ISY5001 IRS

NATIONAL UNIVERSITY OF SINGAPORE

ISS-IS

AI-BASED BOOK GENRE DETECTION SYSTEM

Author:

ChenHaoyang

HuangRunxiang

YangWenkai

Course Lead:

Sam

October 26, 2022

Contents

List of Figures

List of Tables

1 Introduction

Book genre classification is one of the popular sections among all NLP areas back in time, and it is widely used in many situations such as libraries, book donation charities, or even simply data-mining programs. There are basically three ways of determining the genre of a selected book.

- Classification with Titles
- Classification with Contents
- Classification with Covers

Do note that the classification with covers can be divided into two sub-classes which are text recognition which ultimately belongs to Natural Language Processing area and Image Classification which belongs to the Computer Vision area.

1.1 Aims and Objectives

On the premise that the current book classification system has been able to effectively classify books according to book information, our goal is to develop a system that can use limited data to classify books as accurately as possible and achieve our goals at a relatively faster speed. The system should also be able to read the cover of the book and 'grade' the book with its cover on scale from 0-4, the score implies the possible degree of popularity of the input book.

1.2 Current Methods

Nowadays, more libraries will use RFID system when carrying out book management, RFID is the abbreviation of Radio Frequency Identification. The principle is the non-contact data communication between the reader and the tag to achieve the purpose of identifying the target. The specific operation is to enter the information of the newly entered library into the RFID tag, paste it on each book, and transmit the information such as the storage location of the book information to the background management database. The service life of RFID tags is longer than that of ordinary bar-codes and magnetic strips. But this method also has its drawbacks. In certain situations, such as large-scale donated books, if you need to add manual tags for classified storage, there will be a lot of work. And our system hopes to be able to complete the classification of books with lower labor costs in this situation.

2 Business Case

2.1 Problem Statement

As mentioned above, most of the libraries in Singapore, such as Orchard Road Library, National Library of Singapore, etc., use RFID systems for comprehensive book management. Although this management method is more efficient and easy to manage, it increases a lot of time costs in certain circumstances. If you encounter a one-time large-scale book storage situation such as book donation, a more efficient and data-saving classification method would be a good solution.

Also, the grading function is really useful for book classification, we know that all the libraries and bookstores have the shelve of best-sellers and 'newly come out' books. Many times bookstores and libraries randomly choose from or simply place all the new books they got on the 'newly come out' shelve which is not efficient and accurate for recommending books to the readers and customers. If the shop or library can predict how popular this new book would be in the future, it will be helpful for their interests and will be better for the customers and readers to choose from tons of books.

2.2 Proposed Solution

We developed an AI-Based book classification and rating system as mentioned in the introduction part to solve the problems stated above. Bookstores or libraries can easily use this system to classify and predict the ratings of the new books they got with only the titles and covers of the certain book. And the functions are not tied to each other which means that the users can also only use one of them or combine both to get the information they need for their demands. Also, our function doesn't limit to book titles, the input of an introduction part of the book will result in more precise classifications.

With the use of our system, bookstores and libraries can classify and predict books in a much easier way which doesn't need to label all the books by human and the prediction function can help them increase their profit and the reading and buying experience of readers and customers.

2.3 Market Analysis

Libraries and bookstores have extremely high penetration and access rates around the world. And according to the survey and research, the number of books read and published around the world in recent years is still showing an upward trend. Although there was a decline in 2020 due to the epidemic, the prospect of the book market is still very optimistic as the normalization of life due to the epidemic continues worldwide. Moreover, our system does not conflict with the RFID system that is widely used now. Our system can be used as a branch solution of the RFID system in special cases, so we do not have a competitive relationship with the current mainstream system.

3 Product Details

3.1 Member Roles

Name	ID	Role
Yang Wenkai	A0261636A	Backend Application Deveopment, UI/UX Development
Huang Runxiang	A0261627A	Backend Application Deveopment, UI/UX Development
Chen Haoyang	A0261905E	Backend Application Deveopment, Algorithm/Model Development and Training

Table 1: Role Table

3.2 Main Features

To reach the goal of labeling books with higher efficiency and lower cost, and at the same time help better set the shelve organizations, we developed the system with 2 functions.

- Function 1. Book Classification with Title

  The libraries and bookstores can easily be able to classify the books with different genres without manually label them with but not restrained with the input of book titles, book introductions or even reviews are tested good in our model.

- Function 2. Book Rating Prediction with Cover

  Libraries and bookstores might encounter problems dealing with new books coming in their stock, With our system, the users can simply be able to predict the 'potentials' of a book, so that they can choose certain books to place at more feasible and visible shelves for a better sale of the book.

- Function 3. Book Classification with Cover

  Similar to function 1, this is also a book classification function but with book covers the user uploads.

3.3 System Architecture

The chart below shows the general Architecture of our system. There are mainly two parts of our system

- The interactive webpage - Our system is reflected on one web page for the user to access, all two functions are on the same page.
- Backend System - The web page is supported by the backend system with API interface corresponding to the functions

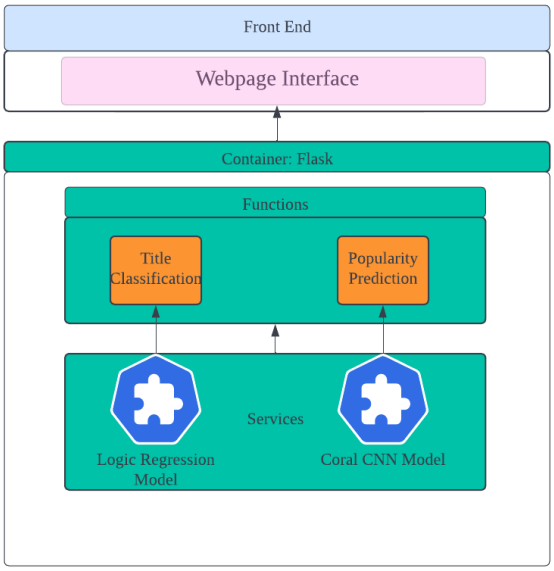

Figure 1: Overall Structure

4 Modeling Details

4.1 Book Classification with Titles

4.1.1 Data Preparation

For this function, we want to classify books by their titles through a machine learning model, and data preparation is a more difficult task. A data-set that is too small will lead to a weakened model's over-fitting and generalization ability, while a data-set that is too large will consume too much time for training.

We firstly wanted to use the Goodreads data-set from Kaggle which has 5000 books with their titles and genres, then we assumed that the data-set might be too small for the model to convergent. So we finally chose the data-set from Uchida Lab which contains around 200,000 book data divided into 30 classes. The labels are ISBN, Book Cover Link, Title, Author, Class, and Genre. This amount of data will be definitely enough for us to use for our aim, and for data size, we also tested a smaller subset from this data set to see the performance difference and it turns out that the whole data set works better than a smaller subset.

No.	Feature Name	Feature Discription	Values
1	ISBN	The ISBN No of the Book	e.g 1623439671
2	Title	Title of the book	e.g Animal Farm
3	Author	Author of the book	e.g Steven King
4	Image Link	url for the cover image	A link
5	Class	number of class of the book	Numerical
6	Genre	Genre of the book	text

Figure 2: Data Details

4.1.2 Text Preprocessing

It is vital to choose a good way to preprocess the text data before getting them into the models, some general preprocessing ways includes Removing Stop words, Lower casing, Tokenization, Stemming and Lemmatization as shown in figure 3. For our case we mainly used normalizing, tokenizing, removing stop words and

lemmatizing to preprocess our data.

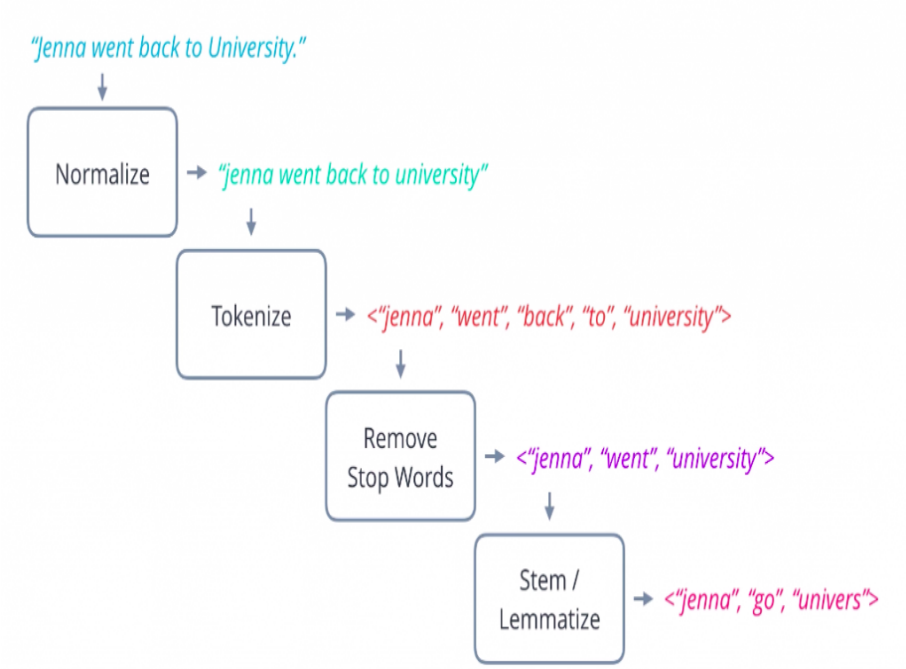

Figure 3: Preprocess Steps From Google Image

4.1.3 Text Data Vectorization

Word vectors represent text in a way that is easier for computers to understand. This text representation should be able to facilitate comparisons, calculations, etc. between texts. And in our case we compared BOW and TF-IDF and chose TF-IDF as our final model of word embedding, TF-IDF stands for term frequency–inverse document frequency. The TF-IDF algorithm reflects the importance of each word by assigning a weight. According to the weight, all words in an article are sorted from high to low. The higher the weight, the higher the importance, and the top words are can be used as keywords for this article.

$$TFIDFi, j = TermFrequencyi, jlog2\frac{(D)}{(DocumentFrequency_i)} \tag{1}$$

4.1.4 Algorithm and Model Training

We tested our data on multiple algorithms such as Gaussian NB, Logistic Regression, Neural Network and SVM to search for the best working model for our project aim, and even with different word embedding methods: Word of Bag and TFIDF. From which the TFIDF Logic Regression model works best. And below table 4 shows how the selected models worked during our training.

Algorithm	Training	Validation
Gussain NB	.5132	.6213
Logic Regression	.5901	.6746
Neural Network	.3550	.5293
SVM	.5018	.5901

Table 2: Algorithms Performance

It can be easily interpreted from the table that the Logic Regression works best in both training and validation set, but we kept in mind that accuracy alone is not enough to fully illustrate how well the model performs, so we tested the model with multiple book titles with or without book description or introduction of the book to see if the model can handle unseen data, some example outputs are shown in table 4.

Input	Output
China Tourism Guide	Travel
Snow White	Children
Tennis Master	Sports&Outdoors

Table 3: Example outputs

4.1.5 Modeling Conclusion and Take Away

As stated above, the Logic Regression model performs best for out task so we chose it as the model we use for classifying books with their title or introduction provided, and among two ways of word embedding, TF-IDF performs better than Bag of Word (state as BOW). That might because in BOW, the representation of the document vector only uses the word frequency information, ignoring the importance information of the word itself, and then the information of the word itself is also very important. For example, the word ”go” appears many times in a document, but the word is not very meaningful to the information of a document, and some documents ”subject words” may only appear once or twice but can fully express meaning of the document.

Another thing we took away from this experiment is that although in general the more complex the model, the better the model will be. However, different problems still require specific analysis of specific problems.

In solving machine learning problems, there is no one-size-fits-all model, and it is often more difficult to in-

terpret the problem to be solved than to understand the model. We generally consider the following points when selecting models in future problems: The first is training time: as the observed data changes, the model needs to be retrained, and the time to train the new model should not be too long; prediction time: when the model is online and working , the time required to predict the score on the newly input data; model storage: how much memory space is required for the model to run.

4.2 Book Genre Prediction Using Covers

4.2.1 Data Preparation

For this function, we are using the same data set we obtained from Uchida Lab as the previous function, the key point is to try to keep the data spread evenly so that none of the classes has way more data than others. But as for us the 200,000 data size is too large for us to manage and modify, after a few trails with different models such as CNN, RNN, CoralCNN, we finally gave up using the whole data set and decided to use a rather smaller data size of around 50,000 which is also a subset of the Uchida Lab data set.

No.	Feature Name	Feature Discription	Values
1	ISBN	The ISBN No of the Book	e.g 1623439671
2	Title	Title of the book	e.g Animal Farm
3	Author	Author of the book	e.g Steven King
4	Image Link	url for the cover image	A link
5	Class	number of class of the book	Numerical
6	Genre	Genre of the book	text

Figure 4: Data Details

4.2.2 Image Preparation

As mentioned above we used the dataset from Uchida Lab, and we found out the 32 labels classification is way too much for the model to train as some of the labels are pretty much imbalanced compared to others. And we dive deeper into this problem and found out that when there is a large difference in the amount of data

between classes, the model tends to discriminate the sample as a sample with a large amount of data, so that a lower loss can be obtained, and there is no motivation to continue optimizing the parameters. Extreme example could be: In the training sample, there is 1 sample in class A and 100 samples in class B, then the model can obtain a very low loss value by discriminating the sample as class B, thus losing the motivation to continue training and optimization, and the network performance is weak. So finally we merged the data into 5 genres and chose some samples from the whole data set as our subset to help the model perform better.
For this part we used ImageDataGenerator from Keras to read in our images and generate the digitalization data for our model.

4.2.3 Algorithm and Model Training

For the algorithm part we firstly tried a two-layer CNN model with a flatten layer with loss function as categorical cross-entropy, but the accuracy on validation set stopped at around .30, then we noticed that a dropout layer might be helpful so we added a dropout layer with the parameter set to .4 but the performance of our model seems had no improvement. Then we decided to add layers to the network to see if this can help. We added two convolutional layers and the accuracy went up to .35 as we hoped, and we kept trying to make the model perform as good as it could. We also tried data augmentation, EfficientNet, Coral CNN and after all the trials the CNN Model without Early Stop and Larger Dropout Layer stands out to be the best one. Then we tested our model with some book cover images, some sample output are shown below.

Algorithm	Test Acc	Valid Acc
2-Layer CNN	.80192	.30123
CNN with Dropout	.84165	.30112
More Layer CNN	.63912	.36115
EfficientNet with Data Augmentation	N/A	.2125
CNN with larger Dropout	N/A	.4562

Table 4: Models & Performance

4.2.4 Modeling Conclusion and Take Away

We tried our best to train a model with good performance with out data but it turned out that no matter what model we use and no matter how we preprocess the data, the overall accuracy is still below .50, which is relatively bad compared to our text model, but it still can predict with some accuracy which is still acceptable.

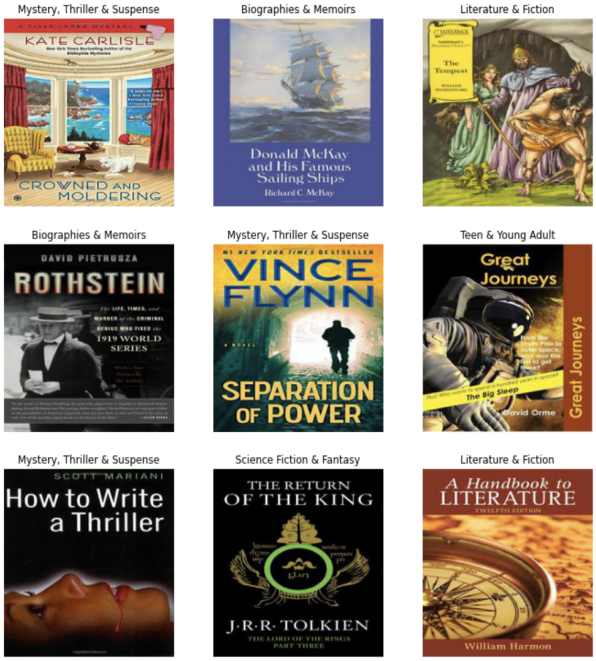

Figure 5: Sample output of model

4.3 Book Cover Grading

4.3.1 Data Preparation

For this function we are using the book data from goodreads website, goodreads is book website with tons of book informations with their rankings, so in order to make prediction of book rankings we managed to download the information of books from goodread. And this data set is obtained from Kaggle Best Books Ever. There are around 54000 samples of book in the set and it is sorted by popularity (A.K.A grades) which is perfectly for us to use.

book_authors	book_desc		book_format		book_pages
Suzanne Collins	Winning will make you famous.		Hardcover		374 pages
book_rating	book_rating_count	book_review_count	book_title	genres	image_url
4.33	5519135	160706	The Hunger Games	Young Adult—Fiction—	Image url

Table 5: Sample Data

4.3.2 Image Preparation

This data contains images no diffrence than what we got from the previous mentioned Uchida Lab data set in terms of form, most of the book cover are RGB image with different sizes, so the first thing we did is to reshape

the images into same scale. We split the data evenly into 5 categories based on their popularity.

4.3.3 Algorithm and Model Training

The algorithm we chose was Coral CNN which was originally applied to predict the age of humans in pictures.The authors chose the ResNet-34 architecture, a modern CNN architecture that has achieved good performance on various image classification tasks. But after several trails, our model is kind of over-fitting that it is highly possible it will output tier 1 as the output(which means the model thinks the cover is really good), but the good news is that our model still can tell whether the book cover is good or not in some degree such as the possibility grades.

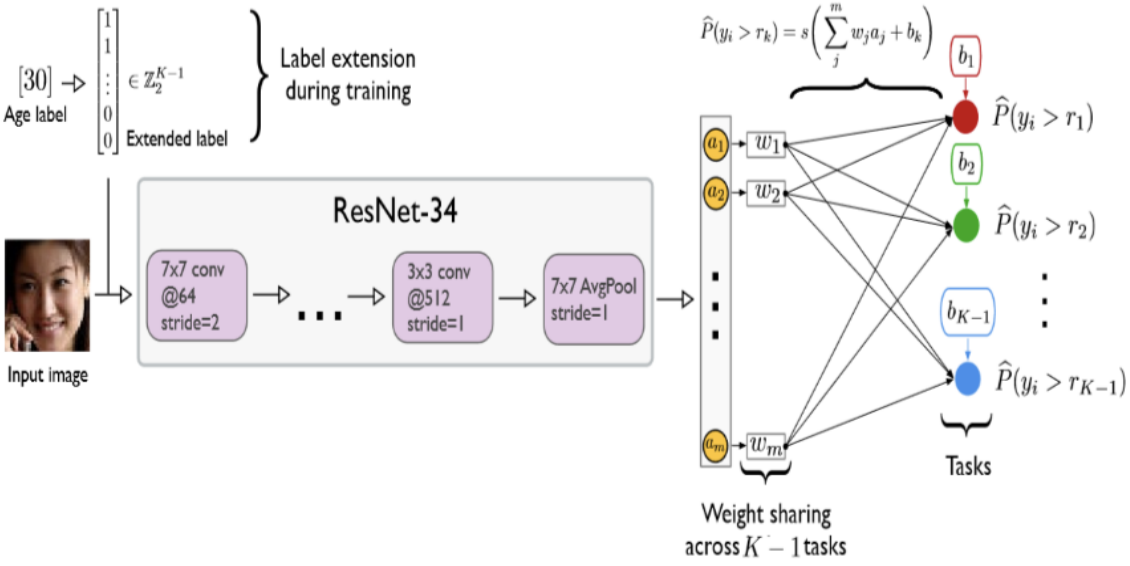

Figure 6: Coral CNN Architecture

4.3.4 Modeling Conclusion and Take Away

Although it seems like our model is a little over-fitting but it still can perform the task we want it to do, in future we might try more methods to improve the performance of this model, potential ways might be adding dropout, data augmentation, collecting more data .etc.

5 System Development & Frontend

The system frontend is based on the basic html file with Javascript functionalities then using Flask as the backend framework connection to achieve our system's usability. Thus, this two parts will be further explained

in this chapter.

5.1 Web page User interface Design

According to the goal and aim of our system, this system does not include a login page for convenient usage. Thus, the system user can directly get their demanded results by providing the required inputs. As our system has three main functionalities that return results through book cover images or book names and descriptions which need different data input handlers for the text and image input separately.

Hence, the system's UI includes an image upload field and a text box for the users to provide their input, most importantly, these two handlers are separated which can provide results independently by clicking the submit button. For image input, the system will return the book's popularity and rough category under the book cover image. Besides, it can also return a more subdivided category of the book at the bottom of the web page if the user inputs the name (description) of the book. These two functionalities can work independently or collaboratively.

5.2 Flask Framework usage

The Flask Framework is a very convenient can efficient toolkit for building HTTP APIs especially for Python coding. Although it is a generally youthful framework, it is considered suitable for this system with the following reasons:

- **Flask has a shallow learning curve:** understanding the concept of Flask is not a complex task. Hence, even the fresh developers can figure out how it works in a short time.
- **Flask is easy to use and flexible:** it allows customizing the code according to the requirements. There are a few parts that cannot be changed or altered.
- **Flask is highly scalable:** the system now is a small web app idea, however, Flask allows easy scaling as per the requirements. It fits the goal and is helpful for libraries and bookstores to merge and upgrade.

6 Conclusion and Limitation

In summary, our model achieves the three goals we set out at the beginning, and achieves good results. Our book classification function by title performs the best among the three functions, which we believe is due to the

relatively large amount of data and adequate preparation in the text preprocessing stage. The performance of the two computer vision models is not as good as expected. We believe that this is because our data is not high-quality and the model used is not the most suitable model. And our experience with computer vision direction is still lacking compared to NLP direction. The biggest limitation of our model is that since the data used by the text model and the image model are not the same, the output of the model is not the same, and sometimes the cover and title of the same book may return different results. But this problem is controllable, because our experiments have proved that the results returned by the cover and the results returned by the title are often subsets of each other. For example, the cover of Harry Potter would be classified as science fiction, and the title would be classified as children's book, there is no conflict between the two.

7 Future Work

Our outlook for the future of our project is as follows. First of all, we hope to find data that satisfies the two models of computer vision and word processing at the same time, hoping to use the same features and achieve the effect of the two models outputting the same results as much as possible. Secondly, we hope that our computer vision model can be optimized so that the accuracy of the model can reach a relatively high value, and try to avoid the occurrence of overfitting.

8 Reference and Datasets

Book Title Classification and Book Cover Classification

https://github.com/uchidalab/book-dataset

Book Cover Grades Predication

https://www.kaggle.com/datasets/meetnaren/goodreads-best-books

Wenzhi Cao, Vahid Mirjalili, Sebastian Raschka (2020): Rank Consistent Ordinal Regression for Neural Networks with Application to Age Estimation. Pattern Recognition Letters.

9 Appendix

9.1 Project Proposal

GRADUATE CERTIFICATE: Intelligent Reasoning Systems (IRS)
PRACTICE MODULE: Project Proposal

Date of proposal: October 2022
Project Title: ISS Project – Book Ranking Category recognition system, for intelligent course scheduling system use case
Sponsor/Client: *(Name, Address, Telephone No. and Contact Name)* Institute of Systems Science (ISS) at 25 Heng Mui Keng Terrace, Singapore NATIONAL UNIVERSITY OF SINGAPORE (NUS) Contact: Mr. GU ZHAN / Lecturer & Consultant Telephone No.: 65-6516 8021 Email: zhan.gu@nus.edu.sg
Background/Aims/Objectives: The proposed intelligent Book categories recognition system will make use of various advanced machine learning techniques and components to deliver a system for library or other usage and provide help on automatic book category recognition.
Requirements Overview: • Research ability • Programming ability • System integration ability
Resource Requirements (please list Hardware, Software and any other resources) Hardware proposed for consideration: • GPU, personal computer, or laptop, etc. Software proposed for consideration: • Python coding environment and tools such as PyCharm, VSCode etc • Pertained machine learning models, e.g. TFIDFVectorizor, BOW, NLP, BERT, Word2Vec, etc • Deep learning tools, e.g. Python Keras, Tensorflow, Scikit-Learn, PyTorch etc • Cloud computing/server, e.g. Google, IBM, Azure, etc. • Front-end and back-end tools/structure, e.g. JavaScript, Flask, etc
Number of Learner Interns required: (Please specify their tasks if possible)

Chen Haoyang, Huang Runxiang, Yang Wenkai

Methods and Standards:

Procedures	Objective	Key Activities
Requirement Gathering and Analysis	The team should meet with ISS to scope the details of project and ensure the achievement of business objectives.	1. Gather & Analyze Requirements 2. Define internal and External Design 3. Prioritize & Consolidate Requirements 4. Establish Functional Baseline
Technical Construction	· To develop the source code in accordance to the design. · To perform unit testing to ensure the quality before the components are integrated as a whole project	1. Setup Development Environment 2. Understand the System Context, Design 3. Perform Coding 4. Conduct Unit Testing
Integration Testing and acceptance testing	To ensure interface compatibility and confirm that the integrated system hardware and system software meets requirements and is ready for acceptance testing.	1. Prepare System Test Specifications 2. Prepare for Test Execution 3. Conduct System Integration Testing 4. Evaluate Testing 5. Establish Product Baseline
Acceptance Testing	To obtain ISS user acceptance that the system meets the requirements.	1. Plan for Acceptance Testing 2. Conduct Training for Acceptance Testing 3. Prepare for Acceptance Test Execution 4. ISS Evaluate Testing
Delivery	To deploy the system into production (ISS standalone server) environment.	1. Software must be packed by following ISS's standard 2. Deployment guideline must be provided in ISS production (ISS standalone server) format 3. Production (ISS standalone server) support and troubleshooting process must be defined.

Team Formation & Registration

Team Name: Group 1
Project Title (repeated): Book Ranking and Category recognition system
System Name (if decided): N.A.
Team Member 1 Name: Chen Haoyang
Team Member 1 Matriculation Number: A0261905E
Team Member 1 Contact (Mobile/Email): Hapyangc0308@gmail.com
Team Member 2 Name: Yang Wenkai
Team Member 2 Matriculation Number: A0261636A
Team Member 2 Contact (Mobile/Email): E0983030@u.nus.edu
Team Member 3 Name: Huang Runxiang
Team Member 3 Matriculation Number: A0261627A
Team Member 3 Contact (Mobile/Email): Runxiang.huang@u.nus.edu

9.2 Appendix B: Personal Report

9.2.1 Chen Haoyang Personal Report

Personal Contribution

Group Communication:

- Synchronize project progress and organize discussions.
- External resource management and in-group coordination.

Backend Algorithm and Application:

- Choose a suitable dataset and perform preprocessing
- Try out many different machine learning models to find the algorithm that best matches our needs.
- Save the trained model and write an application to call the model to achieve our expected functions and present the model training results to the user.

What I learned from this Project

I have learned a lot from this project experience, both from the project itself, from the team members I work with, and even from the papers we read while looking for materials. Some of the important points are as follows. First of all, my understanding of machine learning has deepened, which can be reflected in the detailed section on the model in the report. I also have a new understanding of the flexible selection of models and the different processing methods of data.

Second, I learned a lot about front-end and back-end connections. Before this, I had no exposure to related fields at all, so it is very pleasant for me to have exposure and understanding of front-end and back-end development through this project experience.

Finally, I think the spirit of my team members is very worth learning. They are very active in the face of difficulties in the project and never back down. They often have a positive attitude toward things they don't understand, and are willing to try and learn over and over again. In addition, I think I have benefited a lot from everyone's time planning and project schedule management.

Future Usage

I think this project experience will provide a lot of help for my future study and life. First of all, a good project cooperation experience can make me more willing to take the first step to break the ice in future cooperation. Secondly, the knowledge learned in this project has a wide range and is applied in today's hot computer vision

and NLP. Finally, in the future, I can better practice the application of machine learning knowledge in real-world problems.

9.2.2 Yang Wenaki Personal Report

Personal Contribution

Group Communication:

- Raise the project inspiration and write the proposal.
- Preliminary survey data integration project framework and organize the team discussion.

UI/UX Development:

- Learn the Vue3 framework and wirte the html code.
- Set up the basic component about the user input, the display of result and user login system.

Backend Algorithm and Application:

- Learn the basic NLP algorithms about the text data processed.
- Help our teammates to choose the algorithm selection and test the trained models accuracy.
- Utilize the flask framework to write the communication between users input and trained models.

What I learned from this Project

After this project experience, I have learned some classic algorithms about text information preprocess such as normalizing, tokenizing, lemmatizing etc. At the same time, my team work ability has improved a lot. From our many times disscussion, my some ideas has been outdated and my teammates has taught me how to set up a entire project scope. Maybe in some area I could do better however, in deployment of machine learning models and backend framework I have learned some useful skills in this project.

To start with, basic programming abilities I have learned a lot. Python programming language is the most frequent tool in Deep learning or Machine learning. In this period, I could use python to write the training and testing parts quickly. I became more familiar with the data structure such as tensor and dataframe. In the past time, I always meet some bug on data process or some error on data type. After intensive writing code, I could fix them up in some minutes.

Another point I intend to mention is that the machine learning algorithms mathematical theory could be more clear in my mind. For instance, when we do the book titles classification, we try many classic machine learning

methods. However, the simple logic regression be the best performance in all models. After knowing more about the details of these formulas, I understand the reason why simple method could do better gradually.
In summary, the artificial intelligence methods are all based on big data training and statistic rules. Hence, the deep understand of basic formulas is significant to our machine learning players. In the other hand, we could ignore the other area algorithms which are also useful to solve the problems. Sometimes we could use dispatch and dynamic programming to optimize our algorithms.

Future Usage

Maybe in several years, I will think this is so simple to work out. But actually I have learned the basic skills. Git workflow really help our team and the Google collab also assist me to deploy the models rather than to set up the environment in the local laptop which always consume us much time. The backend and frontend skills are also mastered in this project. Because in the past the web development is never used by me. At present we have construct our own website which give me a sense of achievement.

9.2.3 Huang Runxiang Personal Report

Personal Contribution

Group Communication:

- Confirm completion of project objectives and summarise the proposal.
- Report writing and meetings arrangement.
- Manage the project as an agile process.

UI/UX Development:

- Complete the front-end UI through HTML and Javascript files.
- Adjust and design the layout of the UI.
- Establish the connection between the front-end UI and the back-end algorithm.

Backend Algorithm and Application:

- Research on the performance of different models and algorithms on the specific training dataset.
- Compare the performance of the models and choose the suitable one for this project.
- Test the result accuracy with some knowing knowledge.

What I learned from this Project

During this project, I have learned a lot not only the knowledge of machine learning and deep learning techniques but also improved my skills of critical thinking and time plan management which I will further explain as follow.

Above all, I practiced some learned knowledge such as the data preprocessing method, the NLP word embedding techniques, and the Neural Network algorithms in a real business-aimed project. It provides me a chance to have a deeper understanding of those models and algorithms I have learned in the pass and analysis their performance on the dataset used in this project. Besides, I also learned some new algorithms for image processing and OCR algorithms no matter it is used in the project or not.

Secondly, I learn the front-end and back-end connection parts from scratch. I learned the React and VUE frameworks for front-end development, although they did not be used in this project, it is a great experience I had. From the back-end part, I studied the Django and Flask structures for the connection between the front-end and back-end before chosen Flask for this project.

Furthermore, I practiced the agile project management process in this project which is really helpful for our project delivery and gave me a clearer view of how a real commercial programming team will develop their products. My teammates are also very helpful in following the management process and providing lots of suggestions during the progress of the project. By doing agile management, the project process became simpler and clearer for the team to check the progress and notice the shortcoming part.

Last but not least, my teammates are so good at time planning and always deliver their work before the demand, which inspired me to push myself for better delivery. After this project, I realized that a good team atmosphere really helps product delivery and is able to provide us with more motivation for better work.

Future Usage

This project is a great experience for my future life, not only my study career but my personal life. It offered me a wonderful chance to have a real practice in establishing a complete system which is new to me. Although this project is not perfect, I believe it can become better as we follow our future plan. Furthermore, apart from the programming knowledge, I also improved my soft skills in cooperation and the ability of working as a team.