2022

TEAM
CHALLENGERS

Internship Project Report on

Book Recommendation System Using IBM Watson

SUBMITTED BY

NAGARAJA S ARALIKATTI   4BD19CS402

SHIVAKUMAR K P   4BD18CS088

NARENDRA B O   4BD19CS403

LIKHITH P   4BD18CS038

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BAPUJI INSTITUTE OF ENGINEERING AND

TECHNOLOGY DAVANAGERE – 577004

ACADEMIC YEAR : 2021-2022

CONTENTS

CHAPTER 1

INTRODUCTION

Recommendation systems are widely used today to recommend products to users based on their interests. A recommendation system is one of the strongest systems for increasing profits by retaining more users in a very big competition. In this article, I'll walk you through how to build a book recommendation system with Machine Learning using the Python programming language.

1.1 OVERVIEW

In this age of information overload, people use a variety of strategies to make choices about what to buy, how to spend their leisure time, and even where to go. Recommendation systems automate some of these strategies with the goal of providing affordable, personal and high-quality recommendations. So, with the help of different techniques like collaborative and content-based filtering, we can build our own recommendation systems. The main objective of the project is to build an application where the user is prompted with various types of book recommendations based on the input given by the use. A recommendation engine is a class of machine learning which offers relevant suggestions to the customer. Before the recommendation system, the major tendency to buy was to take a suggestion from friends. But Now Google knows what news you will read, Youtube knows what type of videos you will watch based on your search history, watch history, or purchase history. A recommendation system helps an organization to create loyal customers and build trust by them desired products and services for which they came on your site. The recommendation system today are so powerful that they can handle the new customer too who has visited the site for the first time. They recommend the products which are currently trending or highly rated and they can also recommend the products which bring maximum profit to the company.

1.2 PURPOSE

The main objective of the project is to build an application where the user is prompted with various types of book recommendations based on the input given by the user. You will be able to know how to pre-process / clean the data using different data pre-processing techniques.
You will able to analyse or get insights of data through visualization. Applying algorithms according to dataset and based on visualization. You will able to know how to find accuracy of the model. You will be able to know how to build a web application using Flask framework.

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING PROBLEM

To complete this project, you should have the following software's and packages
Anaconda Navigator:

Anaconda Navigator is a free and open-source distribution of the Python and R programming languages for data science and machine learning related applications. It can be installed on Windows, Linux, and macOS.Conda is an open-source, cross-platform, package management system. Anaconda comes with so very nice tools like JupyterLab, Jupyter Notebook, QtConsole, Spyder, Glueviz, Orange, Rstudio, Visual Studio Code. For this project, we will be using Jupiter notebook and spyder.

2.2 PURPOSED SOLUTION

The method or solution is Jupiter notebook and spyder we used to complete this project. and you will use this jupiter notebook for you recommended.

To build Machine learning models you must require the following packages

Sklearn: Scikit-learn is a library in Python that provides many unsupervised and supervised learning algorithms.

NumPy: NumPy is a Python package that stands for 'Numerical Python'. It is the core library for scientific computing, which contains a powerful n-dimensional array object

Pandas: pandas is a fast, powerful, flexible, and easy to use open source data analysis and manipulation tool,built on top of the Python programming language.

Matplotlib: It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits

CHAPTER 3

THEORETICAL ANALYSIS

3.1 BLOCK DIAGRAM

In this Block Diagram Book Recommendation System Using IBM watson, says recommended by users, for example in Institute or College have college libraries is there, in library have lot of books and different Departments books are stored. the college students have college library ID card must and should, those students are goto college library and what the students want Books they access it come back to class, Some Students don't have college library ID cards.
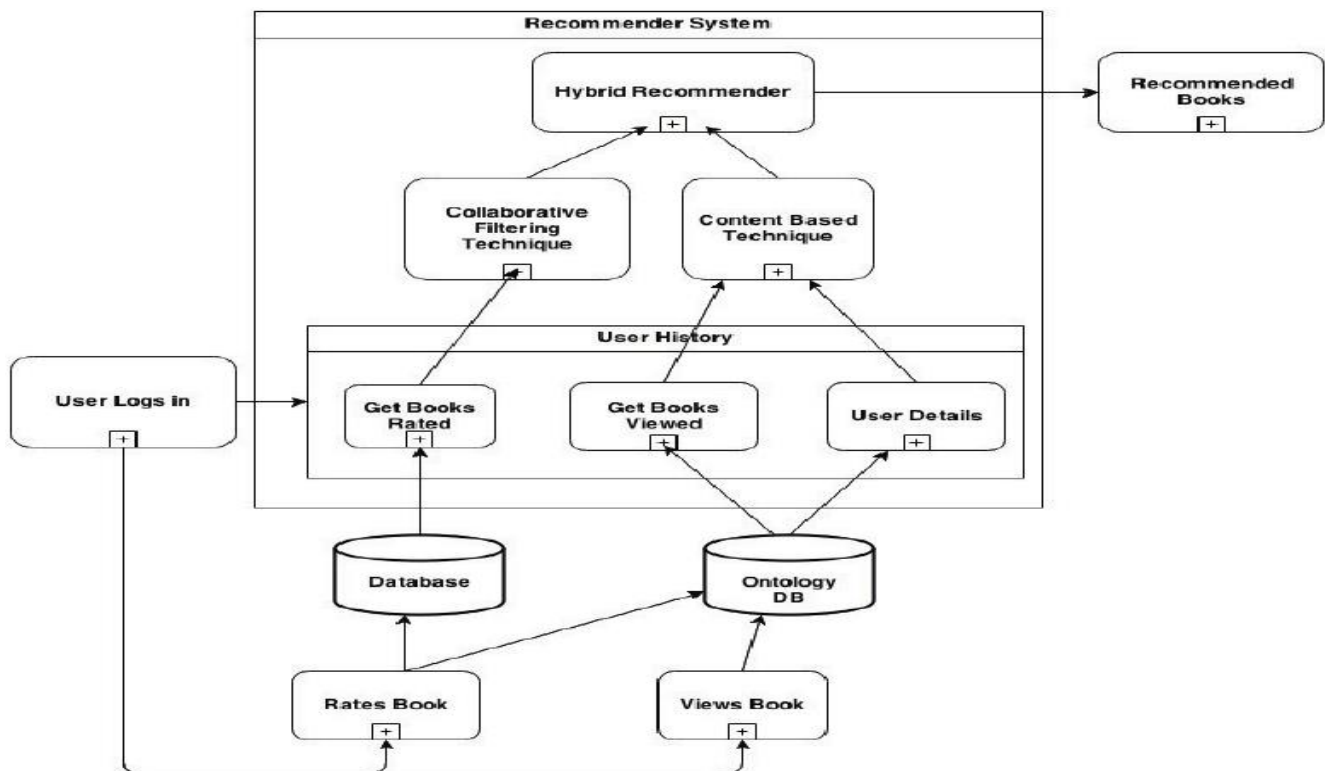

Fig3.1 Block diagram of Book Recommendation System Using IBM Watson.

because those students are joined colllege later, those students wants books. that situation students are use the college lecture library ID card and access the class Books. this is the recommended of there class teachers to access there class books.   In diagram Book recommended of both Users. in this diagram have two Users are there, the User is Accessed the that book and read it, after the Similar User are read the same book, before taking the recommended by first user, because the first user is already read that book. that's why the second user is take the recommendation and read the book, recommendation is how the book is good or bad, and good things or not and what is story

3.2 HARDWARE / SOFTWARE DESIGNING

The hardware required for the development of this project is:

Processor   : Intel CoreTM i5-9300H
Processor speed : 2.4GHz
RAM Size   : 8 GB DDR
System Type   : X64-based processor

SOFTWARE DESIGNING:

The software required for the development of this project is:

Desktop GUI   : Anaconda Navigator
Operating system   : Windows 10
Front end   : HTML, CSS,JAVASCRIPT
Programming   : PYTHON
Cloud Computing Service : IBM Cloud Services

CHAPTER 4

EXPERIMENTAL INVESTIGATION

IMPORTING AND READING THE DATASET

Importing the Libraries
First step is usually importing the libraries that will be needed in the program.
Pandas: It is a python library mainly used for data manipulation.
NumPy: This python library is used for numerical analysis.
Matplotlib and Seaborn: Both are the data visualization library used for plotting graph which will help us for understanding the data.
csr_matrix() :A dense matrix stored in a NumPy array can be converted into a sparse matrix using the CSR representation by calling the csr_matrix() function.
Train_test_split: used for splitting data arrays into training data and for testing data.
Pickle: to serialize your machine learning algorithms and save the serialized format to a file.

Reading the Dataset

For this project, we make use of three different datasets (Books_Ratings, Books, Users). We will be selecting the important features from these datasets that will help us in recommending the best results.

The next step is to read the dataset into a data structure that's compatible with pandas. Let's load a .csv data file into pandas. There is a function for it, called **read_csv().**We will need to locate the directory of the CSV file at first (it's more efficient to keep the dataset in the same directory as your program).If the dataset in same directory of your program, you can directly read it, without any path. After the next Steps we made following bellow:

1.Data visualization

2.Collabrative and filtering

3.Creating the Model

4.Test and save the model

5.Buil Python Code

6.Build HTML Code

7.Run the Application

 We are the following above sections we did and investigate it.
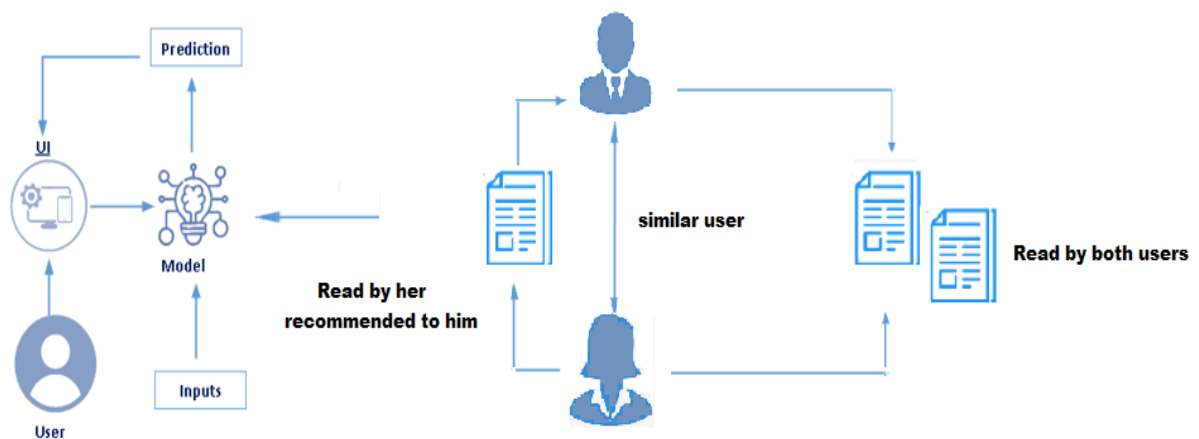
CHAPTER 5

FLOWCHART

Fig 5.1 Flowchart of the project

Project Flow:

- User interacts with the UI (User Interface) to upload the input features.
- Uploaded features/input is analyzed by the model which is integrated
- Once model analyses the uploaded inputs, the prediction is showcased on the UI.

1. Data Collection.

- Collect the dataset or Create the dataset

2. Data Pre- processing.

- Import the Libraries.
- Importing the dataset.
- Exploratory Data Analysis
- Data Visualization

3. Collaborating Filtering

- Merging datasets
- Creating the Model
- Predicting the results
- Saving our model and dataset

4. Application Building

- Create an HTML file
- Build a Python Code
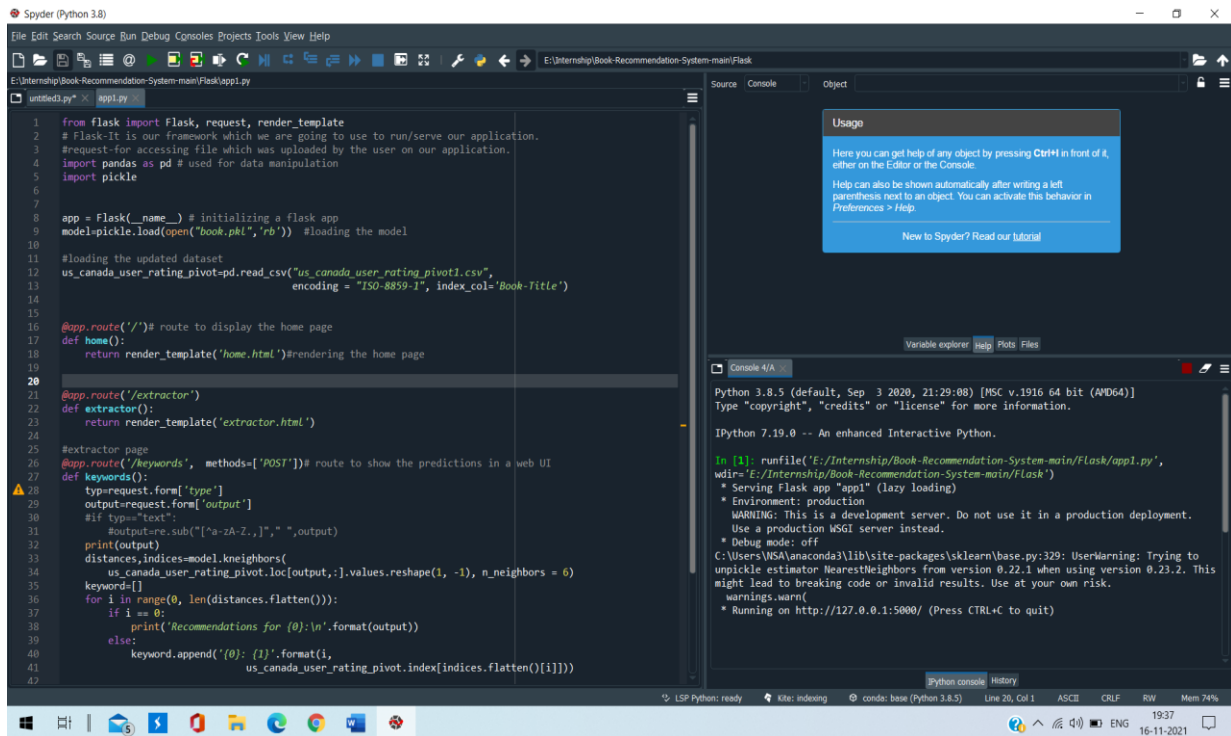
CHAPTER 6

RESULT
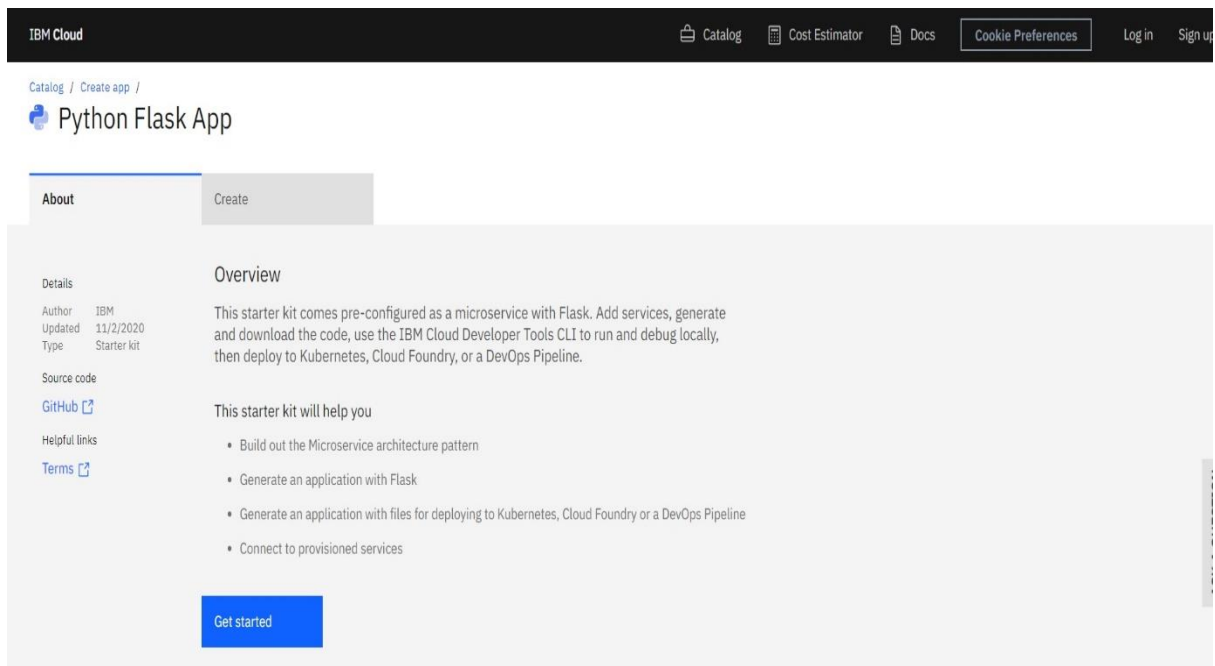

Fig 6.1 Flask App Code with Output Page

Fig 6.2 Uploading the Project of Book Recommendation System using IBM Watson, In IBM
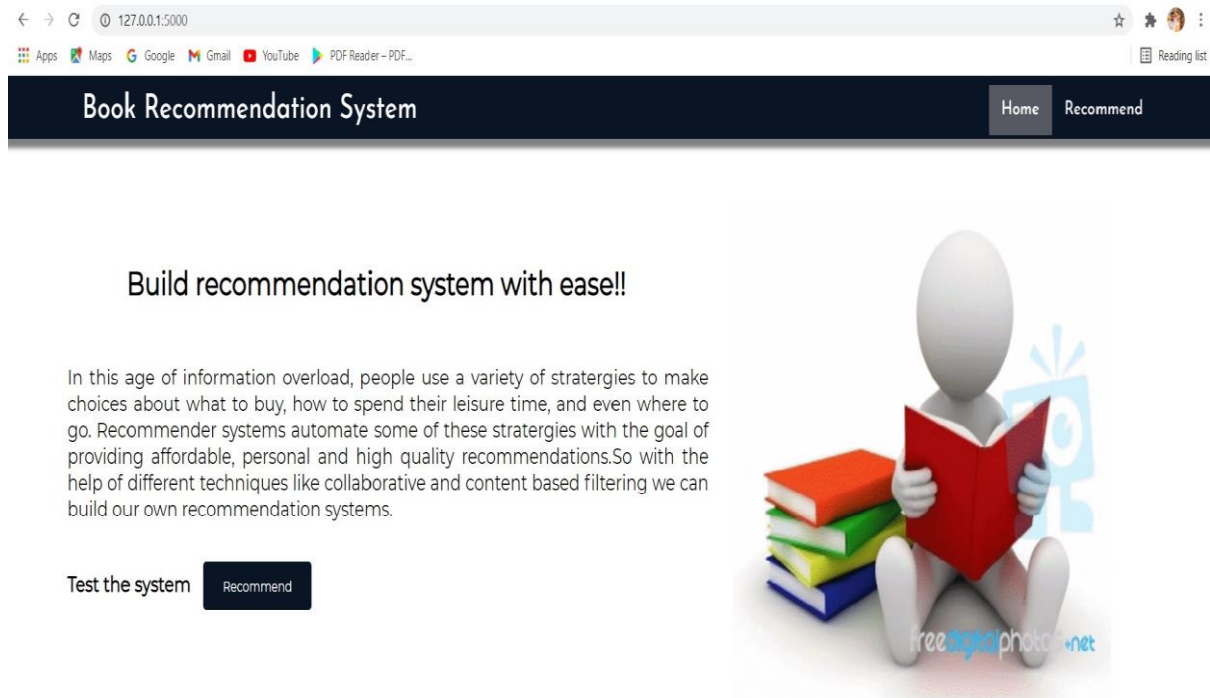
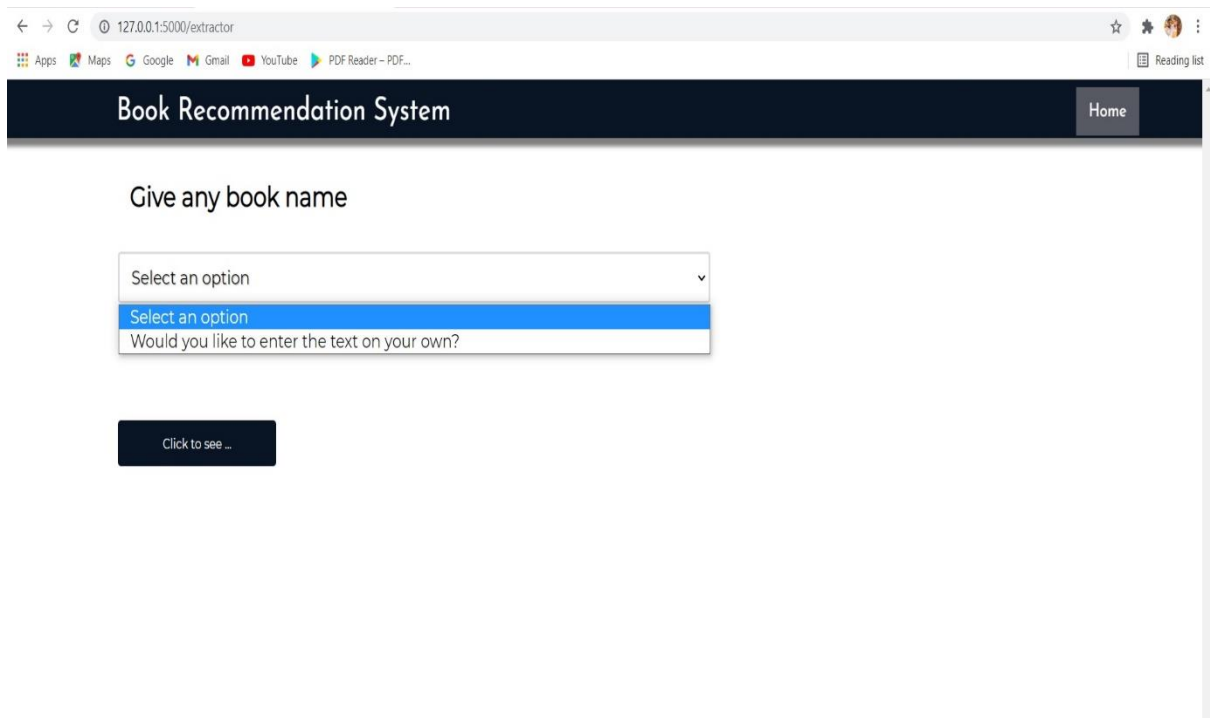Fig 6.3 Home Page of Book Recommendation System using IBM Watson

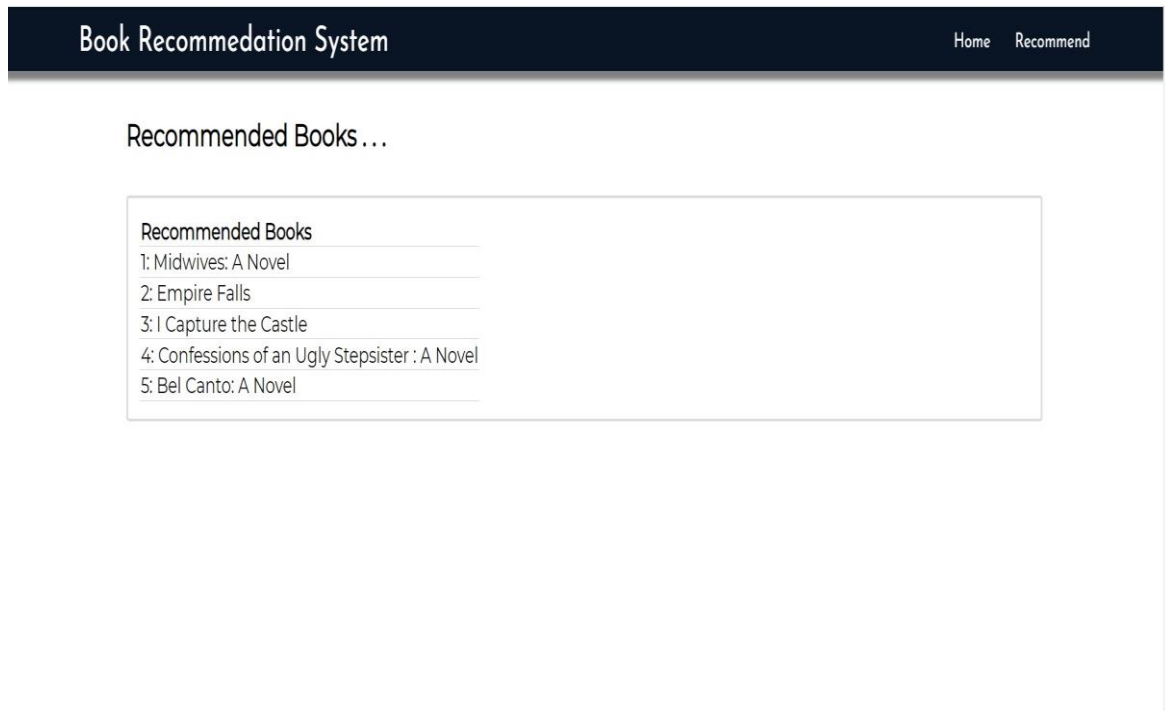Fig 6.4 Extractor Page of Book Recommendation System using IBM Watson

Fig 6.5 Keywords Page of Book Recommendation System using IBM Watson

CHAPTER 7

ADVANTAGES AND DISADVANTAGES

ADVANTAGES

- advantage of recommender systems is that they provide personalization for customers of e-commerce, promoting one-to-one marketing. Amazon, a pioneer in the use of collaborative recommender systems, offers "a personalized store for every customer" as part of their marketing strategy.
- Today the majority of the recommendation systems are based on machine learning, so its main disadvantages partially correlate with the usual issues we face during typical machine learning development, but are still slightly different. Let's have a closer and a more dedicated look.

1. Revenue and sales increase

2. User satisfaction growth

3. Turnover increase

DISADVANTAGES

Problems of Recommender Systems

- Lack of Data. Perhaps the biggest issue facing recommender systems is that they need a lot of data to effectively make recommendations. ...
- Changing Data. ...
- Changing User Preferences. ...
- Unpredictable Items. ...
- This Stuff is Complex!

CHAPTER 8

APPLICATIONS

The applications of recommender systems include **recommending movies, music, television programs, books, documents, websites, conferences, tourism scenic spots and learning materials**, and involve the areas of e-commerce, e-learning, e-library, e-government and e-business services.

1. Travel and tourism recommendation system

2. Education recommendation system

3. Book recommendation system

4. Movie recommendation system

CHAPTER 9

CONCLUSION AND FUTURESCOPE

CONCLUSION

Recommender systems are a powerful new technology for extracting additional value for a business from its user databases. These systems help users find items they want to buy from a business. Recommender systems benefit users by enabling them to find items they like. Conversely, they help the business by generating more sales. Recommender systems are rapidly becoming a crucial tool in E-commerce on the Web. Recommender systems are being stressed by the huge volume of user data in existing corporate databases, and will be stressed even more by the increasing volume of user data available on the Web. New technologies are needed that can dramatically improve the scalability of recommender systems.

In this paper we presented and experimentally evaluated a new algorithm for CF-based recommender systems. Our results show that item-based techniques hold the promise of allowing CF-based algorithms to scale to large data sets and at the same time produce high-quality recommendations.

FUTURESCOPE

In the final chapter, I will wrap up by giving the summary of the book and the topics covered. We will focus on the future scope of the research that you will have to undertake. Then we will provide a brief introduction to the current research topics and advancements happening in the field of recommendation systems. I will also list book references and online resources during the course of this book.

CHAPTER 10

BIBILOGRAPHY

1. [ABE 09] ABEL M.-H., LEBLANC A., "Knowledge sharing via the E-MEM0RAe2.0 platform", 6th International Conference on Intellectual Capital, Knowledge Management & Organisational Learning, Montreal, Canada, pp. 10-19, October 2009.

2. [ADE 14] ADENIYI D., WAI Z., YONGQUAN Y., "Automated web usage data mining and recommendation system using k-nearest neighbor (KNN) classification method", Applied Computing and Informatics, 2014.

3. [ADO 05] ADOMAVICIUS G., TUZHILIN A., "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions", IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 6, pp. 734-749, 2005.

4. [ADO 08] ADOMAVICIUS G., TUZHILIN A., "Context-aware recommender systems", Proceedings of the 2008 ACM Conference on Recommender Systems (RecSys '08), New York, NY, ACM, pp. 335-336, 2008.

5. [AGR 93] AGRAWAL R., IMIELIŃSKI T., SWAMI A., "Mining association rules between sets of items in large databases", ACM SIGMOD Record, vol. 22, no. 2, pp. 207-216, June 1993.

APPENDIX

A Source Code of Flask:

```
from flask import Flask, request, render_template
# Flask-It is our framework which we are going to use to run/serve our application.
#request-for accessing file which was uploaded by the user on our application.
import pandas as pd # used for data manipulation
import pickle


app = Flask(__name__) # initializing a flask app
model=pickle.load(open("book.pkl",'rb'))  #loading the model
#loading the updated dataset
us_canada_user_rating_pivot=pd.read_csv("us_canada_user_rating_pivot1.csv",
                      encoding = "ISO-8859-1", index_col='Book-Title')


@app.route('/')# route to display the home page
def home():
    return render_template('home.html')#rendering the home page


@app.route('/extractor')
def extractor():
    return render_template('extractor.html')


#extractor page
@app.route('/keywords',  methods=['POST'])# route to show the predictions in a web UI
def keywords():
    typ=request.form['type']
    output=request.form['output']
```

```python
#if typ=="text":

    #output=re.sub("[^a-zA-Z.,]"," ",output)

print(output)

distances,indices=model.kneighbors(

    us_canada_user_rating_pivot.loc[output,:].values.reshape(1, -1), n_neighbors = 6)

keyword=[]

for i in range(0, len(distances.flatten())):

    if i == 0:

        print('Recommendations for {0}:\n'.format(output))

    else:

        keyword.append('{0}: {1}'.format(i,

                    us_canada_user_rating_pivot.index[indices.flatten()[i]]))


  # showing the prediction results in a UI

  return render_template('keywords.html',keywords=keywords)



if __name__ == "__main__":

  # running the app

  app.run(debug=False)
```