



# Team 3: Book Recommendation System

Shaina Chauhan  
Stephanie La Belle  
Mandara Kadya  
Mike Korzeniewski  
Allister Rebello

# Table of contents

01.

Introduction

03.

Recommendation  
Model

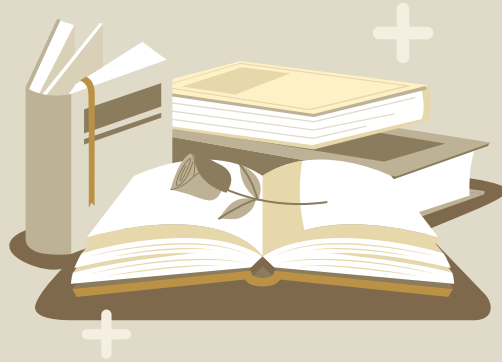
05.  
References

02.

Insight &  
Visualizations

04.

Conclusions





01

# Introduction

An overview of our  
presentation and dataset

# Introduction

- With the emergence of platforms such as YouTube, Amazon, and Netflix, recommender systems have become increasingly integrated into our daily lives.
- These systems play a key role in other various areas, such as e-commerce, online advertising, and news platforms
- Recommender systems are algorithms designed to propose relevant items to users, and in our case, book recommendations
- In this presentation, we will offer key insights about our dataset, and demonstrate the recommendation system we created



# Our Dataset



## Books.csv

- Books are recognized using their unique ISBNs
- Additionally, the dataset includes certain content-based details such as the book title, author, publication year, publisher, and links to cover images



## Ratings.csv

- Ratings are expressed on scale from 1 to 10 (Integers)
- They are paired with a unique user ID, as well as the books ISBN



## Users.csv

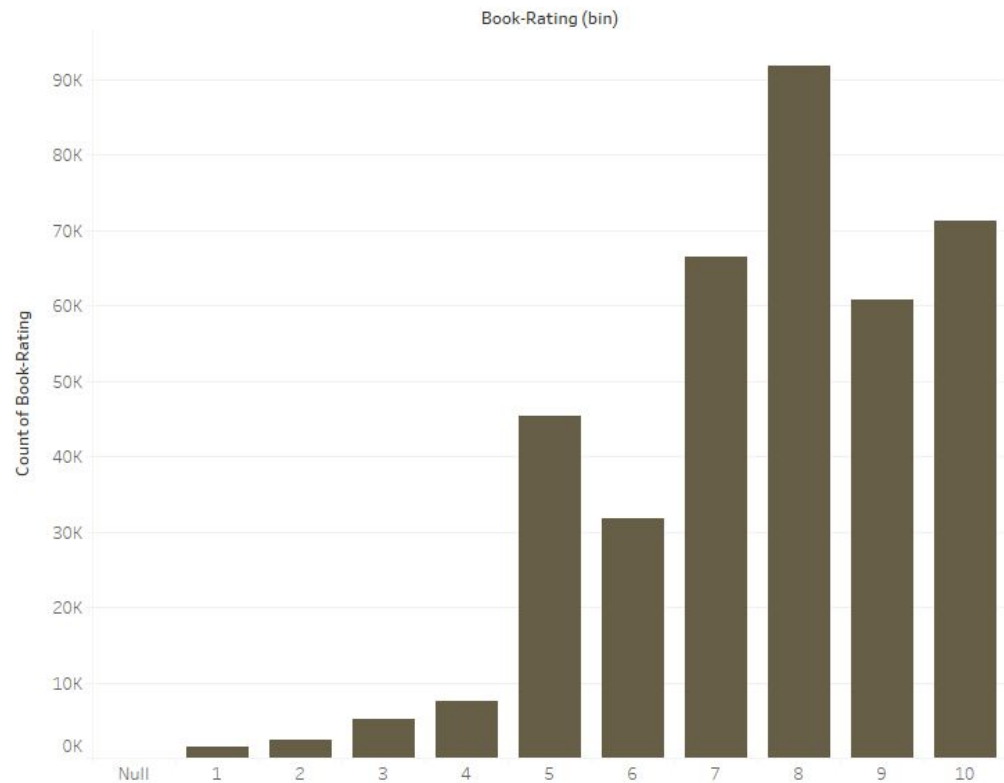
- This file contains the user information: user ID, the location of the user, and the user's age

- After cleaning the data, we left merged the Books and Ratings csv's

# Insights & Visualizations

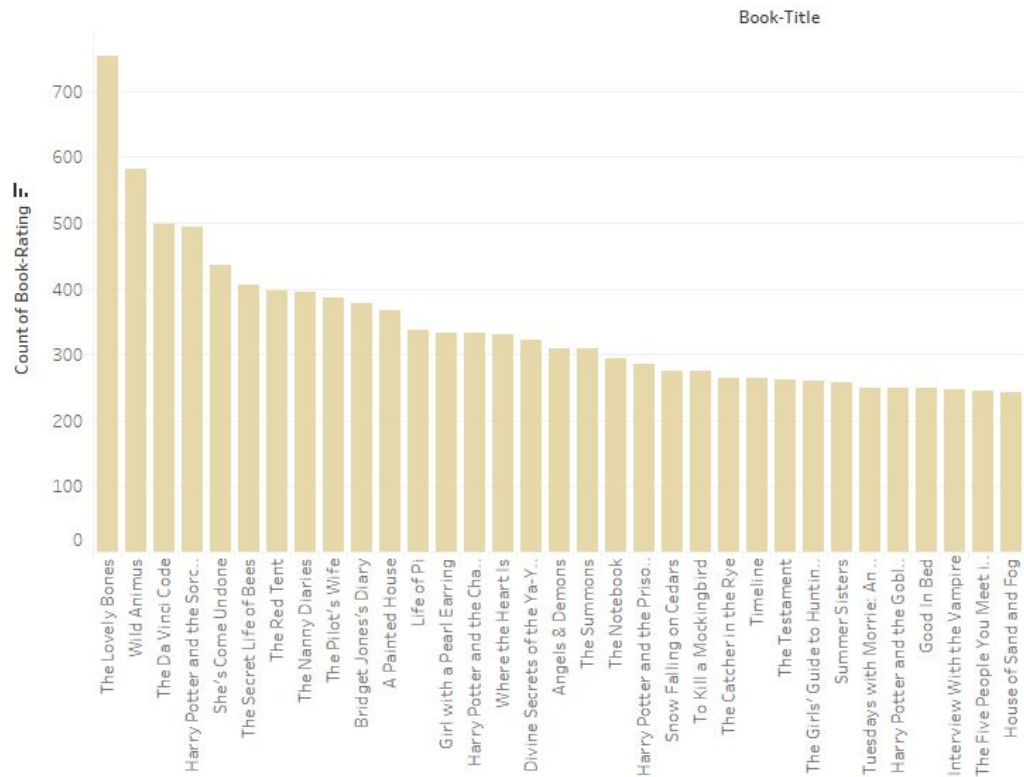


## Rating Distribution



- Our dataset contains ~380,000 ratings
- The average book rating in our dataset is ~7.6
- The most common rating is 8

## Rating Count (Books)



## Most Rated Books

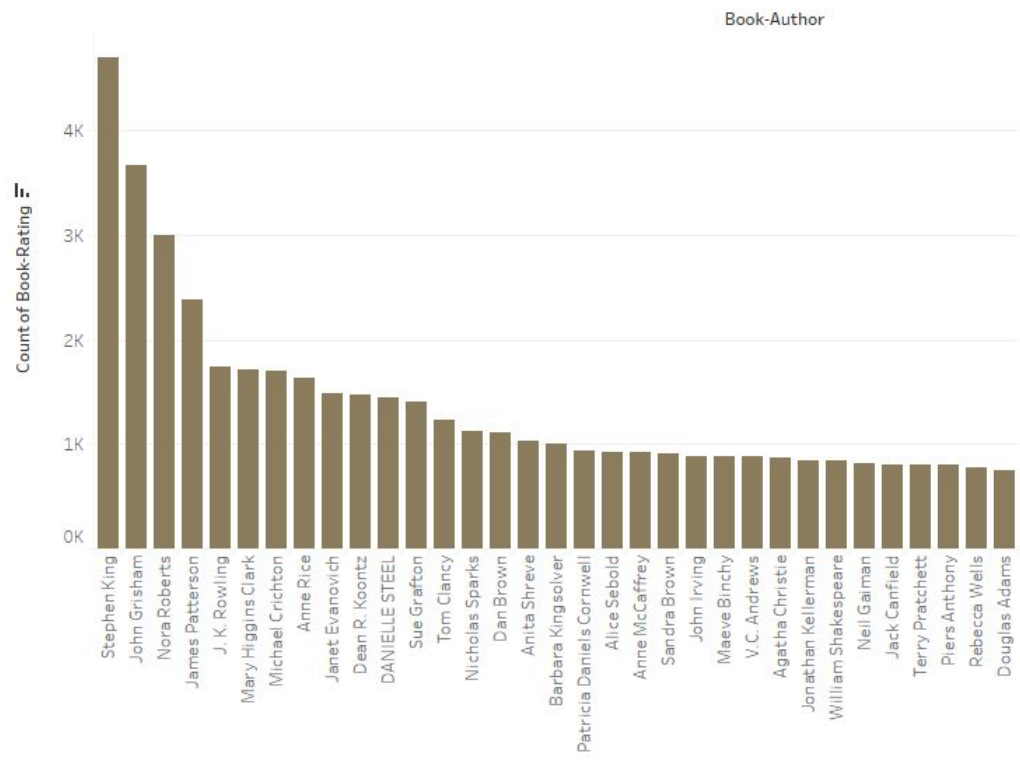
1. The Lovely Bones (754)  
*Alice Sebold*
2. Wild Animus (581)  
*Rich Shapero*
3. The Da Vinci Code (499)  
*Dan Brown*
4. Harry Potter and the Sorcerer's Stone (493)  
*J. K. Rowling*
5. She's Come Undone (436)  
*Wally Lamb*



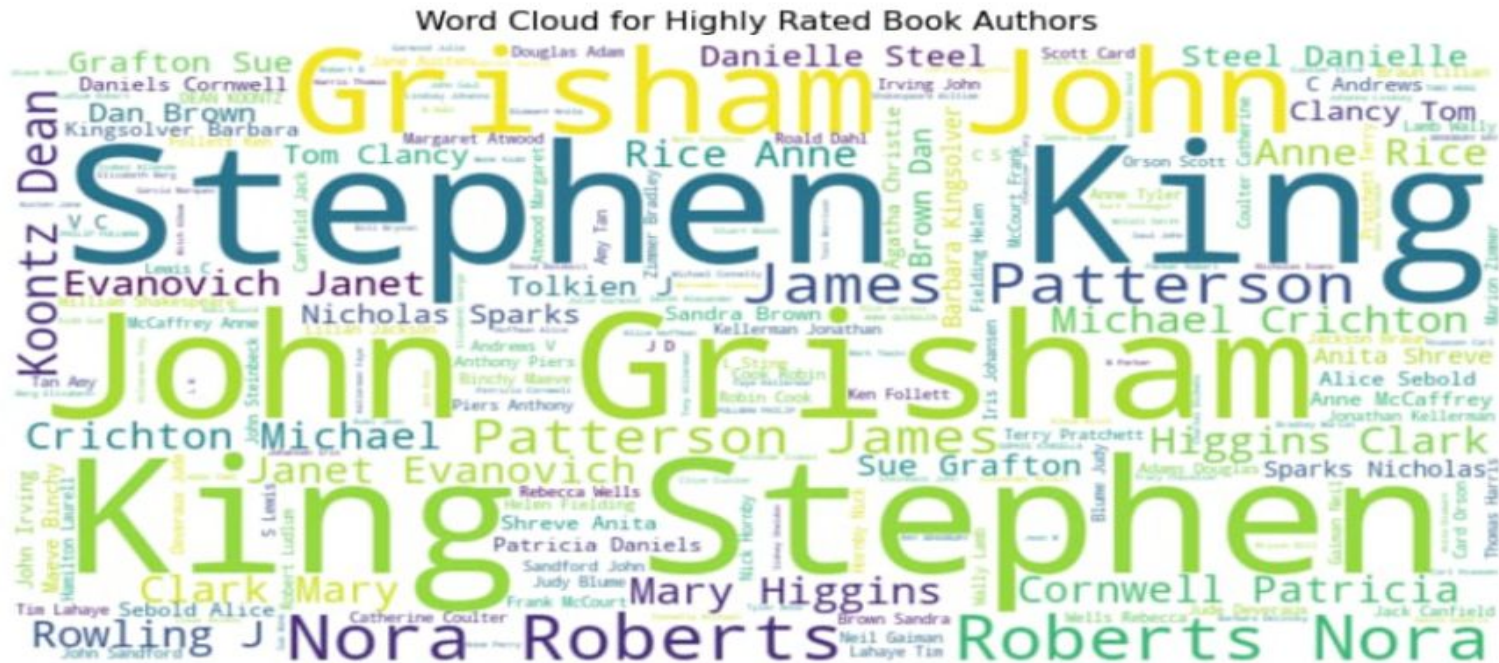
## Most Rated Authors

1. Stephen King (4,703)
2. John Grisham (3,670)
3. Nora Roberts (3,001)
4. James Patterson (2,387)
5. J. K. Rowling (1,746)

Rating Count (Author)



## Word Cloud of Highly Rated Authors



Word cloud titled "Word Cloud for Highly Rated Book Authors," featuring a variety of words in different sizes and orientations. Prominent words like "Stephen," "King," "John," and "Grisham" suggest these are common authors among highly rated books. The size of each word likely indicates the frequency of its occurrence in book authors.

# Top 10 Publishers by Book Wise

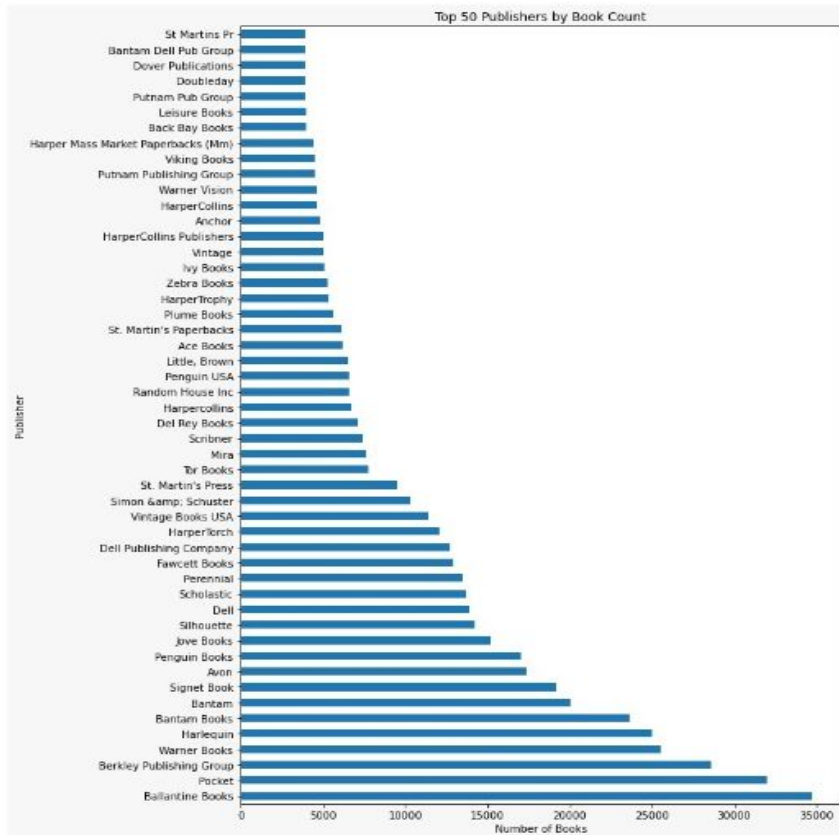


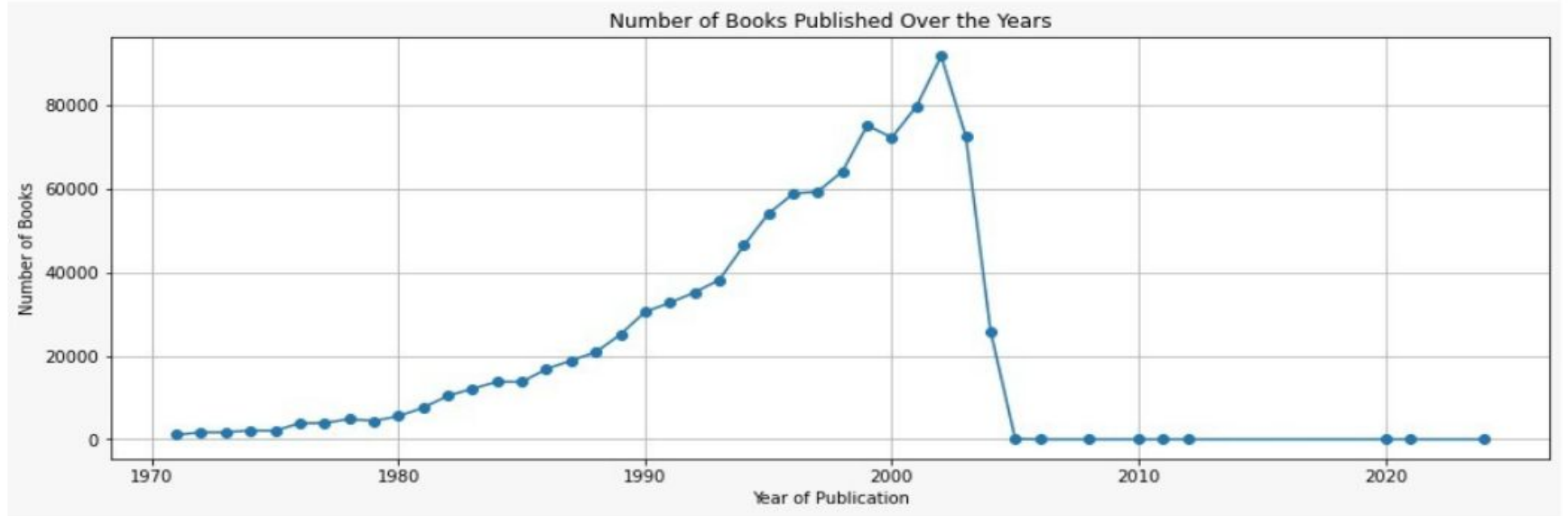
Chart titled "Top 50 Publishers by Book Count," ranking publishers based on the number of books they have published. The bars represent individual publishers, with the length of each bar indicating the total number of books published. Ballantine Books appears to have the highest book count, significantly more than the others, with the rest following in descending order.

# Top 10 Most Active Users Rating Wise



Chart titled "Top 10 Most Active Users by Number of Ratings," showing the distribution of ratings across users identified by their user IDs. The user with ID 11676 is the most active, having given the highest number of ratings, while the rest follow in descending order. The chart reflects the varying levels of activity among the top users within a rating system.

# Number of Books Published Over Years



Graph titled "Number of Books Published Over the Years," showing a trend in book publication from 1970 to around 2020. There is a steady increase in the number of books published until it peaks sharply around the year 2000, after which there is a drastic decline, returning to lower levels similar to the early 1970s.

# Recommendation Models

K-Nearest Neighbors  
&  
TensorFlow







# K-Nearest Neighbors

# KNN Model Data Preparation

- **Data Input Structure:**

- Pivot Table of Book-Titles by User ID with values as Rating
- Converted to Compressed Sparse Matrix with SciPy

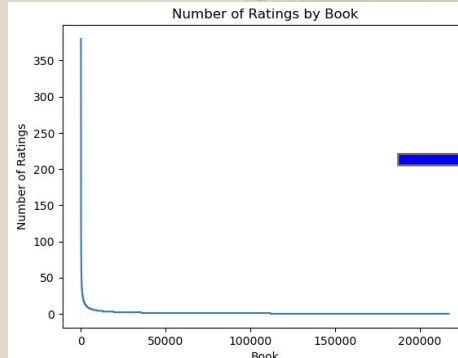
- **Filtered Out:**

- Users with  $< 15$  books purchased
- Books with  $< 50$  purchases

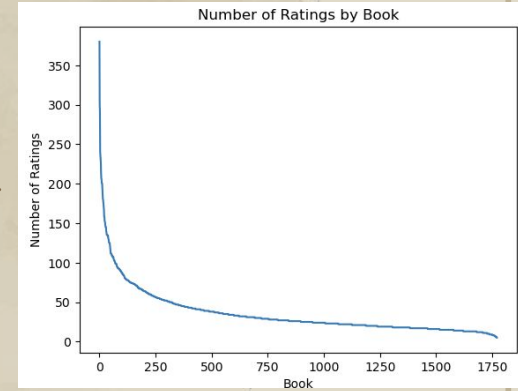
- **Sparsity= 1-(# non-zeros/# of cells):**

- <99.5%

## Before Filtering

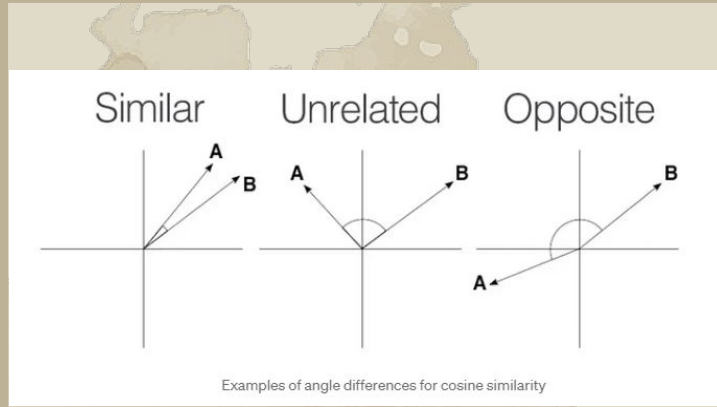


## After Filtering

[illegible]



# KNN Model Design



<https://medium.com/@milana.shxanukova15/cosine-distance-and-cosine-similarity-a5da0e4d9ded>

**K-Nearest Neighbors Algorithm:**  
Utilizes proximity of data points to make classifications or predictions.

**Cosine Metric:** Measure of the angle difference between data points.

**Brute Algorithm:** Calculating cosine metric between input data and all other data points

# Recommender in Action

```
print_recommender("jurassic park", matrix_books_users, model_knn, 20)
```

Book Query: Jurassic Park

Your top 5 recommended books are:

1. The Lost World -- by Michael Crichton -- Cosine Distance 0.79
2. Congo -- by Michael Crichton -- Cosine Distance 0.84
3. Silence of the Lambs -- by Thomas Harris -- Cosine Distance 0.86
4. Rising Sun -- by MICHAEL CRICHTON -- Cosine Distance 0.87
5. Red Dragon -- by Thomas Harris -- Cosine Distance 0.88

Authors you can try:

1. Michael Crichton
2. Thomas Harris
3. MICHAEL CRICHTON
4. John Grisham
5. Stephen King

```
print_recommender("harry potter and the chamber of secrets", matrix_books_users, model_knn, 30)
```

Book Query: Harry Potter and the Chamber of Secrets

Your top 5 recommended books are:

1. Harry Potter and the Prisoner of Azkaban -- by J. K. Rowling -- Cosine Distance 0.4
2. Harry Potter and the Goblet of Fire -- by J. K. Rowling -- Cosine Distance 0.41
3. Harry Potter and the Sorcerer's Stone -- by J. K. Rowling -- Cosine Distance 0.44
4. Harry Potter and the Order of the Phoenix -- by J. K. Rowling -- Cosine Distance 0.6
5. The Fellowship of the Ring (The Lord of the Rings, Part 1) -- by J.R.R. Tolkien -- Cosine Distance 0.81

Authors you can try:

1. J. K. Rowling
2. J.R.R. Tolkien
3. Lemony Snicket
4. E. B. White
5. Harper Lee

```
print_recommender("the secret life of bees", matrix_books_users, model_knn, 15)
```

Book Query: The Secret Life of Bees

Your top 5 recommended books are:

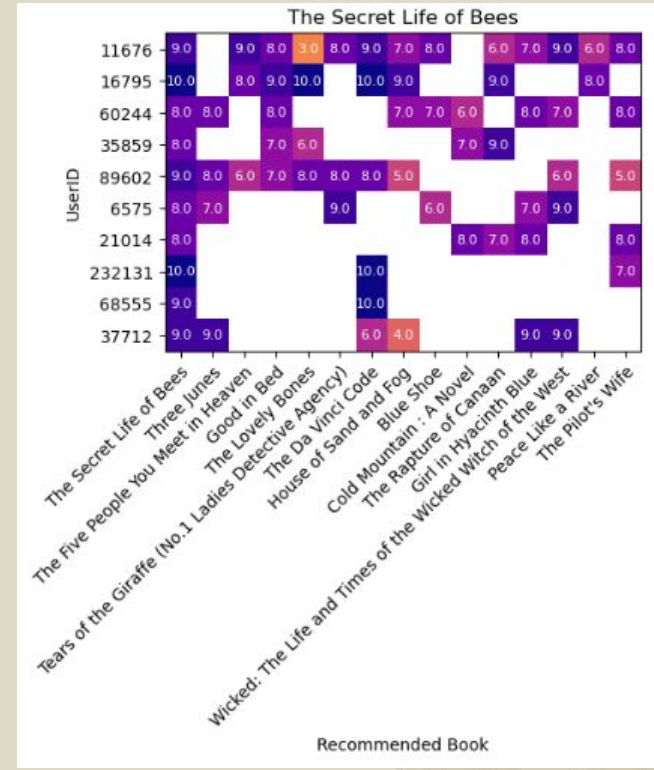
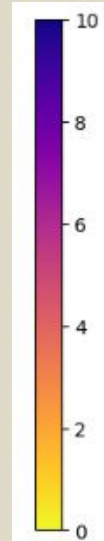
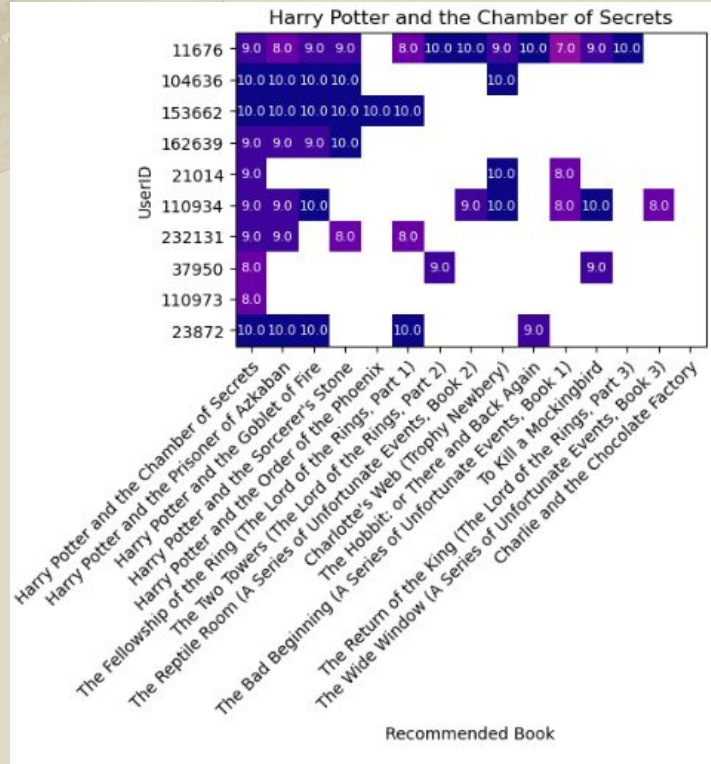
1. Three Junes -- by JULIA GLASS -- Cosine Distance 0.85
2. The Five People You Meet in Heaven -- by Mitch Albom -- Cosine Distance 0.85
3. Good in Bed -- by Jennifer Weiner -- Cosine Distance 0.85
4. The Lovely Bones -- by Alice Sebold -- Cosine Distance 0.86
5. Tears of the Giraffe (No.1 Ladies Detective Agency) -- by Alexander McCall Smith -- Cosine Distance 0.86

Authors you can try:

1. Sue Monk Kidd
2. JULIA GLASS
3. Mitch Albom
4. Jennifer Weiner
5. Alice Sebold



# Recommender Assessment



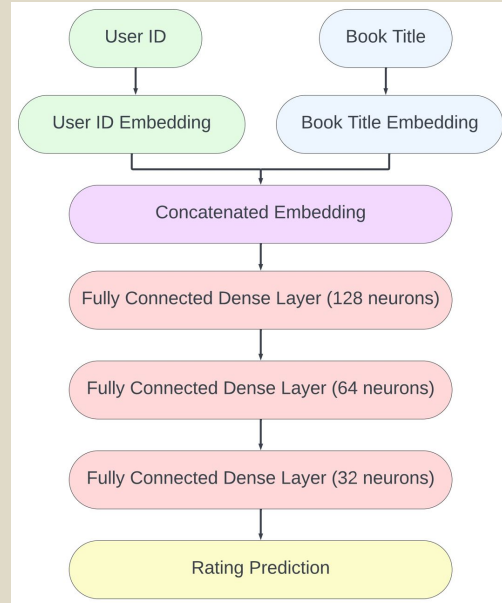


# TensorFlow

# Recommendation Model using Tensorflow

## Model Design:

- There are two input features to the model (Book Title, User ID)
- The model predicts the ratings of books based on the user ID.
- The data is divided into 80% training data and 20% testing data.



# First Iteration

For the first iteration of the model,

- the three datasets are merged and the columns “Locations”, “Image-URL-S”, “Image-URL-M” and Image-URL-L are dropped
- the Book-Title column is encoded using LabelEncoder
- maximum number of User IDs and maximum number of Book-Titles are used for the embedded layer vector
  - *Embedding layer converts positive integers (indexes) into dense vectors of fixed size.*
- three dense layers with 128, 64 and 32 neurons each are used
- loss is calculated using mean squared error and optimized using adam optimizer
- the model is run for 3 epochs
- dropout layers are used in between the dense layers
  - *The dropout layer randomly sets input units to 0 with a frequency of rate at each step during the training to avoid overfitting.*
- the training data has a loss of 7.6% and the testing data has a loss of 12.6%



# Model Summary

Layer (type)	Output Shape	Param #	Connected to
input_2 (InputLayer)	[(None, 1)]	0	[]
input_1 (InputLayer)	[(None, 1)]	0	[]
embedding_1 (Embedding)	(None, 1, 50)	1205355 0	['input_2[0][0]']
embedding (Embedding)	(None, 1, 50)	1394275 0	['input_1[0][0]']
flatten_1 (Flatten)	(None, 50)	0	['embedding_1[0][0]']
flatten (Flatten)	(None, 50)	0	['embedding[0][0]']
concatenate (Concatenate)	(None, 100)	0	['flatten_1[0][0]', 'flatten[0][0]']
dense (Dense)	(None, 128)	12928	['concatenate[0][0]']
dropout (Dropout)	(None, 128)	0	['dense[0][0]']
dense_1 (Dense)	(None, 64)	8256	['dropout[0][0]']
dropout_1 (Dropout)	(None, 64)	0	['dense_1[0][0]']
dense_2 (Dense)	(None, 32)	2080	['dropout_1[0][0]']
dense_3 (Dense)	(None, 1)	33	['dense_2[0][0]']
=====			
Total params: 26019597 (99.26 MB)			
Trainable params: 26019597 (99.26 MB)			
Non-trainable params: 0 (0.00 Byte)			

# Model Performance

## Training the model

```
Epoch 1/3  
1612/1612 [=====] - 1068s 661ms/step - loss: 11.8857 - mse: 11.8857 - mae: 2.8092  
Epoch 2/3  
1612/1612 [=====] - 1090s 676ms/step - loss: 9.6164 - mse: 9.6164 - mae: 2.3632  
Epoch 3/3  
1612/1612 [=====] - 1069s 663ms/step - loss: 7.5621 - mse: 7.5621 - mae: 1.9105
```

## Testing the model

```
6445/6445 [=====] - 28s 4ms/step - loss: 12.6602 - mse: 12.6602 - mae: 2.6943  
Loss: [12.660223960876465, 12.660223960876465, 2.694279193878174]
```



# Recommendations

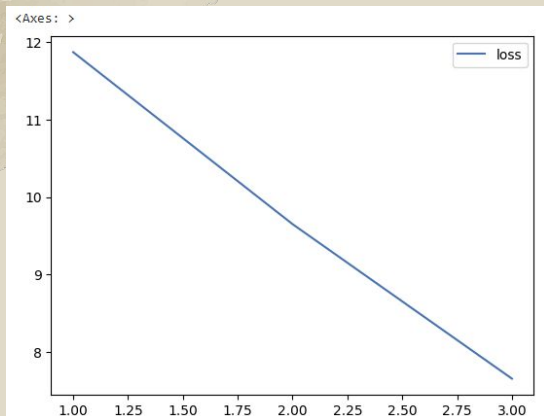
Below are the top 10 recommendations for the book “**The Secret Life of Bees**”

Top 10 recommendations for book The Secret Life of Bees

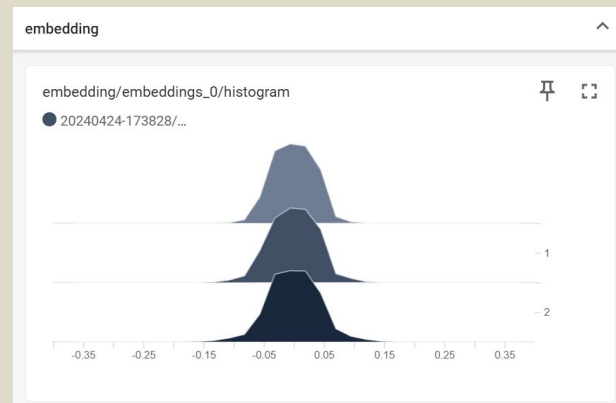
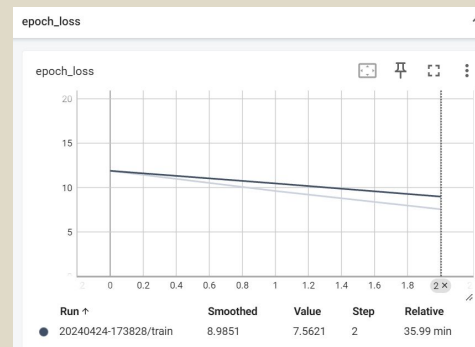
	Book-Title	Book-Author	User-ID	Book-Rating
780909	The Miracle	Irving Wallace	226362	10
780900	Creating Love: The Next Great Stage of Growth	John Bradshaw	226362	10
780908	Thy Neighbor's Wife	Gay Talese	226362	10
780893	A Civil Action	JONATHAN HARR	226362	10
780892	The Partner	John Grisham	226362	10
780907	Dark Crimes	Ed Gorman	226362	10
780896	Eye of the Beholder	Lowell Cauffiel	226362	10
780906	The Canfield decision	Spiro T Agnew	226362	10
780905	Needles	William Deverell	226362	10
780904	The Haldeman Diaries: Inside the Nixon White H...	H.R. Haldeman	226362	9

# Model Performance Visualizations

## Loss graph



## TensorBoard Visuals






# Model Optimization Attempt

- The model optimization was done using a smaller dataset.
- The dataset was cleaned and filtered for the KNN model.
- Loss in training data is 10.9%.
- Loss in testing data is 11.8%.

## Observations

- The loss in training data is higher compared to the original dataset.
  - The loss in testing data is lower compared to the original dataset.
  - The performance is better in terms of response time for the model prediction (1m 16s for original model and 15s for optimized model).
  - The recommendations made by the optimized model is different from the recommendation made by the original model for the same input.
  - The recommendations of tensorflow model is different from the recommendations of KNN model.
- 

## Model Summary

Layer (type)	Output Shape	Param #	Connected to
input_4 (InputLayer)	[(None, 1)]	0	[]
input_3 (InputLayer)	[(None, 1)]	0	[]
embedding_3 (Embedding)	(None, 1, 50)	88600	['input_4[0][0]']
embedding_2 (Embedding)	(None, 1, 50)	1394260 0	['input_3[0][0]']
flatten_3 (Flatten)	(None, 50)	0	['embedding_3[0][0]']
flatten_2 (Flatten)	(None, 50)	0	['embedding_2[0][0]']
concatenate_1 (Concatenate )	(None, 100)	0	['flatten_3[0][0]', 'flatten_2[0][0]']
dense_4 (Dense)	(None, 128)	12928	['concatenate_1[0][0]']
dropout_2 (Dropout)	(None, 128)	0	['dense_4[0][0]']
dense_5 (Dense)	(None, 64)	8256	['dropout_2[0][0]']
dropout_3 (Dropout)	(None, 64)	0	['dense_5[0][0]']
dense_6 (Dense)	(None, 32)	2080	['dropout_3[0][0]']
dense_7 (Dense)	(None, 1)	33	['dense_6[0][0]']
=====			
Total params: 14054497 (53.61 MB)			
Trainable params: 14054497 (53.61 MB)			
Non-trainable params: 0 (0.00 Byte)			

Total Parameters  
(Original Model):  
**26,019,597**

Total Parameters  
(Optimized Model):  
**14,054,497**

# Optimized Model Performance

## Training the model

```
Epoch 1/3  
292/292 [=====] - 116s 391ms/step - loss: 13.2622 - mse: 13.2622 - mae: 2.9475  
Epoch 2/3  
292/292 [=====] - 126s 431ms/step - loss: 11.2300 - mse: 11.2300 - mae: 2.6824  
Epoch 3/3  
292/292 [=====] - 110s 377ms/step - loss: 10.9191 - mse: 10.9191 - mae: 2.6180
```

## Testing the model

```
1166/1166 [=====] - 3s 2ms/step - loss: 11.8036 - mse: 11.8036 - mae: 2.7978  
Loss: [11.803647994995117, 11.803647994995117, 2.7977802753448486]
```

# Recommendations using Optimized Model

Below are the top 10 recommendations given by the model.

Top 10 recommendations for book The Secret Life of Bees

	Book-Title	User-ID	Book-Rating
164431	The Summons	244685	10.0
164420	The Children of Men	244685	9.0
164418	The Angel of Darkness	244685	9.0
164417	The Absence of Nectar	244685	9.0
164416	Summer Light	244685	9.0
164415	Still Waters	244685	8.0
164414	Second Child	244685	8.0
164413	Ransom	244685	8.0
164412	Praying for Sleep	244685	8.0
164411	Phantoms	244685	8.0

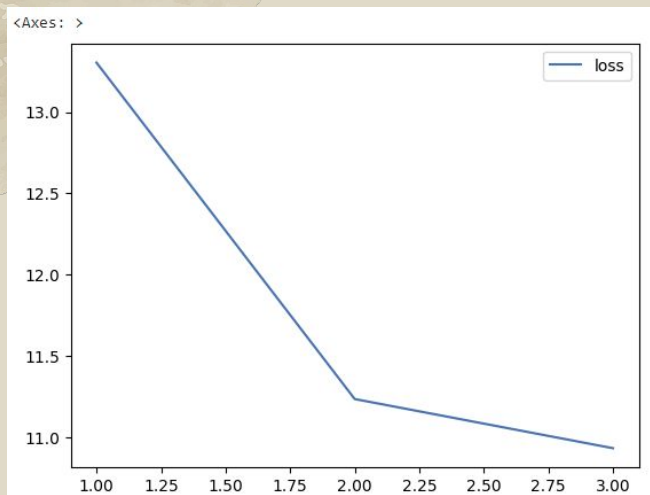
Top 10 recommendations for book Jurassic Park

	Book-Title	User-ID	Book-Rating
164400	Black and Blue	244685	10.0
164424	The Killing Game: Only One Can Win...and the L...	244685	8.0
164417	The Absence of Nectar	244685	9.0
164418	The Angel of Darkness	244685	9.0
164419	The Brethren	244685	10.0
164420	The Children of Men	244685	9.0
164421	The Door to December	244685	8.0
164422	The Funhouse	244685	8.0
164423	The General's Daughter	244685	5.0
164425	The Magic of You (Malory Novels (Paperback))	244685	8.0

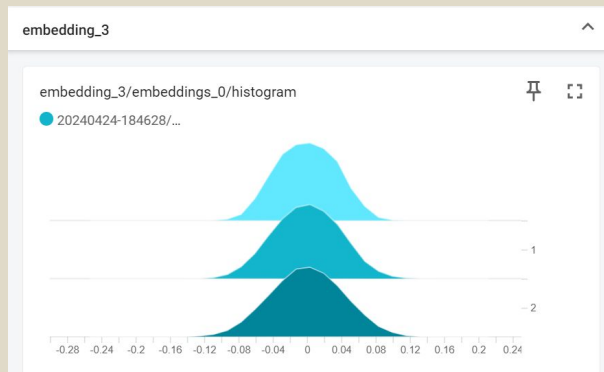
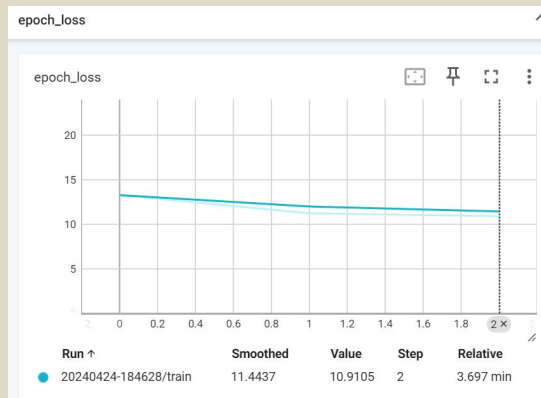


# Optimized Model Performance Visualizations

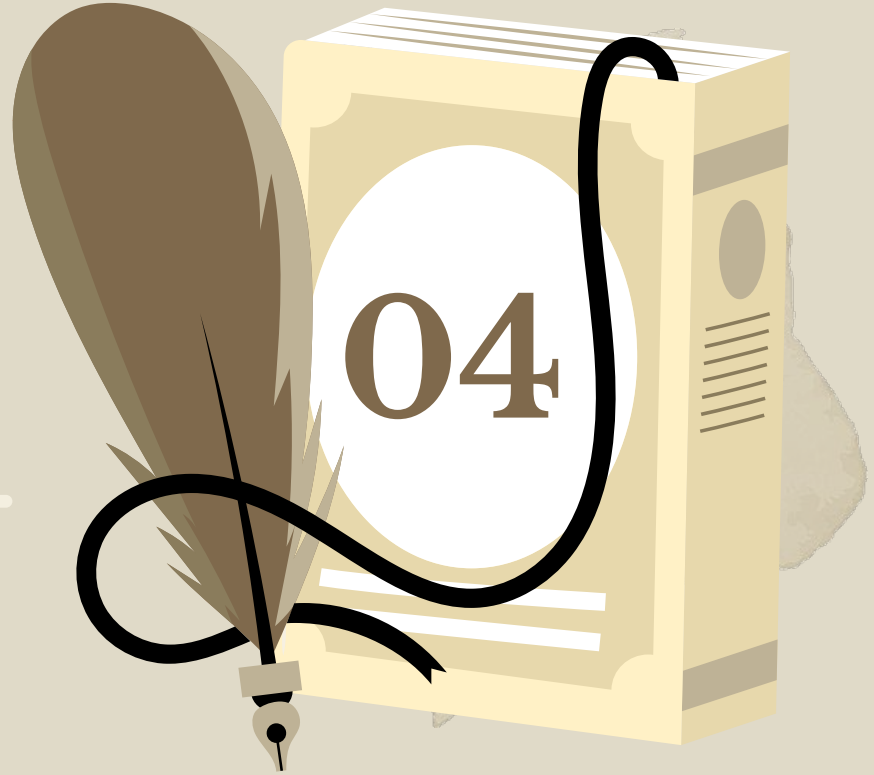
## Loss graph



## TensorBoard Visuals



# Conclusions





# Conclusions

1

## KNN Model

Book recommendations from the KNN model are intuitive and have predictive value.

Cold-Start problem: Popular books will be recommended more often, with newer/less known books either being filtered out entirely or underrepresented.



## Tensorflow Model

2

Tensorflow models are robust and can be used with large training and testing datasets.

But, with large datasets, performance is an issue. Also, Tensorflow model can be more difficult to learn and use.



05

# References

# References

## **Book Recommendation Dataset:**

<https://www.kaggle.com/datasets/arashnic/book-recommendation-dataset>

## **Github:**

<https://github.com/korzenim/proj-4-team-3/tree/main>

<https://medium.com/@milana.shxanukova15/cosine-distance-and-cosine-similarity-a5da0e4d9ded>





# UI Presentation