Design and Development of a Book Recommendation System

BIA - 660 Web Mining

Mukunth Rajendran
Shraddha Barde
Amey Thombre
Vignesh Tirumalai

Under the noble guidance of Prof. Rong Liu

Stevens Institute of Technology
Hoboken, NJ 07307, USA.

Table of Contents

I. Motivation

II. Objective

III. Introduction

- Roadmap
- Business Implication

IV. Related Work

V. What's New

VI. Methodology

VII. Performance Evaluation

VIII. Analysis of Results

- What part of the methodology worked (or didn't work)
- Why did the methodology work or (didn't work)
- How to improve?
- Utilization of results

IX. Conclusion and future work

MOTIVATION

The face of the book retailing industry has transformed remarkably. Book Retailing has found its potential market and has transitioned its operations through online channels after the dissemination of the Internet. With the rise of this new online business, book retailing companies are constantly focusing on innovation through technology to deliver an overall enhanced customer experience to its users.

In this market, Goodreads has emerged as one of the potential players of the market since its inception in 2007. Goodreads provides a wide range of books from different categories and genres and also very diverse usability to its users.

With so much evolution and constant developments even today what Goodreads lack is an acute category identification system. This is a major issue that most of the players in the industry are struggling with and we thought of addressing this issue from the organization's perspective. Moreover, another prevailing issue is with the top reads recommendation module. The existing models are not as apt as required from a user's perspective.

Furthermore, today almost every organization is focusing on considering and integrating emotions into their legacy systems. Consider all these parameters and aspects, we were motivated to bring out solutions for better operations and placed our focus towards designing and developing an overall recommendation system that focuses on our three requirements to address

the issues faced by the organization, the user and inject the emotion to pace up the business of Goodreads.

OBJECTIVE

The most primary objective of the project focuses towards the case when a new book comes into the market it is a bit difficult for some players of the market to determine which category the book should be placed. In this scenario, what most players do is that they have to either rely on experts who work on the genre or category identification or they have to expect it to be supplied from the publisher or author. Our recommendation system focuses on placing this book into the most accurate category to be a bestseller in the market through a completely automated process.

Secondly, another objective that we are focusing is towards, when a user visits Goodreads he will be furnished with significant recommendations and top reads. This sharp top reads recommendation module utilizes a metric which is a linear combination of many factors which helps in determining a top list of must-read books.

Furthermore, we have focused our objective towards harnessing the polarity, subjectivity, compound, negative, neutral and positive factors and most importantly the emotion of the book.

In totality, we want to provide the customers of Goodreads with a very superior and enhanced user experience through serving them with the best quality literature by establishing the recommendation system which solves problems associated with the three target areas that we discussed in the motivation.

INTRODUCTION

This project aims at providing Goodreads with the ability to integrate acute recommendations modules into their legacy systems. This integration will enable Goodreads to achieve superior productivity and proficiency in the industry.

The first module involves the development of an automated category identification feature for a new entrant book. This will enable Goodreads will a seamless ability to place every new book that comes in the market into the desired and respective category with the desired outcome to make it as a bestseller.

The second module involves the development of a top reads or best reads recommendation system which can replace the existing generalized recommendation module on Goodreads. Here, the new model will determine a suitable book based on the books the users have read and match the content of the book with similar books and throw up a recommendation. Presently, most of the users in the market are operating in this recommendation segment by tracking the users' search pattern and not on the content of the book and the theme of the book.

Lastly, in the final module, we are focused on integrating the emotion factor into the Goodreads usability. Upon deployment of this, we shall be able to determine what is the emotion of a particular book.

To achieve these three different modules that sum up the entire recommendation system, we follow a project roadmap which splits the modules into a number of working stages that help in building the system. Firstly, in the phase of project initialization, we first understand the

potential of the data and what we can do with it. Post which we focus towards scrapping the data and work towards data pre-processing which helps us deal with null values, duplicity, etc. In the second stage, we are working towards categorization which is followed by Classification as the third. In classification, we have utilized one versus one classifier, SVM linearSVC and Convolutional neural networks - Deep Learning. In the fourth stage, we place our emphasis towards sentiment analysis where we scrapped six different emotions from power thesaurus following which we work towards deriving the emotion for each book from the reviews collected. After this, we work towards feature derivation and proceed towards procuring the top 15 books from the data.

Once these steps were accomplished we worked on content-based filtering based on descriptions and a very important concept of Item-based Collaborative Filtering because the data is very large and we need very specific results for every user such as one book should throw five books as a recommendation. In the final stage, we proceed towards evaluation and optimization. We have explained each of these briefly introduced steps with appropriate details and results in this script further.

This recommendation system will help attract new customers by providing what they are looking for. It will also help reduce the search and surf time of customers. This is an important factor as when it comes to the e-commerce business segment, the major user base loss is due to loss of interest of users because of ambiguity in generating required outcomes. It will also generate opportunity for business alliances with publishers and authors that helps in forming collaborative business elements thereby increasing the market value of Goodreads amongst the readers. Also, a requirement of experts which help in deciding in which category the book needs to be placed is

eliminated due to the automated process. Moreover, The ability to bring out the emotion of the books proves to be a technological breakthrough that will help as a promoting element in driving the business further. Overall, this will help Goodreads have a competitive edge in the market based on their sharp focus towards delivering better usability for the customers.

Related Work

A significant amount of work has been done by academicians, industry experts and researchers across the globe in the area of book recommendation by using the average rating, and the number of ratings each book received. Further related work in association with this project involves some recommendation systems for books that have been built based on the correlation. There are some recommendation systems which are based on the rating counts. Some recommendation systems place their focus based on age distribution to recommend books based on the age group. For example, adventure or thriller or science fiction base genres to users lying the age group of 20 to 30 years.

Secondly, the existing recommendation systems that have been adopted by major online book retail websites are related and focused on tracing the search pattern and recommending books based on the user's search, etc.

There is some really remarkable work done in the book recommendation domain such as the book recommendation system based on Collaborative filtering and Association Rule Mining. We have also included the concept of collaborative filtering into our project to yield concrete results.

WHAT'S NEW

What new were are doing is that we are focusing on the content of the book, the theme of the book which helps reveal the true essence of the book. This provides a better edge in the recommendation system as we get to have a close association with the user requirements and not just leads of the user's requirement such as ratings, reviews, search traces, etc. We have also incorporated the use of a user metric using a weighted average formula which most of the websites use to list out the top popular books in the market and is currently lacking in our case.

METHODOLOGY

I. WEB SCRAPING

The following is the sequence of steps that were involved in scraping the data from Goodreads:

STEP I: Initially we are scraping the 27 categories that are available on Goodreads.

STEP II: Each category is then explored to scrap the first 300 books. This includes the Category, names, links, scores, and the number of people voted.

STEP III: The list of books in each category is stored in a CSV file along with the corresponding details.

STEP IV: The name, author, author profile, awards, description, ratings of each book and other categories (this column specifies what other categories the book is being listed in - This is

supplied by the user) is extracted.

To have a summary of the scrapping process -

- 8700 records were scraped
- 300 book per category was scraped
- A total of 27 categories
- The important features of each book such as the reviews, reviewedby, reviewedprofile and likes are stored in a CSV file for 300 books in each category.

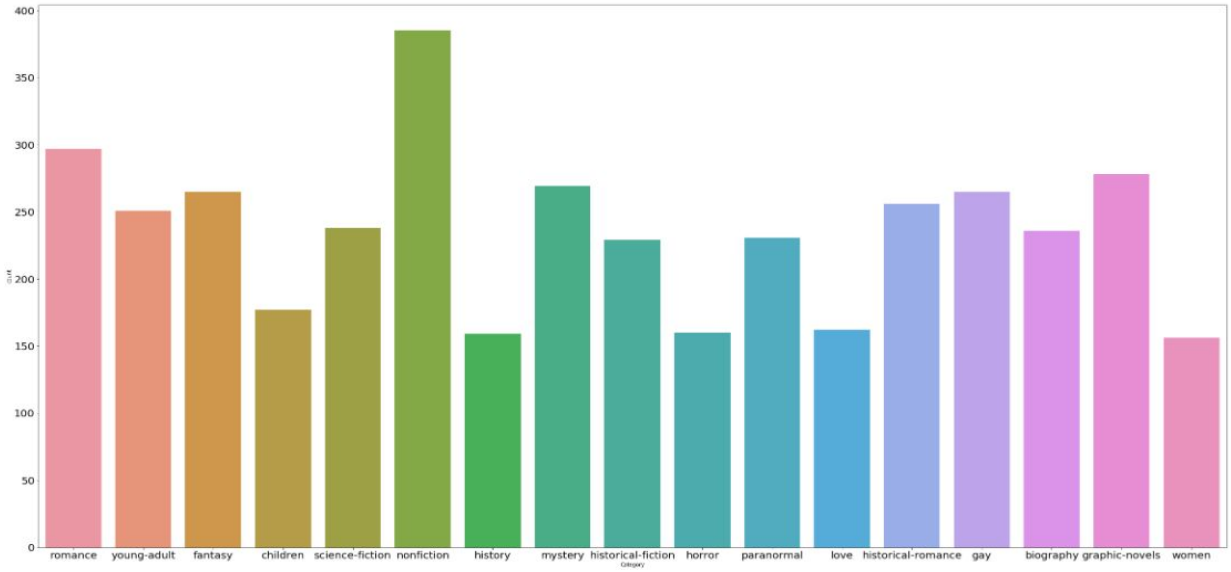
DATA PRE-PROCESSING

- Converted the entire textual to the lower case.
- Removed the null values.
- Dropped duplicate records.
- Created a column called “Training Text” using the various features like description, other categories, Author name, and book name.
- Worked on the low performing categories and added weights to the important features

II. Exploratory Data Analysis

With the help of Exploratory data analysis, We tried to understand the distribution of the books among the derived 17 categories that was scraped from Goodreads.

The following is one of the results we acquired from the Exploratory data analysis process -



III. Emotions

In this stage,we categorized the wide range of emotions into a subset of 6 core emotions. Love, surprise, anger, happy, sadness and fear. For this we used https://www.powerthesaurus.org to scrape a list of 500 unique keywords for each emotion. We tried to maintain a list of unique words for each category at the same time keeping each list with the same number of keywords. We used the CountVectorizer to get the word count and divided it by the total number of words of the reviews for a single book. (Term Frequency) (Normalized)

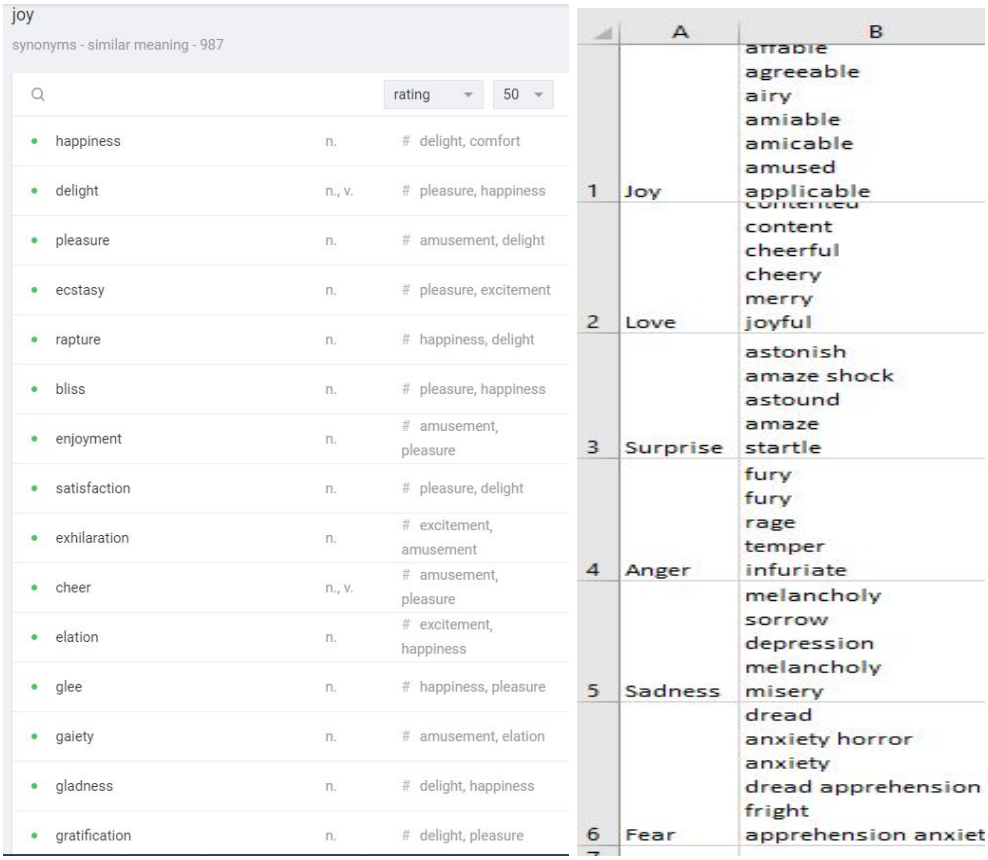


Fig: Snapshot of how the extraction process from power thesaurus helped us build a dictionary of words pertaining to various categories such as Joy, Love, Surprise, Anger, Sadness, fear.

IV. Deriving the features from the scraped data

```
Book Name : Fifty Shades of Grey (Fifty Shades, #1)
Book Review text
0.03458902730040215
0.5233158909919973
compound: -0.9954,
neg: 0.135,
neu: 0.728,
pos: 0.137,
Love,Joy,Anger,Fear
Book Name : Hopeless (Hopeless, #1)
Book Review text
0.18532178680436207
0.6000859301544365
compound: 1.0,
neg: 0.108,
neu: 0.681,
pos: 0.211,
Love,Joy,Sadness,Fear
Book Name : The Fault in Our Stars
Book Review text
0.0895342225364329
0.5543051519916277
compound: -0.9991,
neg: 0.158,
neu: 0.69,
pos: 0.152,
Love,Joy,Anger,Sadness
Book Name : Divergent (Divergent, #1)
Book Review text
0.09827392692355072
0.5306313694690039
compound: 1.0,
neg: 0.104,
neu: 0.729,
pos: 0.168,
Love,Joy,Surprise,Anger
```

	Name	Polarity	Subjectivity	Compound	Negative	Neutral	Positive	Emotion
0	Beautiful Disaster (Beautiful, #1)	0.03234042	0.5694315	-1	0.157	0.71	0.133	Love,Joy
1	Fifty Shades of Grey (Fifty Shades, #1)	0.03458903	0.5233159	-0.9954	0.135	0.728	0.137	Love,Joy
2	Hopeless (Hopeless, #1)	0.18532179	0.6000859	1	0.108	0.681	0.211	Love,Joy
3	The Fault in Our Stars	0.08953422	0.5543052	-0.9991	0.158	0.69	0.152	Love,Joy
4	Divergent (Divergent, #1)	0.09827393	0.5306314	1	0.104	0.729	0.168	Love,Joy
5	Slammed (Slammed, #1)	0.15986865	0.5706714	1	0.101	0.698	0.201	Love,Joy
6	Effortless (Thoughtless, #2)	0.20623599	0.5756692	1	0.089	0.69	0.22	Love,Joy
7	Easy (Contours of the Heart, #1)	0.16577932	0.5672696	1	0.116	0.691	0.193	Love,Joy
8	Bared to You (Crossfire, #1)	0.12701953	0.5566326	1	0.1	0.739	0.16	Love,Joy
9	Thoughtless (Thoughtless, #1)	0.06450958	0.5742567	-0.9996	0.173	0.668	0.159	Love,Joy
10	The Hunger Games (The Hunger Games, #1)	0.12850926	0.5197938	1	0.117	0.735	0.148	Love,Joy
11	Fallen Too Far (Rosemary Beach, #1; Too Far, #1)	0.12863951	0.5586169	1	0.111	0.699	0.19	Love,Joy
12	The Edge of Never (The Edge of Never, #1)	0.14973561	0.5443326	1	0.109	0.704	0.187	Love,Joy
13	Obsidian (Lux, #1)	0.15951647	0.5966561	1	0.099	0.677	0.224	Love,Joy
14	Walking Disaster (Beautiful, #2)	0.12320702	0.5485445	1	0.129	0.71	0.161	Love,Joy
15	City of Bones (The Mortal Instruments, #1)	0.08289836	0.5441439	1	0.099	0.739	0.162	Love,Joy
16	On Dublin Street (On Dublin Street, #1)	0.15839425	0.5547096	1	0.106	0.698	0.196	Love,Joy
17	Clockwork Angel (The Infernal Devices, #1)	0.11285989	0.5526003	1	0.104	0.708	0.187	Love,Joy
18	Pride and Prejudice	0.15063587	0.5304424	1	0.096	0.727	0.176	Love,Joy
19	Anna and the French Kiss (Anna and the French Kiss, #1)	0.12795272	0.5600886	1	0.115	0.688	0.197	Love,Joy
20	Hush, Hush (Hush, Hush, #1)	0.05024653	0.5397661	-0.9998	0.145	0.718	0.137	Love,Joy
21	Vampire Academy (Vampire Academy, #1)	0.16686075	0.5756972	1	0.123	0.696	0.181	Love,Joy
22	Wallbanger (Cocktail, #1)	0.16595423	0.5997667	1	0.098	0.685	0.216	Love,Joy
23	Fifty Shades Freed (Fifty Shades, #3)	0.10452274	0.516469	1	0.109	0.71	0.181	Love,Joy
24	Gabriel's Inferno (Gabriel's Inferno, #1)	0.13750045	0.571671	1	0.086	0.713	0.201	Love,Joy
25	Reflected in You (Crossfire, #2)	0.13541421	0.5568131	1	0.112	0.721	0.167	Love,Joy
26	Fifty Shades Darker (Fifty Shades, #2)	0.07430943	0.4862657	1	0.111	0.727	0.162	Love,Joy

Fig: Feature Derivation

V. CATEGORIZING

- **One vs One Classifier**

We are categorizing the books based on their tags.

In the one-vs.-one (OvO) reduction, one trains K (K − 1) / 2 binary classifiers for a K-way multiclass problem; each receives the samples of a pair of classes from the original training set and must learn to distinguish these two classes. At prediction time, a voting scheme is applied: all K (K − 1) / 2 classifiers are applied to an unseen sample and the class that got the highest number of "+1" predictions gets predicted by the combined classifier.

A precision of 77% was achieved when we tried to one vs one classification.

	precision	recall	f1-score	support
biography	0.86	0.89	0.88	28
children	0.75	0.56	0.64	16
fantasy	0.86	0.69	0.76	35
gay	0.86	0.78	0.82	23
graphic-novels	0.90	1.00	0.95	27
historical-fiction	0.77	0.90	0.83	30
historical-romance	0.93	0.96	0.94	26
history	0.81	0.68	0.74	25
horror	0.60	0.35	0.44	17
love	1.00	0.56	0.72	16
mystery	0.89	0.86	0.87	28
nonfiction	0.52	0.70	0.60	33
paranormal	0.72	0.82	0.77	22
romance	0.60	0.71	0.65	21
science-fiction	0.76	0.84	0.80	19
women	0.50	0.46	0.48	13
young-adult	0.54	0.61	0.57	23
avg / total	0.77	0.75	0.75	402

Fig: Tabulation of Precision, recall, F1 - score and support.

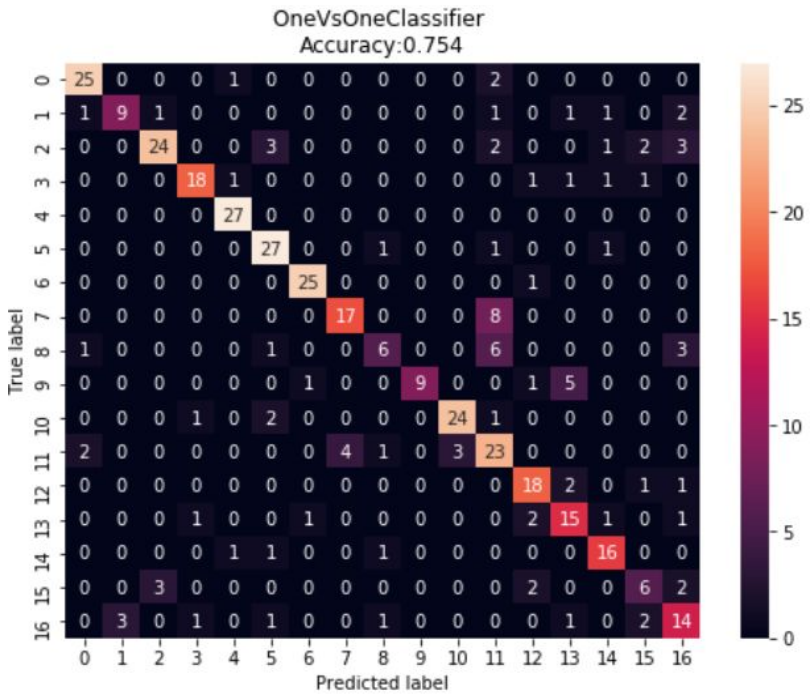


Fig: Confusion Matrix

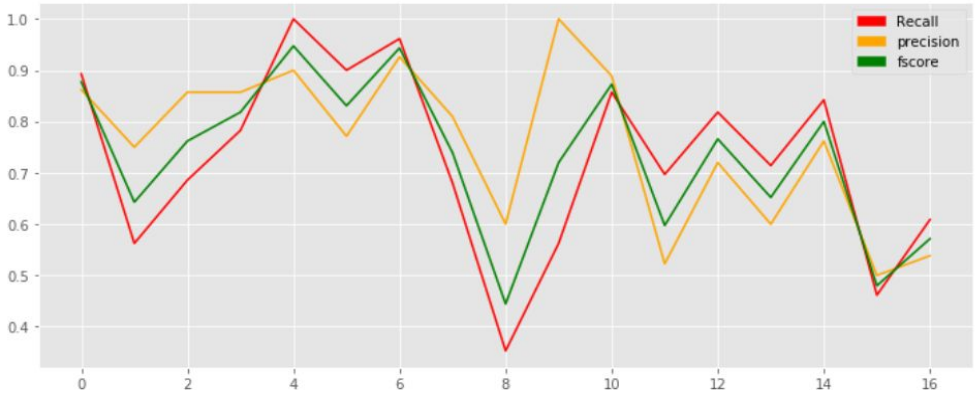


Fig: Precision, recall and F1-score plot of various categories

- **SVM Linear SVC**

We are categorizing the books based on their tags.

A regular SVM with default values uses a radial basis function as the SVM kernel. This is basically a Gaussian kernel aka bell-curve. Meaning that the no man's land between different classes is created with a Gaussian function. The linear-SVM uses a linear kernel for the basis function, so you can think of this as a ^ shaped function. It is much less tunable and is basically just a linear interpolation.

A precision of 75% was achieved when we tried to SVM LinearSVC classifier.

	precision	recall	f1-score	support
biography	0.83	0.86	0.84	28
children	0.75	0.56	0.64	16
fantasy	0.90	0.74	0.81	35
gay	0.80	0.87	0.83	23
graphic-novels	0.90	1.00	0.95	27
historical-fiction	0.71	0.83	0.77	30
historical-romance	0.87	1.00	0.93	26
history	0.79	0.76	0.78	25
horror	0.60	0.53	0.56	17
love	0.90	0.56	0.69	16
mystery	0.82	0.82	0.82	28
nonfiction	0.58	0.64	0.61	33
paranormal	0.80	0.73	0.76	22
romance	0.58	0.67	0.62	21
science-fiction	0.76	0.84	0.80	19
women	0.58	0.54	0.56	13
young-adult	0.45	0.43	0.44	23
avg / total	0.75	0.75	0.75	402

Fig: Tabulation of Precision, recall, F1 - score and support.

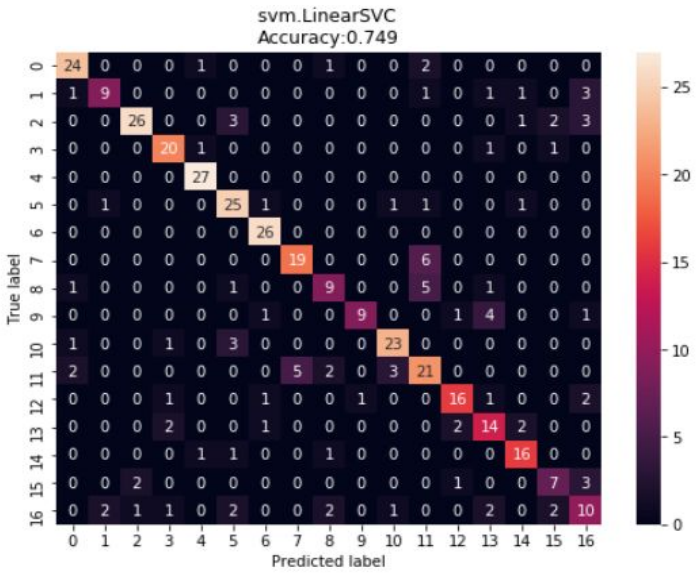


Fig: Confusion Matrix

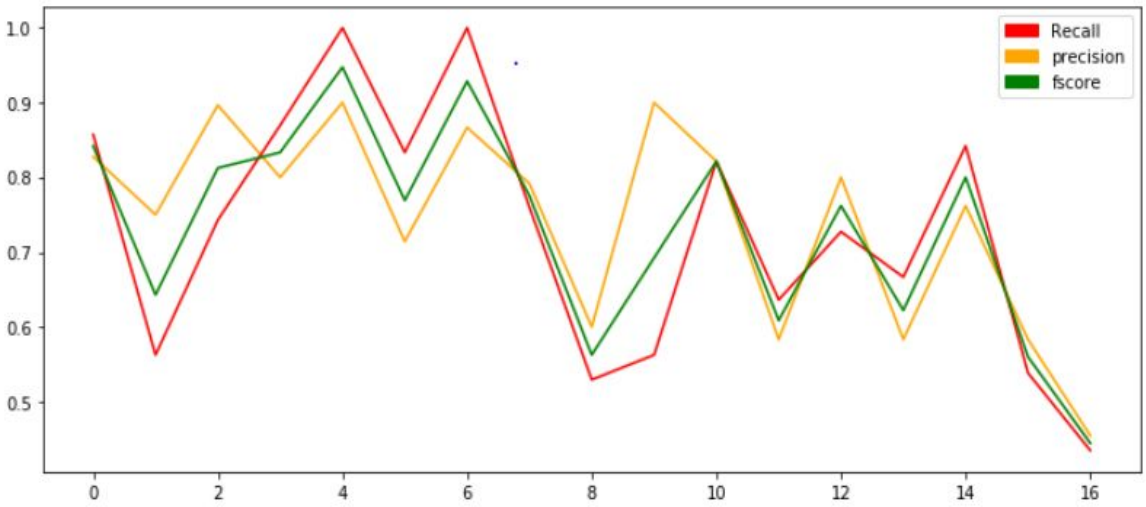


Fig: Precision, recall and F1-score plot of various categories

V. Convolution Neural Network – Deep Learning

Multi-Label Classification using CNN - Keras

We have made use of Keras which is an open source neural network library in Python. For the process of Multi-Label Classification using convolutional neural networks, we converted each document to a list of word index as a sequence following which we went towards padding all sequences into the same length. We created a CNN model by altering the filter size, the total number of words, max words in the document, vector dimension, drop-out rate, number of units in the dense layer and number of output units.

We compared the performance of our model with and without the Pre-trained word of vectors.

```
Train on 3644 samples, validate on 912 samples
Epoch 1/20
 - 66s - loss: 0.5681 - acc: 0.6975 - val_loss: 0.5272 - val_acc: 0.7341

Epoch 00001: val_loss improved from inf to 0.52724, saving model to best_model
Epoch 2/20
 - 70s - loss: 0.4535 - acc: 0.7818 - val_loss: 0.3696 - val_acc: 0.8313

Epoch 00002: val_loss improved from 0.52724 to 0.36963, saving model to best_model
Epoch 3/20
 - 63s - loss: 0.3602 - acc: 0.8392 - val_loss: 0.3311 - val_acc: 0.8489

Epoch 00003: val_loss improved from 0.36963 to 0.33109, saving model to best_model
Epoch 4/20
 - 63s - loss: 0.3279 - acc: 0.8571 - val_loss: 0.3137 - val_acc: 0.8606

Epoch 00004: val_loss improved from 0.33109 to 0.31370, saving model to best_model
Epoch 5/20
 - 64s - loss: 0.2975 - acc: 0.8718 - val_loss: 0.3043 - val_acc: 0.8625

Epoch 00005: val_loss improved from 0.31370 to 0.30425, saving model to best_model
Epoch 6/20
 - 61s - loss: 0.2749 - acc: 0.8838 - val_loss: 0.3003 - val_acc: 0.8678

Epoch 00006: val_loss improved from 0.30425 to 0.30029, saving model to best_model
Epoch 7/20
 - 61s - loss: 0.2429 - acc: 0.9000 - val_loss: 0.2979 - val_acc: 0.8687

Epoch 00007: val_loss improved from 0.30029 to 0.29792, saving model to best_model
Epoch 8/20
 - 60s - loss: 0.2120 - acc: 0.9152 - val_loss: 0.3006 - val_acc: 0.8698

Epoch 00008: val_loss did not improve from 0.29792
Epoch 00008: early stopping
```

Fig: Early stop implemented in Epoch to achieve validation accuracy

```
# predict
pred=model.predict(X_test)
# evaluate the model
scores = model.evaluate(X_test, Y_test, verbose=0)
print("%s: %.2f%%" % (model.metrics_names[1], scores[1]*100))
```

```
acc: 86.98%
```

Fig: Accuracy of the model - 86.98%

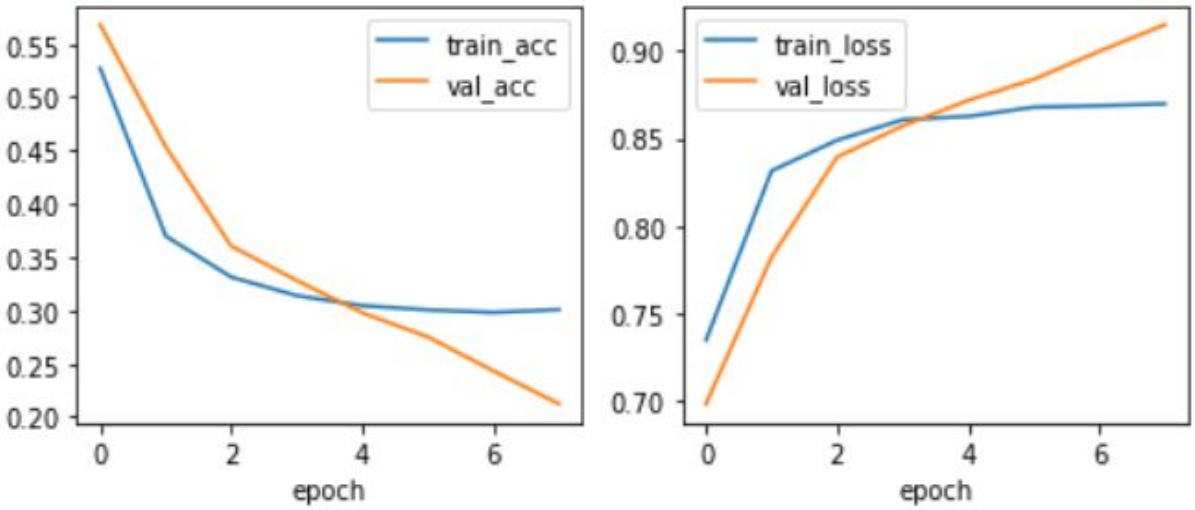


Fig: Training & validation dataset - Accuracy & loss

VI. Simple Recommendation System

In this step, we worked towards working on deriving the top 15 books using metrics.

So basically, The simple recommendation system is a system that recommends the top Books based on a certain metric.

$$\text{weighted rating } (WR) = (v/(v+m))R + (m/(v+m))C$$

```
### Finding the Wieghted Rating -- Aviod a situation of less voters and better scores (WR) = (vv+m.R)+(mv+m.C)
# v - No.of Voters (people who voted)
# R - Ratings
# C - Mean of all the Ratings
# m - minimum votes required
```

This metric is used by best recommendation system websites, so we tried implementing it in our systems. We decided to use the “Weighted Rating method” to calculate the metric. The metric is derived using the metadata of the books that we scrapped. The normalizing measure is done to make sure that the decision towards recommendation isn’t taken in a biased manner using the ratings alone.

Technical Implementation Process:

STEP I: Calculating the mean of all the ratings, which is denoted as “C”.

STEP II: Calculating the minimum number of votes required, denoted as “m”. -- Calculating 90 percentile

STEP III: Filtering all the qualified books into a dataframe based on a scale, if “v” is greater than or equal to “m”.

STEP IV: Creating a function ‘weighted_rating()’ that computes the weighted rating of each book.

STEP V: Creating a new feature called 'Metrics' and calculate its value with ‘weighted_rating()’

STEP VI: Sort Books based on the score calculated above

STEP VII: Print the top 15 Books based on the Ratings from all the categories.

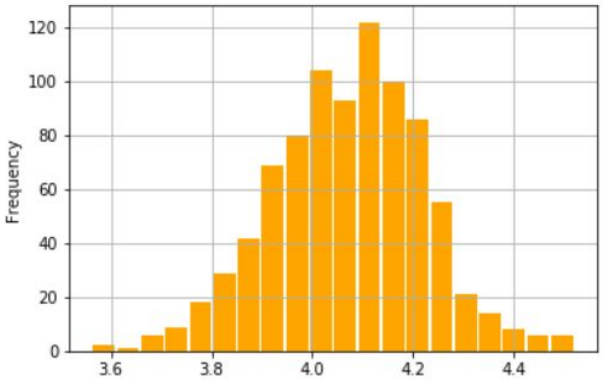


Fig: Frequency of the metrics

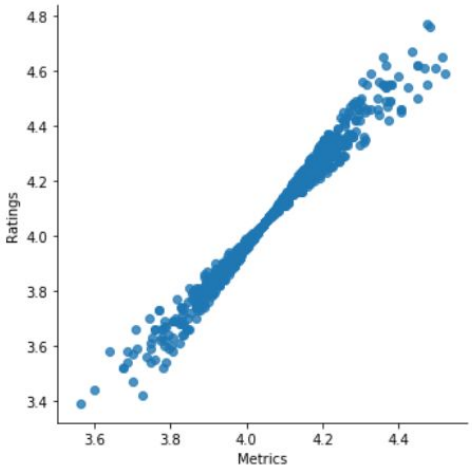


Fig: Positive correlation between the metric and ratings

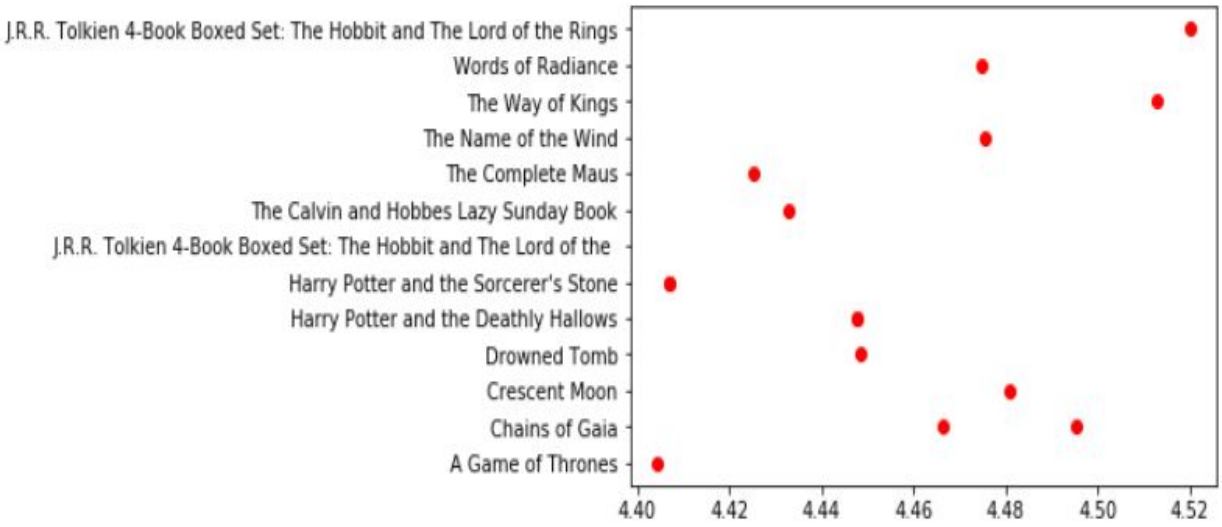


Fig: Top-rated books compared with the metrics

VII. Content-Based Recommendation

In this section, we focus towards building a system to recommend books that are similar to a particular book. We found the index of the Book given its title. Then we got the list of cosine similarity scores for that particular movies with all movies and converted it into a list of tuples where the 1st element is its position and the 2nd element is the similarity scores. We computed Pairwise Similarity Scores for all the books using the cosine similarity and sorted them based on the scores. The 9 most similar books are taken into consideration and recommended to the users.

Technical Implementation Process:

STEP I: We used the pre-processed data

STEP II: We used the TF-IDF vectorizer from the scikit-learn library and removed the stopwords

STEP III: We replaced the nan with empty strings

STEP IV: Constructed the TF-IDF matrix by fitting & transforming the data.

STEP V: Imported the linear_kernel to compute the cosine similarity matrix

STEP VI: We Construct a reverse map of indices and book titles

STEP VII: Asked for input 'Title'

STEP VIII: Get the index of the book that matches the 'Title'

STEP IX: Get the pairwise similarity scores of all the books with that book.

STEP X: Sorted the books based on the similarity score

STEP XI: Get the scores of fifteen most similar books.

```
Please enter the name of the Book in Lower Case: harry potter and the deathly hallows
272                harry potter and the half-blood prince
349                harry potter and the sorcerer's stone
411                  harry potter and the goblet of fire
2512                        harry potter series box set
455               harry potter and the chamber of secrets
403              harry potter and the prisoner of azkaban
301             harry potter and the order of the phoenix
5193                                 the leopard prince
3086                                       storm front
6802    the invisible wall: a love story that broke ba...
7423                              the tale of one bad rat
5250                                       as you desire
1303                                 the complete tales
5126                                tempt me at twilight
7090                                 harry the dirty dog
Name: Book Name, dtype: object
```

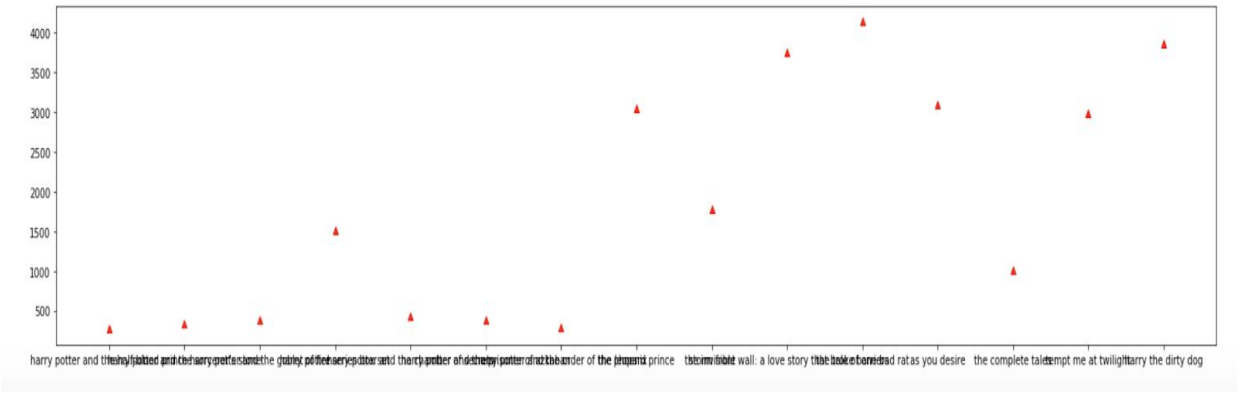


Fig: Book name and the indexes

VIII. Collaborative Filtering (Item - Item)

Here we work on, Recommending Books based on the books you like. A person who liked math book can get bored and start liking other books, but the math book is always similar to other math books. Hence, using this ideology, initially, we build a matrix with a column and a row for each user's rating and books (Most of the values in this matrix will be 'Nan' - Users wouldn't have read all the books). We replace those 'NaN' values to 0. Then we calculate the similarity between every book and store those values in an empty Data Frame. This Data Frame consists of both the column and row names being the 'Title of the book'.The intersection of the row and column represents the similarity of the books. We use Cosine

Similarity to find the distance between the books. Finally, with the cosine distance, we are able to find similar books.

Technical Implementation Process:

STEP I: Created a sample from the entire dataset with using the user, title and the rating.

STEP II: Created a pivot table having the index as the user, column as the title and values as the rating

STEP III: Handle the nan values (because not all readers read all the books)

STEP IV: Create a user review matrix and replace the nan values with zero.

STEP V: Finding the title similarities by creating a pandas dataframe having the index and columns

STEP VI: Got a matrix having column & row names as titles.

STEP VII: Look through each book and found the cosine similarities for each book.

STEP VIII: Get the index of each book

STEP IX: Most similar books are exported to a CSV file.

STEP X: The exported file consist of the book name and the books that have the highest similarity score.

Fig: Flowchart of how our collaborative filtering system works

Book Name	1	2	3	4	5
A Court of Mist and Fury	Wild Man	Come Away with Me	Going Too Far	Glass Houses	Knight
A Walk to Remember	Wild Man	Eclipse	Lover Eternal	Darkfever	Fifty Shades Trilogy
Acheron	Lover Unbound	The Dex-Files	Rock Chick Revenge	Rock Chick	The Last Olympian
All In	Lover Unbound	City of Glass	Release Me	My Life Next Doc	Seduction and Snack
Along for the Ride	A Court of Mist and Fury	Sweet Evil	Darkhouse	Fangirl	Reaper's Property
Angelfall	A Court of Mist and Fury	Sins & Needles	Going Too Far	Harry Potter and	The Marriage Bargai
Anna and the French Kiss	Bloodlines	The Sea of Tranquilit	Reflected in You	Clockwork Angel	The Hunger Games
Archer's Voice	Lover Unbound	Knight	The DUFF: Designated Ugly	Rock Chick	Release Me
Backstage Pass	Lover Unbound	Chain Reaction	Fangirl	City of Glass	Into the Hollow
Bared to You	The Boy Who Sneaks in M	If I Stay	Bloodlines	The Hunger Game	Catching Fire
Beautiful Bastard	Lover Unbound	Chain Reaction	Fangirl	City of Glass	Into the Hollow
Beautiful Creatures	Wild Man	Perfection	A Walk to Remember	City of Ashes	City of Glass
Beautiful Disaster	If I Stay	Reflected in You	Love Unscripted	The Hunger Game	The Boy Who Sneak
Beautiful Stranger	Rock Hard	Mockingjay	Harry Potter and the Death	Fallen Crest High	Eleanor & Park
Because of Low	The Truth About Forever	On the Island	Naked	My Favorite Mist	Because of Low
Beneath This Man	A Court of Mist and Fury	The DUFF: Designate	Fallen Crest High	Fifty Shades Trilo	From Ashes
Bloodlines	The Coincidence of Callie	Anna and the French	Slammed	The Mighty Storm	The Mortal Instrume
Breaking Dawn	Rules of Attraction	The Iron King	Dark Lover	That Boy	Just for Now
Breathe	Graceling	Pride and Prejudice	The Sea of Tranquility	The Edge of Neve	Rock Me
Catching Fire	If I Stay	The Sea of Tranquilit	The Hunger Games	The Fault in Our S	The Edge of Never
Chain Reaction	Lover Unbound	City of Glass	Release Me	My Life Next Doc	Seduction and Snack
Charade	Clockwork Princess	The Notebook	Lover Avenged	Lover Eternal	The Gamble
Checkmate	Wild Man	Going Too Far	Sweet Evil	Seduction and Sn	Reaper's Property
City of Ashes	Lover Unbound	Law Man	Ten Tiny Breaths	Sweet Evil	Sins & Needles

Fig: Book with top 5 recommended books.

PERFORMANCE EVALUATION

The performance evaluation of our model is a very cumbersome and complicated process. We consider doing it in near future. Hence in the absence of our evaluation of the performance, we thought of providing our readers with a real-time comparison of our model with the pre-existing model by means of a video.

Kindly refer the following links :

I. COLLABORATIVE FILTERING: https://www.youtube.com/watch?v=IDB9nIhDLa4

II. CONTENT-BASED FILTERING: https://www.youtube.com/watch?v=QmjehsYaq2s

III. BOOK RECOMMENDATIONS: https://www.youtube.com/watch?v=PsfLAjTllXI

EVALUATION BARRIERS

The following factors restrained us from evaluating our results better:

1. The constriction on the number of books (300) that were scrapped in each category.
2. Goodreads.com is a dynamic website, recently there was a change in the framework of the website since the time we had scraped the data.
3. The continuous activity of the users contributing to the change in the reviews and ratings of the book on a day to day basis.

ANALYSIS OF RESULTS

WHAT PART OF THE METHODOLOGY WORKED (OR DIDN'T WORK)

We made sure whatever we are incorporating into our methodology functioned appropriately.

WHY DID THE METHODOLOGY WORK OR (DIDN'T WORK)

To make sure our methodologies work we used the best practices followed in the development process of various recommendation systems currently in the market.
Example: IMDb

How to improve?

Looking at various other factors which can actually make an impact on the betterment of the system.

Utilization of results

We are focusing on not just building a model but paying attention to the minutest features that can make an impact and aid in making a model which stands as one of the best recommendation systems in the market. We are also working towards making our sample model unique and best amongst the rest.

Conclusion and future work

In Conclusion, we have achieved our three objectives of delivering category identification for a new entrant book in the market which can now be placed accurately in the most relevant category to be a bestseller in that particular segment or the market as a whole. Moreover, the top reads recommendation system will now enable users to receive top or best reads based on the content of the book. Furthermore, we have also worked towards the emotion factor which we have been emphasizing the most so as to let Goodreads understand the user and serve with the most apt products (i.e. books) and services.

In terms of the future work, the following is the progress that we envision to accomplish in the streamline of the current progress of the project.

1. We look forward to establishing a Chatbot facility for users to interact with the system to have the best recommendation
2. Moreover, we want to place our focus on throwing a recommendation to the user as soon as he hits the first visit of the day -

   "Today's Best read for the day tailored with love and wisdom by Goodreads just for you"
3. Also, work on establishing a learning pattern for new users based on a series of questions to make their profile apt for machine learning purposes.
4. Lastly and most importantly, work more on injecting emotions in the operability of Goodreads.

TASK LIST

Task Id	Tasks	Owner	Signature
1	Discuss project ideas	ALL TEAM MEMBERS	
2	Scrap data from GoodReads (Part 1 - Book Details)	Shraddha Bande	
3	Scrap data from GoodReads (Part 2 - Review Details)	Mukunth R Amey Thombre	
4	Scrap emotions from powerthesauraus.org	Amey Thombre Vignesh T	
5	Exploratory Data Analysis	Vignesh Tirmale	
6	Feature Derivation - Sentiment Analysis	Shraddha Bande Mukunth R	
7	OneVsOne classification	Vignesh Tirmale	
8	SVM Linear SVC	Amey Thombre	
9	Convoluted Neural Network - Deep Learning	Mukunth R Shraddha B	
10	Simple recommendation model (weighted rating)	Vignesh T Shraddha B	
11	Content based recommendation	Mukunth R	
12	Collaborative filtering (item to item)	Mukunth R	

****

THANK YOU