\_name='" + loginName + "' AND password='" + password + "';"

    try:

        connection = mysql.connector.connect(host="localhost", database="sinhala\_book\_recommendation", user="root", password="admin");

        cursor = connection.cursor();

        cursor.execute(query);

        users = cursor.fetchall();

    except Exception as e:

        print("Exception when login", e);

    finally:

        if connection.is\_connected:

            connection.close();

    if len(users) > 0:

        session['user\_id'] = users[0][0]

        return redirect('/home');

    else:

        return redirect('/');

if \_\_name\_\_ == '\_\_main\_\_':

    app.run(debug=True)



Figure : Login Page



Figure : Top Rated books



Figure : Popular Book List



Figure : Recommend book list



Figure : Selected book with reviews



Figure : Recommended books

# Authenticate

The authentication also integrated with the application so that none of a user can view the data without login to the system and it was done to improve the security of the application. Admin user will be added as the main user who can view all the top rated, most popular and recommended books. User management and book management will be implemented as the project is still under implementation.

# Database

In order to store persistence data like book details, user login details my sql database is used. MySql workbench is used to manage data in mysql databse. Even there are multiple database like oracle, postgres available and can be used for the same purpose, Mysql was used as it is open source and can easily be managed. Mysql connector in python was used to connect the application with the database.



Figure : User stored data



Figure : Saved Book Data

import mysql.connector

# Chapter Summary

The chapter explained how the implementation was done mainly using python programming language. Initially the data needed to be preprocessed to remove unwanted data. Then it describes what the methods available for sentiment analysis and how the own model was build and trained. The implementation of UI and authentication was described with screen shots. Finally, how the database is connected to the application was explained.

# Evaluation and Results

# Chapter Overview

Implementing any application without proper testing or evaluation is considered as incomplete system. In industry also when we implement the application, after the QA test we hand over Client for User Acceptance Test (UAT) and get the feedback of the client. The feedback is kind of evaluation of what we have implemented.

# Evaluation

There are two options to validate the system named as offline validation and online validation. For offline validation, I will be having user data and performing a standard machine learning training-test split in order to learn and train the model for the evaluation. Mean Absolute Error (MAE) or Root Mean Square Error (RMSE) or any other evaluation function could be used. The main point here is, this kind of evaluation cannot be done without interacting large amount of data. For online validation, a recommender model will be created based on information taken from other domains which is also called as cross-domain recommender system and test the system with live data. Since the implementing system do not have access to some live system, I will be focusing on finding a data set that is more than enough for the offline validation. In order to perform offline validation for the application, we can make use of the concept of precision-recall. Recall describes, what ratio of items that a user like will be actually recommended. And the precision describes out of all recommended items, how many items user actually will like. The main idea of any recommending system is recommending only items user likes. This is the optimal recommender and My target is to get as close as possible.

In order to validate the model further, expert authors will be contacted. He would check the recommended book list is matches with searched book or the list contain books which is preferred by the user.

According to the proposed solution, expecting accuracy level would be more than 80%. We can increase the accuracy by collecting and allocating more data for training. At the end of the project a user can find a best recommend books according to his preference and the rates given by other users. Once the recommended book is read the user may realize the accuracy of the application and no need to waste time on finding the books in everywhere. Once the model is developed, we can use it to make recommendation for that we need to save the desired model and restore it when we need to do recommendation through it

# Chapter Summary

Evaluate the implemented system was discussed in this chapter with test results. In order to evaluate the system, some experts will be contacted and get their feedback. The accuracy of the evaluation will also be discussed in the phrase.

# Conclusion and Future work

# Chapter Overview

The conclusion chapter presents the final part of the application and discuss the aim and the objectives have been achieved successfully. Furthermore, the limitations and future work will be discussed in order for someone to add the missing feature and enhance the application.

# Conclusion

The idea to implement an application for Sinhala Book recommendation came to the mind by surfing Facebook book related groups. Many people ask so many questions like is this book, I like books related to history and romance and please suggest me some books. The process of implementing such system began by finding any data set. Kaggle and many dataset providers have not provided any data set related to Sinhala books, thus a google form was created to collect the data which was a challenge. Even the form was share among groups, not able to collect data set as expected. Therefore each member of groups were contacted and shared the form to be filled.

In parallel of collecting data some research papers were read and understand what are the system available and limitation of them. In literature review chapter all the details were discussed. Then the implementation started by learning Python programming language. In order to apply the hybrid model, First Contant based filtering was applied to the dataset and then Collaboration filter was applied on top of that. Based on the feedback provide by the authors or experts the system will be evaluated.

# Limitations

Even most of the features were able to completed as per the proposal, some limitations were found to be implemented as an enhancement.

1. The age and gender fields were not considered for recommendation.
2. The latest books were not considered.
3. If user have not read any books, he is not able to use the system without select at least a book from the list.

# Future Enhancement

The application was implemented as per the proposed system and there are some features that could be added as enhancements for the features of the system.

1. Since there is a drop down to select books, most books were selected from the top of the list.
2. The drop-down book list was created in English. Most people entered data, suggested to display them in Sinhala as the project is related to Sinhala books.
3. The user should be able to select any books based on some categories like author and genre like history, romance, detective.
4. The dataset was limited like only around 4500 were able to be collected.

# Chapter Summary

The main goals and objectives of the application were defined at the introduction chapter and the conclusion chapter discussed whether all of them have been successfully achieved. The limitation and future enhancements were discussed.

# References

Barai, M.K., 2021. Sentiment Analysis with TextBlob and Vader. Analytics Vidhya. URL https://www.analyticsvidhya.com/blog/2021/10/sentiment-analysis-with-textblob-and-vader/ (accessed 10.30.23).

Dhanda, M., Verma, V., 2016. Recommender System for Academic Literature with Incremental Dataset. Procedia Computer Science 89, 483–491. https://doi.org/10.1016/j.procs.2016.06.109

Ijaz, F., n.d. Book Recommendation System using Machine learning.

Machine Learning Project | Classification | Sentiment Analysis | Sinhala - YouTube [WWW Document], n.d. URL https://www.youtube.com/playlist?list=PL495mke12zYDPRGhXd6JGY5EUoksIVwYU (accessed 10.29.23).

Marappan, R., 2022. Create a Book Recommendation System using Collaborative Filtering. IJMEBAC 1, 44–46. https://doi.org/10.31586/ijmebac.2022.341

Mercy Milcah Y, Moorthi K, Jansons Institute of Technology, 2020. AI based Book Recommender System with Hybrid Approach. IJERT V9, IJERTV9IS020416. https://doi.org/10.17577/IJERTV9IS020416

Murali, M.V., Vishnu, T.G., Victor, N., 2019. A Collaborative Filtering based Recommender System for Suggesting New Trends in Any Domain of Research, in: 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS). Presented at the 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), IEEE, Coimbatore, India, pp. 550–553. https://doi.org/10.1109/ICACCS.2019.8728409

Sarma, D., Mittra, T., Shahadat, M., 2021. Personalized Book Recommendation System using Machine Learning Algorithm. IJACSA 12. https://doi.org/10.14569/IJACSA.2021.0120126

Shah, K., 2019. Book Recommendation System using Item based Collaborative Filtering 06.

Tian, Y., Zheng, B., Wang, Y., Zhang, Y., Wu, Q., 2019. College Library Personalized Recommendation System Based on Hybrid Recommendation Algorithm. Procedia CIRP 83, 490–494. https://doi.org/10.1016/j.procir.2019.04.126

Wadikar, D., Kumari, N., Bhat, R., Shirodkar, V., 2020. Book Recommendation Platform using Deep Learning 07.

Wang, D., Liang, Y., Xu, D., Feng, X., Guan, R., 2018. A content-based recommender system for computer science publications. Knowledge-Based Systems 157, 1–9. https://doi.org/10.1016/j.knosys.2018.05.001