C964: Computer Science Capstone

Zane Knightwood

[redacted]

12/21/2023

ZK Book Recommender

Table of Contents

Part A: Letter of Transmittal 3
Part B: Project Proposal Plan 5
Project Summary 5
Data Summary 5
Implementation 6
Timeline 7
Evaluation Plan 8
Resources and Costs 8
Part C: Application 9
Part D: Post-implementation Report 10
Solution Summary 10
Data Summary 10
Machine Learning 12
Validation 13
Visualizations 14
User Guide 17

Part A: Letter of Transmittal

December 21, 2023

Jane Doe
Chief Technology Officer
Books R Fun
324 Book Ln
Detroit, MI 48127

Subject: Proposal for Book Recommendation Software

Dear Jane,

As discussed in our recent annual meeting, sales have been steadily decreasing over the last 2 quarters, resulting in a net decrease of 8% for the year. As head of software development, you tasked me with developing a technology-based solution that will address the core issue of this development and aid in the increase of sales in the coming year. After careful deliberation, I believe the problem centers around customer retention and return. Book readers tend to be a voracious lot, seeking new reading material at a regular pace. Our customers, however, are turning to our competitors more often than us. The problem stems from our current method for book recommendation. At this time, a customer will only see recommendations based on the author of the book they've purchased. When that author has only produced one or two books, likewise the recommendations are limited. We need a better system for recommending books to our customers. A system they can trust to suggest books they will love. To that end, I propose we turn to machine learning.

In the past, when implementing a new feature, we have first produced a test case to ensure viability and customer interest. This project should be treated no differently. That is why I am proposing that we develop a new web application for book recommendations. Through publicly available datasets processed via machine learning algorithms, this application will be able to take user input in the form of a book title and return meaningful recommendations to our customers. These recommendations will not be limited to author-only matching. Instead, we will be able to recommend books based on several data points that better encompass the full breadth of a book's content. This will result in customers being able to easily find their next favorite book, which in turn will increase sales as customers purchase those books and return for the next recommendations.

With your authorization, I can begin work on this project immediately. I estimate a production timeline of 6 to 8 weeks and a development cost of $18,459. Due to my experience with machine learning models and my several years as head of software development for our company, I am confident in my analysis of our core sales issue and the solution I am presenting.

Please reach out if you have any further questions. I look forward to getting started on the project and I thank you for your time and consideration.

Sincerely,

Zane Knightwood

Head of Software Development, Books R Fun

Part B: Project Proposal Plan

Project Summary

Books R Fun has seen a decrease in sales over the last two quarters, resulting in a net decrease of 8% for the year. This decrease can be linked to the lack of customer return and retention. Our customers are turning to our competitors to fulfill their needs. Our current method of book recommendation is limited to author-only matching. This limits our recommendations and makes them less helpful to our customers. This project will eliminate this issue by creating a new way to generate recommendations.

The recommendation algorithm created by this project will utilize machine learning algorithms to sort through a dataset of books. This algorithm will pull book recommendations based on a wider range of data points, thus curating a list that is meaningful and helpful for our customers. By presenting them with options they will love, customer return and retention will increase. This increase will directly improve our sales numbers, resulting in higher profits and more stability for Books R Fun.

As this project is a test case, the deliverables will include a collection of preprocessed data for the algorithm to utilize, a functional web application with a user interface, and a recommendation algorithm that can be reformatted and used in our online store at a later date. The project will be considered a success if, after a six-month period of maintenance and monitoring, customer feedback indicates a 70% or greater approval rating. Customer feedback will be collected via email surveys sent out at three and six months post deployment.

Data Summary

The raw data for this project will be collected from this link via kaggle.com:
https://www.kaggle.com/datasets/mdhamani/goodreads-books-100k

The dataset contains several useful fields including author, book format, description, genre, a link to a cover image, the ISBN, the ISBN13, a link to the review page on Goodreads, number of pages, Goodreads rating, title, and total ratings. This dataset has been collected from Goodreads.com and is publicly available.

This dataset is ideal as it contains a large number of books (100k upon download and approximately 68.8k after preprocessing) that can be searched. The large size of the dataset is particularly helpful as a larger set provides for more potential accuracy in the returned recommendations. The dataset is also well-suited due to its contents. Each entry contains an array of genres that fully encompass the contents of the book described. The program will be able to match not only authors, as we have in the past, but genres and ratings as well. Page count can also be used, as it helps identify the length of a book and is a useful identifier for most readers.

The dataset will be preprocessed before being imported into the application. The dataset includes some rows with missing or incomplete entries, these will be dropped. Columns for book format and description, ISBNs, and total ratings will also be dropped as they are not well suited to the proposed analysis. Further, some titles have been found to contain brackets such as "[". These entries will be removed to prevent processing issues within the program.

The dataset will be downloaded as a .csv file that will then be imported by the program. This method of data access will allow the program to be easily updated with new data should the need arise, or new data become available.

Implementation

Due to the well-defined requirements, the unlikeliness of changes to those requirements, and the narrow scope of the project, the waterfall development methodology will be used. The project structure will follow the standard waterfall phases and is described in detail here:

Requirements: All resources and requirements will be assessed and assembled. The necessary dataset will be preprocessed and fully evaluated.

Design: Machine learning methods will be examined for best fit to the needs of the project. The user interface will be designed. The algorithm will be defined.

Implementation: The algorithm and web application will be programmed.

Verification: Testing will be done to ensure the user interface is functional, the web application is functional, and the algorithm produces the required results.

Deployment: The web application will be deployed to the server and made available to customers.

Maintenance: The application will be monitored for performance and any needed bug fixes will be deployed.

After implementation, the project will enter a six-month period of maintenance and monitoring to test its viability and gauge customer interest. Feedback will be collected from customers via email surveys sent out at three- and six months post-deployment.

Timeline

The project can begin as soon as approval is acquired. The following timeline assumes approval within 3 days from submission of this proposal.

Milestone or deliverable	Duration (hours or days)	Projected start date	Anticipated end date
Project Approval	3 days	December 21, 2023	December 24, 2023
Requirements Gathering	2 days	December 27, 2023	December 28, 2023
Analize Machine Learning Methods, Design User Interface, Define Algorithm	2 weeks	January 2, 2024	January 12, 2024
Programming of Algorithm and Web Application	3 weeks	January 15, 2024	February 2, 2024
Testing and QA of Algorithm and Web Application, Web Application Deployment	1 week	February 5, 2024	February 9, 2024
Maintenance and Monitoring, First Email Survey Sent	3 months	February 9, 2024	April 9, 2024
Maintenance and Monitoring, Second Email Survey Sent	3 months	April 9, 2024	July 9, 2024

Evaluation Plan

During development, testing will be performed to ensure the product is meeting the necessary benchmarks.

Unit tests will be written to test for bugs within the user interface, the overall web application, and the algorithm itself. These tests will ensure the application meets specifications and requirements.

The following evaluation criteria will be monitored throughout the development process:

User-Friendliness – The user interface should be easy to understand, using common methods for user interaction.

Application Function – The user should be able to input a book title and receive six books that are similar to the input.

Helpful Recommendations – Output books should be similar in content to the input book. This will be evaluated by an accuracy rating of $accuracy = \frac{elements\ matched}{elements\ to\ be\ matched}$ where elements relate to the author(s), genre(s), page count, and rating of the books.

At three and six months after deployment, a survey will be sent out to customers to evaluate their satisfaction with the application. A customer approval rating of 70% or greater will indicate the success of the application.

Resources and Costs

The company computer already in use will satisfy the hardware needs of this project, and so has no additional cost. Similarly, the dataset is freely available and will incur no cost. Beyond that, the following expenses are expected for the project.

Resource	Description	Cost
Software	Pycharm, VSCode	$1,065
Labor	Expected Total Hours: 260 Expected Hourly Rate: $60	$15, 600
Hosting and Maintenance	Python Anywhere Hosting Package: $99 per month	$594
Survey Design and Marketing	Expected Total Hours: 30 Expected Hourly Rate: $40	$1,200
	Total	$18,459

Part C: Application

The application may be viewed at zaneknightwood.pythonanywhere.com.

Application source code is attached.

Part D: Post-implementation Report

Solution Summary

Books R Fun, an online bookseller, needed a way to recommend books to their customers based on books they had already purchased. The recommendations needed to be similar in content to the customer's already liked books as a way to entice them to make further purchases.

This program was developed as a test case for the needed recommendation algorithm. The program utilizes a machine learning algorithm to generate a set of six books similar in content to a user's input book. The test case proved that the algorithm functions as needed. The algorithm can now be used to provide the same service to customers. This should result in increased sales once it can be replicated for the Books R Fun online store.

Data Summary

The data analyzed by the recommendation algorithm was obtained from https://www.kaggle.com/datasets/mdhamani/goodreads-books-100k

The original dataset, once downloaded, was 114MB and therefore preprocessing was performed to make it more manageable before being used by the algorithm. The following code was applied to the dataset. This code passed the dataset into a data frame using Python's Pandas library. Once in the data frame, it was searched for missing entries and any rows with missing entries were deleted. Next, unnecessary columns were removed. Then the title column was searched for the character "[" and any rows where it was found were removed. These brackets caused issues within the program so removing these entries was the solution. Finally, the data frame was saved for future use. The output .csv file was decreased from the initial 114MB to 22.2 MB. This improved runtime and allowed the .csv file to be uploaded to the server.

```
 7
 8  import pandas as pd
 9
10  filePath = "GoodReads_100k_books.csv"
11
12  df = pd.read_csv(filePath)
13
14  df = df.dropna()
15  df = df.drop(columns=['bookformat', 'desc', 'isbn', 'isbn13', 'reviews', 'totalratings'])
16  df = df.drop(df[df['title'].str.contains('\[')].index)
17  df.to_csv('GoodReads_100k_books.csv')
18
```

Figure 1. Preprocessing code.

After preprocessing, the dataset was able to be utilized by the program. The following module was created to handle the initial processing of the data by the program. This module contained the function “getDF” which read the dataset’s .csv file into a data frame and ensured no empty cells remained in it.

```python
# -*- coding: utf-8 -*-
"""
Created on Fri Dec  1 16:04:49 2023

@author: Zane Knightwood
"""

# Creates a dataframe from the dataset and edits it for use by the program.

import pandas as pd
from c964CapstoneZK.settings import BASE_DIR

def getDF():
    file_path = BASE_DIR / 'staticfiles' / 'GoodReads_100k_books.csv'
    df = pd.read_csv(file_path, sep=',')
    df = df.dropna()
    return df

```

Figure 2. Dataset processing module.

Machine Learning

The K-Nearest Neighbor (KNN) was used as a basis for the development of the program. This algorithm is a supervised learning method that has proven to be useful for pattern recognition. KNN is typically used for classification when standard Euclidean distance is the metric being measured, but it can be applied to any measurable metric. The concept behind KNN is that it takes in a specific point, and then searches for the next point based on the input metric such that the next point is closest to the initial point. The metrics used in the ZK Book Recommender were not numerical, however, being a mix of strings and integers. Due to this, the algorithm had to be adjusted to fit the data being used.

To measure the next closest book, the data was filtered through a series of search parameters. First, the list of authors was pulled from the user book. This list was then applied to a search function that went through the database and set the ID of each book whose author matched with an author from the user book into a list. Next, the same was done for the user book's list of genres, and those matches were also added to the list. Then the Goodreads rating of the user book was extracted and compared to the database. Finally, the page count of the user book went through the same process.

At the end of this search process, when all the user book elements had been extracted and matched to books in the database, a list was created containing the indexes of all the books that have been matched to the user book. The list contained multiple instances of these indexes, one for each match case that was found. The instances were then tallied up and the tallies were paired with the proper book index.

Now that the data had a "distance" associated with each book, the KNN could be more readily applied. It was at this point that the data was sorted from the highest number of element matches to the lowest. Once that had been completed, the algorithm was able to return the six closest books.

Utilizing this method allowed the application to provide a list of books that are similar in nature to the user input. Due to the varied nature of the data, it would be difficult to apply a typical clustering algorithm. By using this method of element matching and then applying the concepts used in the KNN method, the application was able to do as intended.

Validation

The accuracy of the method was determined by a match percentage, which was calculated for each book the algorithm supplied. This percentage can be seen on the book view page of the application.

Figure 3. Match percent shown on book view page.

The match percentage was calculated by comparing the number of elements in the input book to be matched with the number of matching elements in the recommended book. This ratio was then converted to a percentage so it could be easily understood.

In the future, a user rating system could be used to further test for accuracy. This system could allow users to rate the accuracy based on metrics such as whether they enjoyed the book that was recommended and if the recommended book was similar in nature to the input book.

Visualizations

Data visualizations are located on the visualizations page of the application. There are three visualizations provided on this page.

Visualization 1

The first is an explanation of the machine learning method used. This is in the form of a model explanation flow chart. The chart shows how the data is categorized, counted, and analyzed by the program to generate book recommendations.

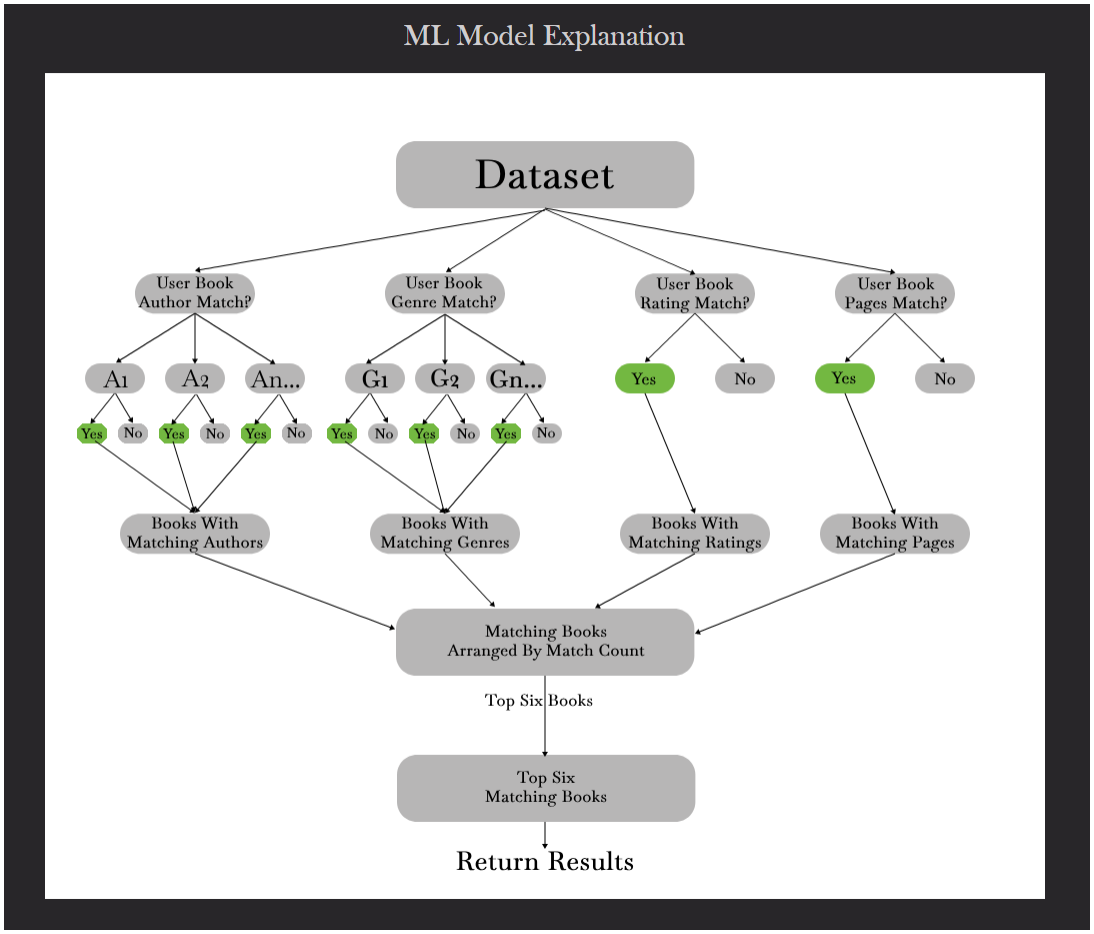

Figure 4. The ML model explanation.

Visualization 2

The second visualization is a pie chart showing the accuracy in the similarity of the generated books to the input book. Each similarity percentage is shown in comparison to the other recommended books.

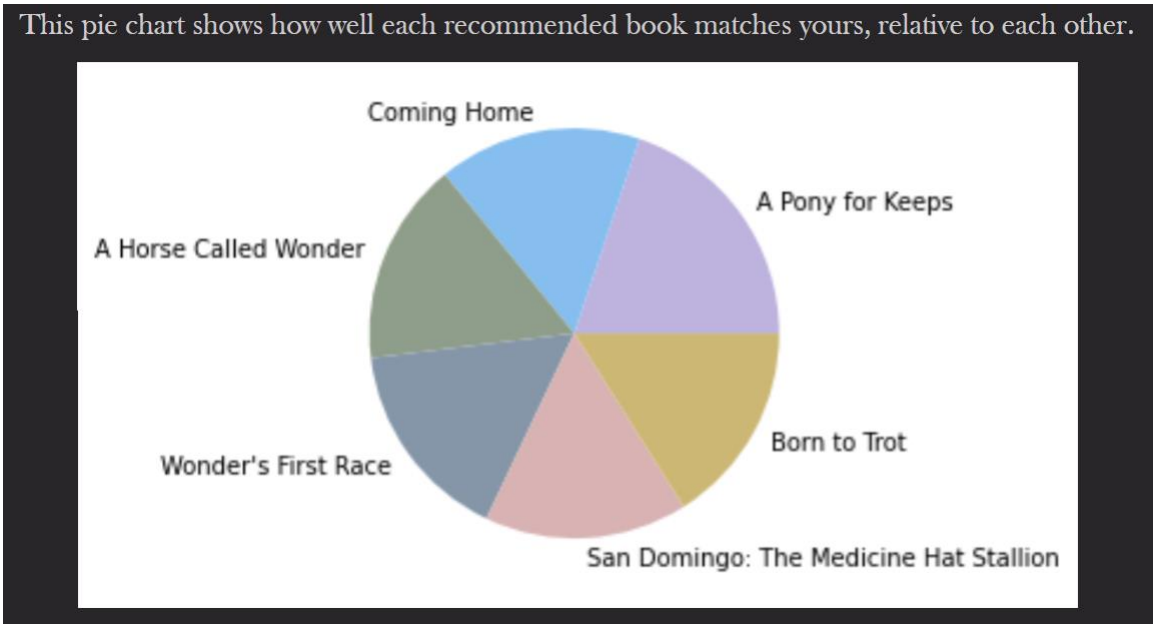

Figure 5. The pie chart.

Visualization 3

The third visualization is a stacked bar graph. This graph shows the distribution of the matching elements found for each recommended book. The overall matching tallies from this information are what was used in the final determination of which books should be recommended.

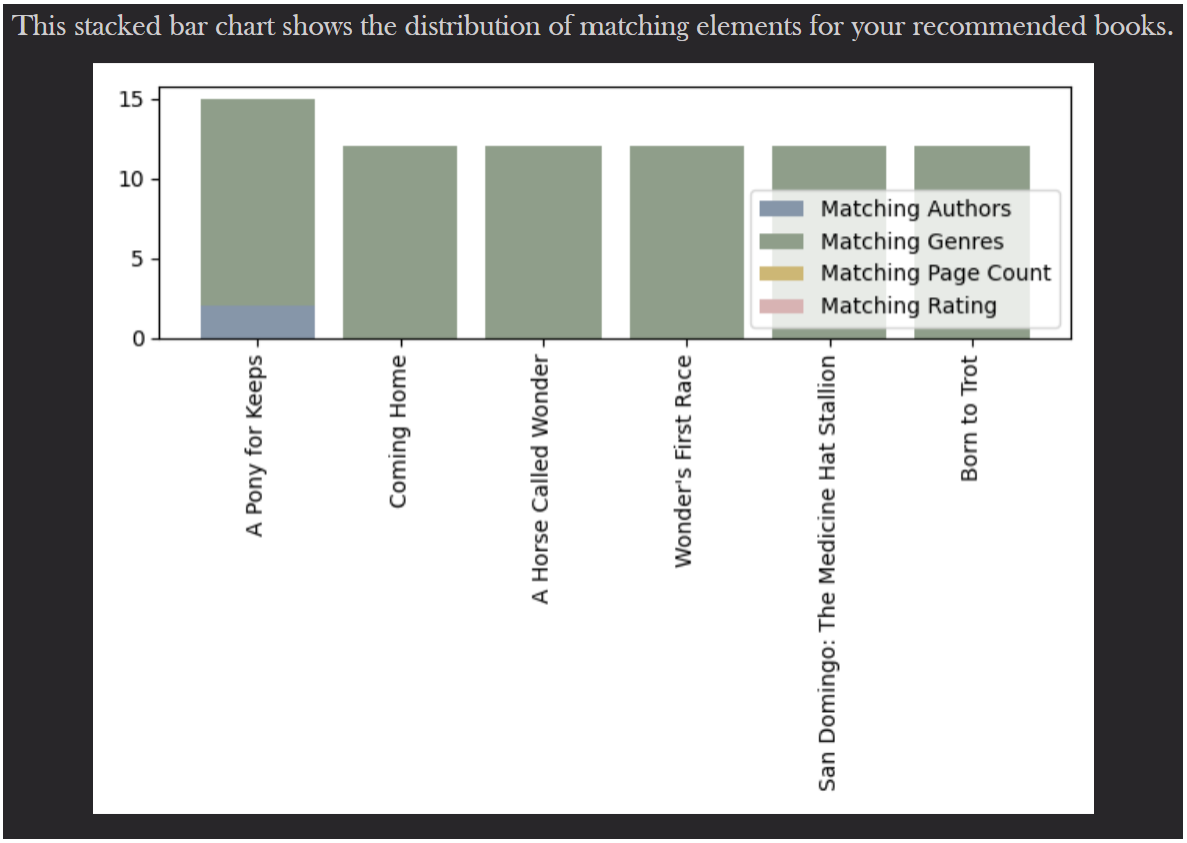

Figure 6. The stacked bar graph.

User Guide

Section 1, Getting Started:

The ZK Book Recommender is a server-hosted web application and therefore needs no installation or special software to access. Only an internet connection and web browser are necessary.

The application has several paths that can be taken through it, depending on user choices and input. This guide will take you step by step through the application, from initial access through each page and path.

Step 1: On a computer or mobile device, navigate to your preferred browser and go to this URL: zaneknightwood.pythonanywhere.com. Once the page has loaded, you will see the name of the application across the top of your screen, with a text box below it.

Step 2: In the text box, type the title of a book. This is not case-sensitive and partial titles are fine. In this example, we will search for "I Want A Pony". Once you have typed your title, click "Go!" to begin.

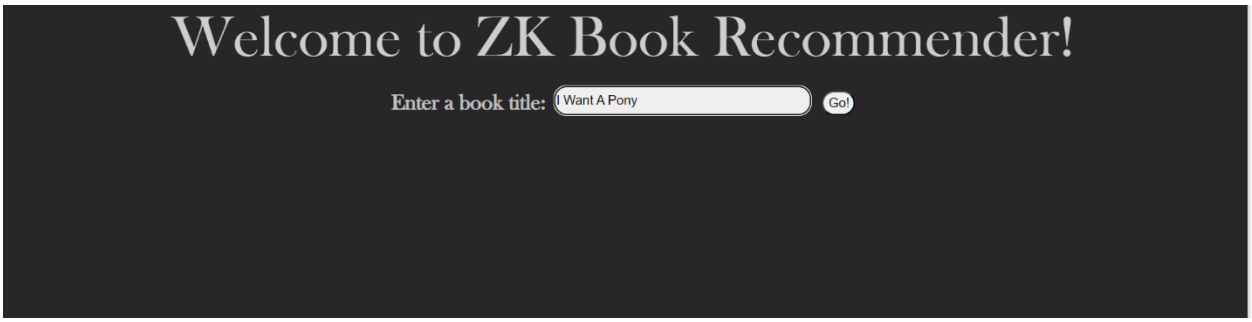

Figure 7. The homepage of ZK Book Recommender.

From the "Go!" button, one of three things will happen. The next portion of the guide will be broken into sections based on each progression path.

Section 2, Progression Paths:

Path 1: A book is found with the input title.

If your book is found in the database, you will be taken to the recommendations page.

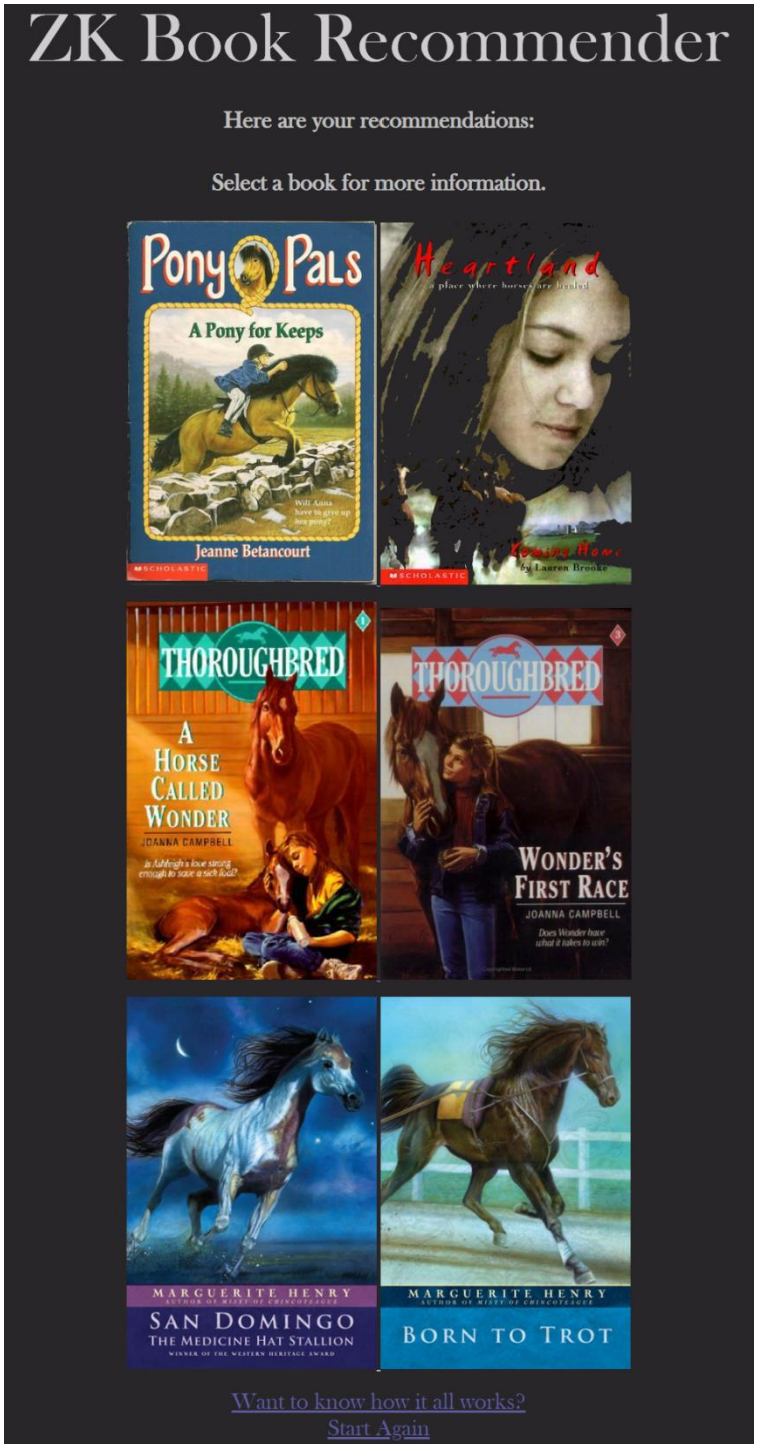

Figure 8. The recommendations page.

Path 1, Step 1: The recommendations page contains the book covers for the six generated recommendations. These book covers are all clickable and lead to a page with more information about the recommended book. This information includes the title, author(s), genre(s), and Goodreads rating. It also includes the match percentage to your book. Click on any of the six cover images to go to that page.

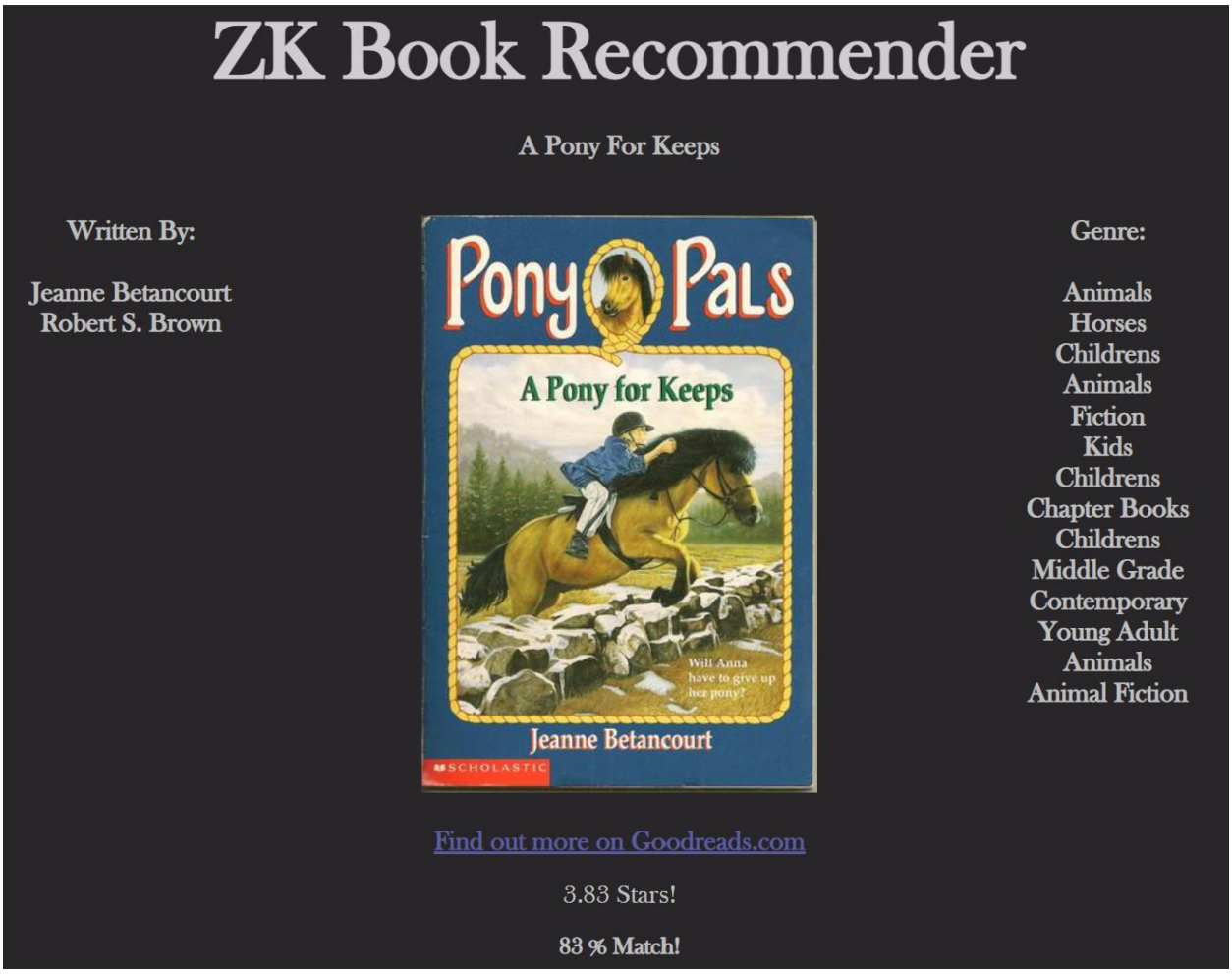

Figure 9. The book view page.

Path 1, Step 2: From the book view page, you have two options.

Option one: Under the cover image is a link. Click this link to go to the Goodreads page for the recommended book. There you can read the synopsis and reviews, leave a review yourself, or follow the Amazon link to purchase the book.

Option two: Click the back button on your browser to return to the recommendations page.

Path 1, Step 3: Once back on the recommendations page, click the link “Want to know how it all works?” to be taken to the visualizations page. On this page, you will find an explanation and diagram of the machine learning model used to generate your book recommendations. You will also find a pie chart showing the relationship the recommended books have towards each other in regard to how well they match your book. The last image on this page is a stacked bar chart showing the distribution of matching elements for each book that the algorithm used to make its recommendations.

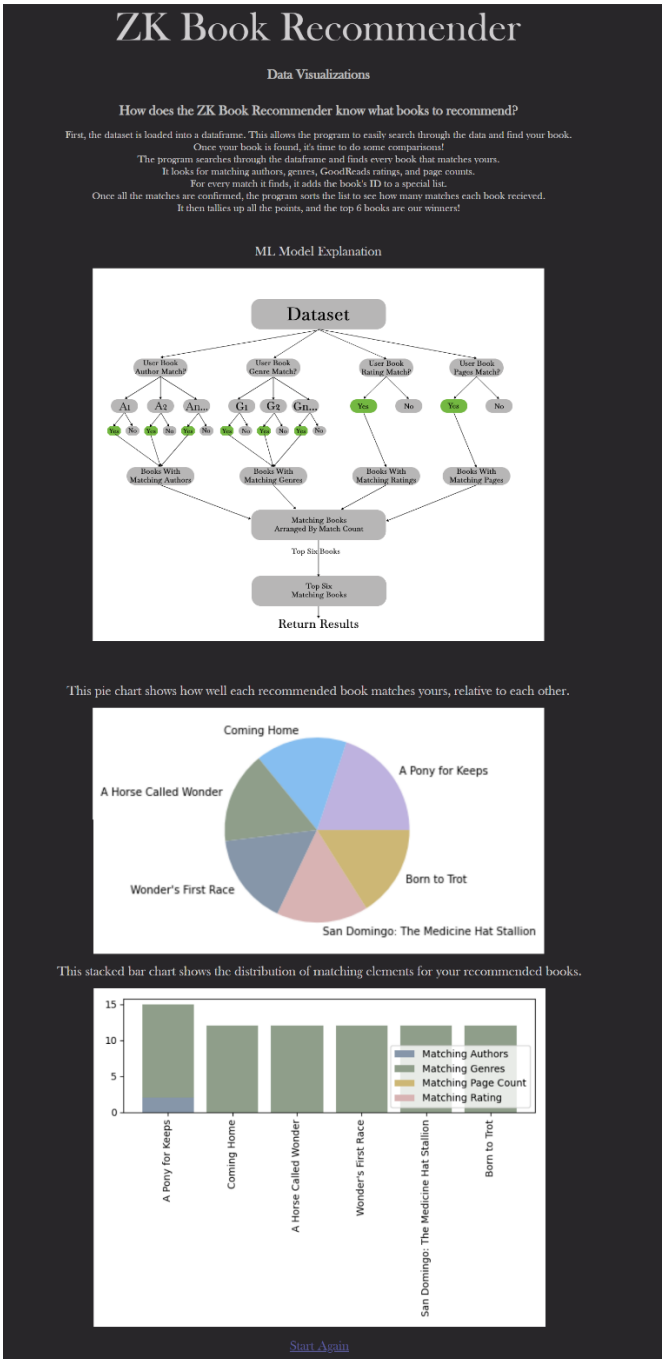

Figure 10. The visualizations page.

Path 1, Step 4: From this page, or the recommendations page (which can be accessed again by clicking the back button on your browser), you can click the “Start Again” link to go back to the homepage and input a different book.

Path 2: No book was found.

If your book was not found in the database, the message “Sorry, no book was found matching that title. Please try again.” Will appear above the text box.

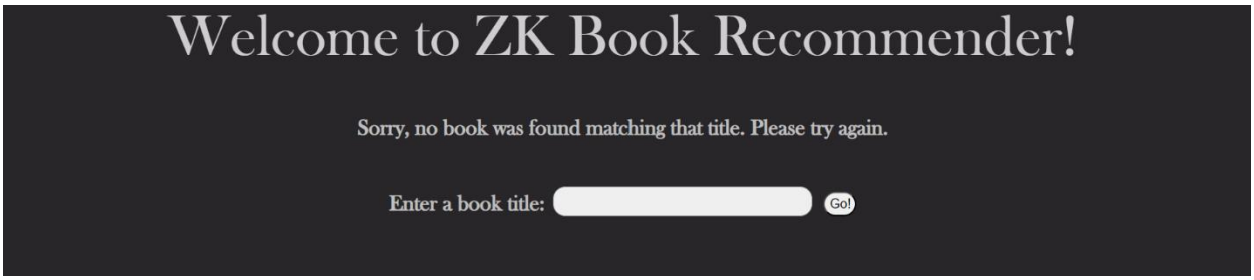

Figure 11. The "No Book" text appears above the text box.

Path 2, Step 1: In the text box, type the title of a book. Once again, this is not case-sensitive, and partial titles are fine.

Path 2, Step 2: Once you have typed your title, click “Go!” to begin.

Return to the top of Section 2, Progression Paths.

Path 3: More than one book was found.

If you entered a partial title, or if your title matches more than one book in the database, you will be taken to the multiple book selection page.

Figure 12. The multiple book selection page.

Path 3, Step 1: The multiple book selection page contains the number of books found and a dropdown box for you to choose your intended book. Select a book from the dropdown box and click the “Go!” button.

Path 3, Step 2: Once you have clicked the “Go!” button, you will be taken to the recommendations page. From here, the path converges with Path 1. Go to Path 1, Step 1 to continue.