Ludwig Maximilian University of Munich

Winter semester 2023

Computational Social Science

Analyzing the sentiment and mood of the society using Newyork Times 2023 Bestsellers

Sameer Singh Rawat

MSc Statistics and Data Science

Matrikulation nr. - 12691600

sameer.rawat@campus.lmu.de

Github link - https://github.com/SameerR007/CSS_project

Table of Contents

1. Introduction

The motivation of this term paper is to investigate the themes and sentiments that book readers preferred in 2023. We assume that the themes and sentiments derived from our analysis can give us insight about the topics that are relevant in todays day and age and will also help us get some understanding of how people are thinking in current times. We also assume that this analysis should help us get some understanding of where we are heading as a species. For example if the themes that are relevant consist of words such as Artificial Intelligence then we can conclude that fields like data science are becoming more popular day by day, and in the days to come we can see more usage of technology to automate our mundane tasks.

In this term paper as the title suggests we will make an attempt to take a deep look at the bestsellers that were declared by Newyork times at the start of each month of the year 2023 and try to derive insights about readers behaviour. The dataset that we create will be done by the help of Newyork times api and the analysis will basically be done by the techniques named as Sentiment Analysis and Topic Modelling.

The overview of the steps involved in this project are summarized below -

1. The first step will be the creation of a custom dataset from scratch. This will basically be done with the help of api (newyork times and google books).

2. After the creation of data set we will start by modelling and analysis. We will try to gauze the sentiments that book readers prefer to read. This will be done by Natural Language technique named as sentiment analysis.

3. Also to delve into the behaviour of the readers we will try to see what kind of themes people have liked to read during 2023. This will be done by the technique named as Topic Modelling.

4. Then to put everything into a practical use case we will create a recommender system that will help user get a closest book from the Newyork times 2023 bestsellers to the book that they have already read.

5. To conclude we will put all the insights derived along the whole process in a concise manner to form our conclusion.

So by the end of this term paper we will be able to get some answer to the following research question - What sentiment and themes are more popular among the readers in the current times?

2. Methodology

2.1. Dataset

Newyork times publishes its bestseller every week. So in this project we will take the bestseller published by Newyork times at the starting of each month of 2023 and concatenate it to create the final dataset.

For this we will send an api request to Newyork times server using python's request function to get the top books of each month of 2023. From this we will get the result in json format which we will store in the form of a dataframe using python's json function. To take care of api call limits we can use python's time function for spacing between api calls. Here we will take into note that the book that has already appeared in the charts once, should be ignored the next times to avoid duplication.

After successful completion of these steps we will have a data set with book titles and their respective summaries. Also we will fetch a detailed description (referred as description in our data set) of the books that are in our initial data set using google books api. To get the importance of this step refer to the photo showing the summary and description of first three books depicted in the figure below.

index	Summary	Description
0	In the sequel to “It Ends With Us,” Lily deals with her jealous ex-husband as she reconnects with her first boyfriend.	PREVIOUS BOOK IN SERIES: IT ENDS WITH US, ISBN 9781501110368. Before 'It Ends with Us', it started with Atlas. Colleen Hoover tells fan favourite Atlass side of the story and shares what comes next in this long-anticipated sequel to the glorious and touching (USA TODAY) 'It Ends With Us'.
1	A battered wife raised in a violent home attempts to halt the cycle of abuse.	In this “brave and heartbreaking novel that digs its claws into you and doesn’t let go, long after you’ve finished it” (Anna Todd, New York Times bestselling author) from the #1 New York Times bestselling author of All Your Perfects, a workaholic with a too-good-to-be-true romance can’t stop thinking about her first love. Lily hasn’t always had it easy, but that’s never stopped her from working hard for the life she wants. She’s come a long way from the small town where she grew up—she graduated from college, moved to Boston, and started her own business. And when she feels a spark with a gorgeous neurosurgeon named Ryle Kincaid, everything in Lily’s life seems too good to be true. Ryle is assertive, stubborn, maybe even a little arrogant. He’s also sensitive, brilliant, and has a total soft spot for Lily. And the way he looks in scrubs certainly doesn’t hurt. Lily can’t get him out of her head. But Ryle’s complete aversion to relationships is disturbing. Even as Lily finds herself becoming the exception to his “no dating” rule, she can’t help but wonder what made him that way in the first place. As questions about her new relationship overwhelm her, so do thoughts of Atlas Corrigan—her first love and a link to the past she left behind. He was her kindred spirit, her protector. When Atlas suddenly reappears, everything Lily has built with Ryle is threatened. An honest, evocative, and tender novel, It Ends with Us is “a glorious and touching read, a forever keeper. The kind of book that gets handed down” (USA TODAY).
2	A scientist and single mother living in California in the 1960s becomes a star on a TV cooking show.	From advice columnist Meredith Goldstein, a dazzling, romantic, and emotionally resonant YA debut about a teen science whiz in Cambridge, Massachusetts, who tries to crack the chemical equation for lasting love and instead wreaks havoc on herself and the boys in her life. For seventeen-year old Maya, the equation for happiness is simple: a dream internship at MIT + two new science nerd friends + a perfect boyfriend = one amazing summer. Then Whit dumps her out of the blue. Maya is miserable until she discovers that her scientist mother, before she died, was conducting research on manipulating pheromones to enhance human attraction. If Maya can finish her mother’s work, maybe she can get Whit back. But when her experiment creates chaos in her love life, she realizes that maybe love and loss can’t be understood using the scientific method. Can she learn to trust the unmeasurables of love and attraction instead?

The need for this step arises because we see that summaries are very short which might not give good results for topic modelling and for the recommender system. On the other hand since the summary seems to talk solely about the story only it seems a sensible choice to use summaries to get the sentiment of the story. Note that description (fetched from google books api) is much more of a glorified review, ofcourse the book mentioned is in the top newyork times charts so the sentiment of such will likely to be most probably positive.

At the final stage of this project we have also fetched other information about the books like genres, authors (using google books api) so that we can make use of it to make content based recommendation. Note that since we have performed content based recommendation system so it was important for us to get these features. Also additionally we will fetch the google book image links to display the image of the recommended book in our cloud server just to make the things look more neat.

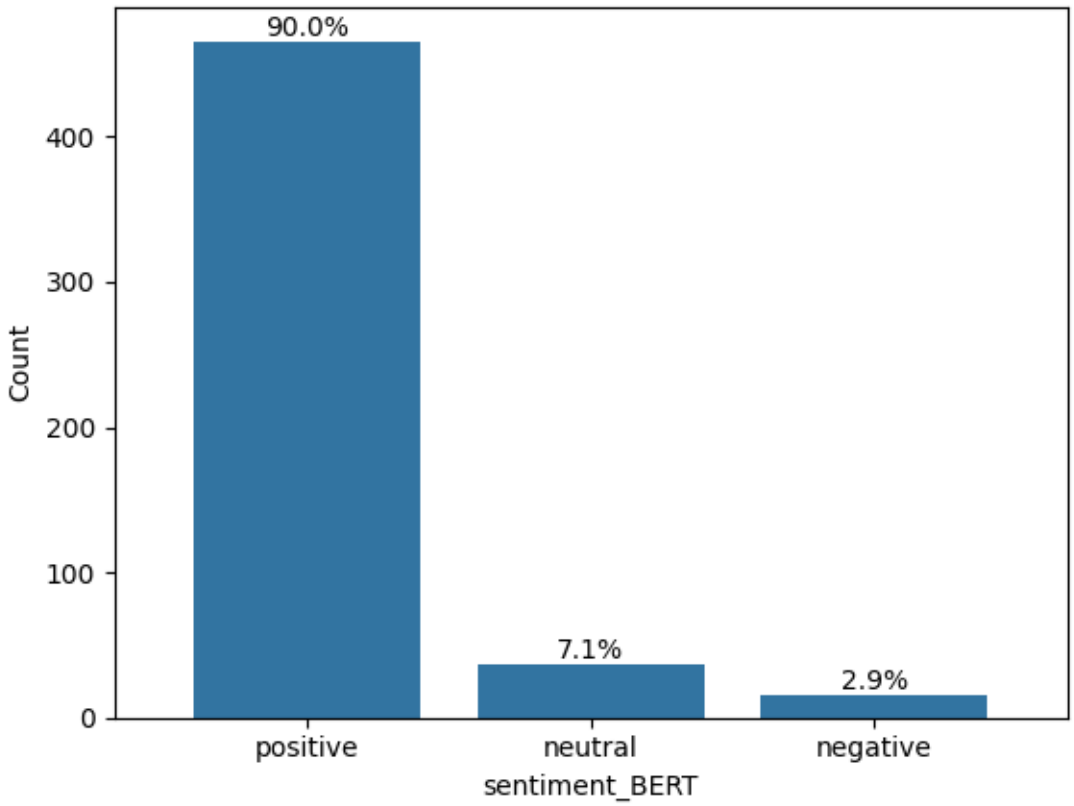
2.2 Sentiment Analysis

After preparation of the dataset the next thing we do is performing sentiment analysis on the summaries of respective books and store their corresponding sentiment. For doing so we would first be implementing it by a state of art model using Bidirectional Encoder Representations from Transformers (BERT) and also with our own built from scratch Recurrent Neural Network (RNN) model trained on a dummy labelled dataset, which is capable of performing low level sentiment analysis. We will then also have some discussion on the results obtained from both the techniques. Ofcourse, in the end we will use the results of BERT to make some speculations about readers behaviour and also use it in our final recommender system.

2.2.1 Sentiment analysis by BERT

Firstly we perform sentiment analysis using BERT. We will use transformers package to load BERT model and make predictions about sentiments of each summary. Since our BERT model outputs a number from 1 to 5 with 1 being most negative to 5 being the most positive. We classify the sentiment scores greater than 3 as positive, equal to 3 as neutral and less than 3 as negative.

From the result we see that out of 518 books in our dataset 466 are of positive sentiments, 37 are of neutral sentiments and 15 are of negative sentiments. The count plot for the same can be seen below.



From this we can very strongly indicate that readers like the books with positive sentiment.

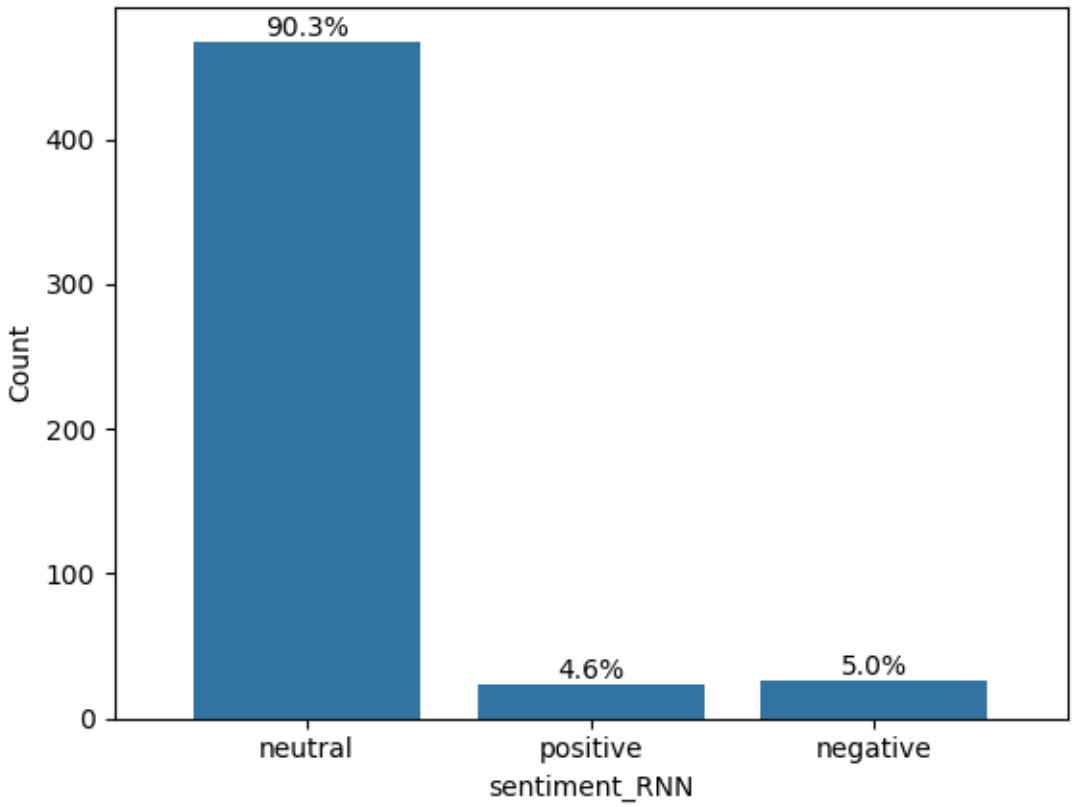
2.2.2 Sentiment Analysis by RNN

For training a recurrent neural network for sentiment analysis we firstly need a dataset. For this we take a dummy labelled dataset containing texts and their respective sentiments. The dataset can be found with name train.csv in the github repository of this project inside the data folder. The process of sentiment analysis using RNN contains several steps.

Firstly, we will denote each of the sentiment with value positive as 2, neutral as 1 and negative as 0. Then we will divide our dataset into 80:20 train test split with the help of scikit learn library in python. Then comes probably the most important task i.e. pre-processing of text data and converting it into the sequences of numbers that can be understood and processed by the computer. For the pre-processing we will make use of keras library in python. We will first convert each of the words in the vocabulary of our corpus into an index. Then we will convert each of the text in the dataset represented as the vector of these word index. For example "This is an example" can be represented as a vector [220 25 16 540] where 220 is the word index assigned to "This", 25 is the word index for "is", 16 is the word index for "an", and 540 is the word index for "example". Now since each of our inputs to RNN should of of same size we have to make each of the sentences of the same length here 22 (which is the median length of the sentences in the dataset). Note here that if sentences are of length less than 22 then it will be padded to make a total length of 22 by appending zeros. Now we are almost ready to create the architecture of our sentiment analysis using RNN. First step will be to create embedding layer using keras that will encode each of the words as a dense vector. We will represent each word as a dense vector of size 2. Taking our previous example, vector [220 25 16 540] will be represented as [[0.02 0.05] [0.03 0.45] [0.58 0.54] [0.9 0.01]] where each word is represented by a vector of dimension 2. Then we will feed this into our RNN layer of size 32 and finally through an output layer with 3 nodes using the softmax activation on it to get the values between 0 and 1 for each of the three nodes. We will train the dataset with loss as Sparse Categorical Cross Entropy and optimizer adam. After training on the train data and testing on the test data we get the validation accuracy of more than 80% in the

first 3 epochs itself. Hence, finally we will train again this time with whole dataset for 3 epochs and use this to make predictions on our summaries of the book.

We see the results we obtain here are very different from the results we obtained previously from BERT. With RNN out of total 518 books we get 468 books to be of neutral sentiment, 26 to be of negative and 24 to be of positive. The count plot for this can be seen below.



This difference in performance can be due to several factors such as bad selection of hyper parameters (the size of the RNN layer, the output dimension of embedding layer). This might make our model inefficient to classify the sentiments into positive or negative but classifying mostly as neutral.

2.3 Topic Modelling

Topic Modelling will help us derive the themes around which the popular books of 2023 revolved around. This is a very useful feature to help us give some clue about the readers behaviour. The themes derived from topic modelling can give us a great overview of the stories and narratives that were dominating peoples psychology and thinking in 2023.

As discussed earlier we would perform and compare the topic modelling on both summaries and detailed descriptions but we will give more weightage to the topic modelling performed on detailed description as they cover the information about the books in more detailed manner.

BERTopic is state of art technique that uses transformers and class based TF-IDF to create dense clusters. It allows to easily interpret and visualize the themes that are generated. To perform the topic modelling we will make use of the topic modelling library named bertopic in python. We have implemented the bertopic in our project with following lines of python code.

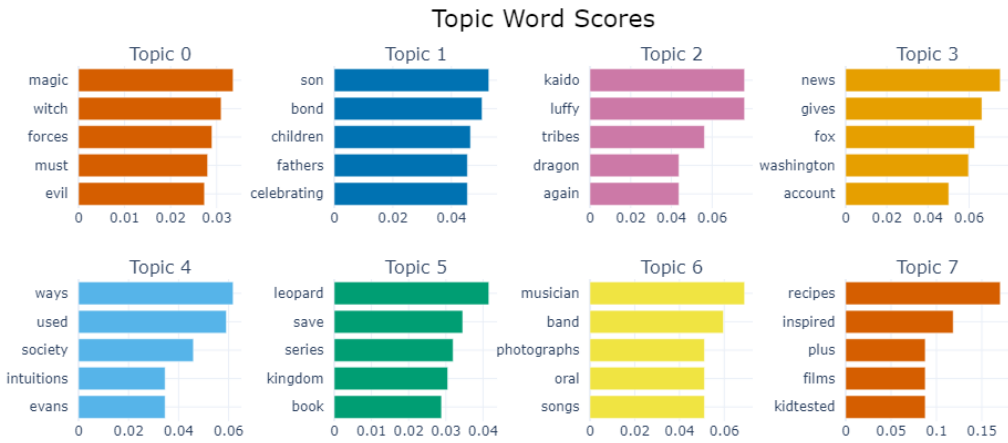
```
from bertopic import BERTopic
from sklearn.feature_extraction.text import CountVectorizer
from umap import UMAP
#vectorizer_model = CountVectorizer(stop_words="english")
umap_model = UMAP(n_neighbors=15, metric='cosine', low_memory=False,random_state=42)
topic_model = BERTopic(min_topic_size=2,top_n_words=7,umap_model=umap_model)
```

In this the min_topic_size=2 means that a topic must have atleast 2 documents referring to it for it to be considered as a theme. Since our dataset is of only 518 chunks of texts, taking a low value of 2 as the min topic size seems reasonable.

top_n_words=7 means that only top 7 words that make up a particular topic or theme will be displayed in the result. Choosing this value to be very less will give only a general context for a particular topic while choosing this to be a greater value will give more insights and more subtopics for a particular topic. 7 seems to be reasonable choice for this as it is not too shallow a number that it gives us a generic overview and neither too deep a number that we get lost in too much specifics.

2.3.1 BERTopic on summary

First we perform this on summaries and visualize the topics. The figure below depicts the output (note the visualize barplot have by default top_n_words=5)

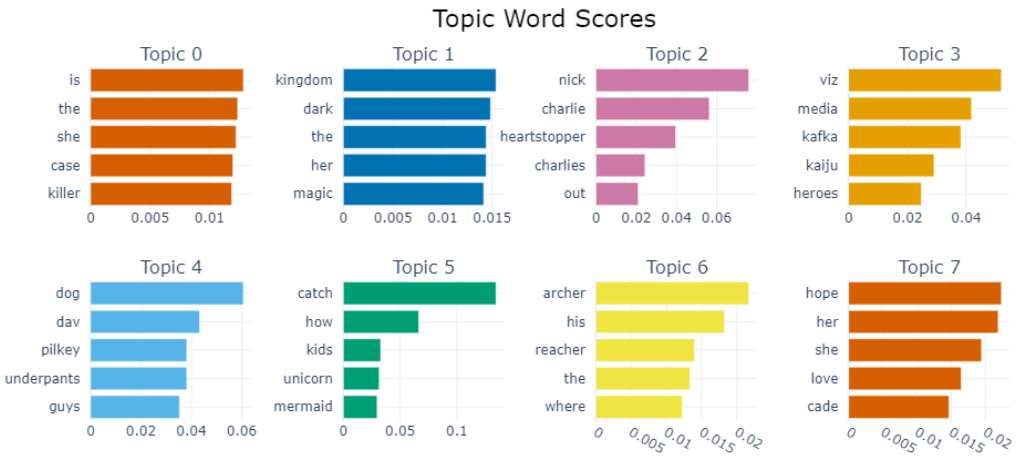


In the figure above the probable themes depicted by summary of the books are 1. Fantasy (as described by the words in Topic 0- magic, witch, forces, evil), 2. Family (as described by the words in Topic1 - son, bond, children, fathers), 3. Politics (as described by the words in Topic3 - news, fox, washington, account), 4. Art (as described by the words in Topic6 - musician, band, photographs, songs).

Hence summaries of the book depicts that the books that were popular among people in 2023 were those that had some fantasy elements in them or had some kind of family bonding in them, or had some kind of political notions attached to them or had some artistic elements in them.

2.3.1 BERTopic on descriptions

Similar to the above we run BERTopic on descriptions and again visualize the results. The results are visualized below



We perform the analysis again for the figure above. The probable themes depicted by descriptions of the books are 1. Crime (as described by the words in Topic 0- case, killer), 2. Fantasy (as described by the words in Topic1 and Topic5 - kingdom, dark, magic, unicorn, mermaid), 3. Kafka (as described by the Topic3), 4. Children (as described by the words in Topic5 - kids, unicorn, mermaid), 5. Positive sentiments (as described by the words in Topic7 - love and hope)

Hence descriptions of the book depicts that the books that were popular among people in 2023 were of crime genres, or had fantasy elements in them, or were written by author Franz Kafka, or were those that were written for the children, or had some positive sentiments like hope and love in their stories theme .

2.4 Recommender System

In this section we will try to create a content based recommender system that could give a tailored recommendation, similar to the book that has already been read by the user. The recommended book will be picked from the dataset we have already created. Along with the book we will display the recommended book's synopsis (i.e. the description from our dataset), the sentiment of the summary, the topic corresponding to the book that has already been derived from BERTopic trained on description (the word in this topic will be displayed as keywords). Finally we will deploy this on cloud so that everyone can use it. This section is just our attempt to have something practical at our hand at the end of this project.

As this is a content based recommendation system we first need to get all the features that will make up each of the book. The content we used are book author (as it plays an important role in describing the books), the book's genre and obviously book's description. We will pre-process the data to make it more suitable like we will remove the spaces between the authors name so that they can be treated as a unique entity when we combine all of our contents (i.e. author+genre+description). The combined features are stored as tags in our dataset. We will then apply stemming to this content (i.e. tags). This is a good practice so that we focus on the root words, and this will help our algorithm to understand and process the data better. The stemming will be done by PorterStemmer function imported from python's nltk library. Again as we discussed while building our RNN model we must find a way to encode this text data so that it can be understood and processed by our computer. For this we will use CountVectorizer function from python's scikit learn function. This will help us implement the Bag of Words technique to our text data. Bag of words represent each text data into a vector with number of times the most recent words appear in that sentence. To understand this concept look at the code below that we used to implement CountVectorizer in our project.

```
#to use count Vectorizer to encode the words so that our model can use it
from sklearn.feature_extraction.text import CountVectorizer
cv=CountVectorizer(max_features=4000,stop_words='english')
```

In this we keep the dimension of each text to be 4000 i.e. set by the max_features=4000. This means we will take the 4000 most frequent words that appeared in our text corpus and represent each text with a vector of these 4000 words. For example if text is represented as [0 2 0 ….0 1] then the most frequent word did

not appear in our text example, 2nd most frequent word appeared in our example text 2 times, and so on with 4000th most frequent word appeared in our example text 1 time. This forms each text as a vector of 4000 dimension and when applied on the tags of 497 books we get a matrix with dimension 497 x 4000. Since each of these 497 tags are vector we can find the similarity among them and this gives us a cosine similarity matrix of 497 x 497 which shows how much each of the 497 book is similar to other books in the dataset. Also note that in the CountVectorizer function we have set the hyperparameter stopwords = english which means we don't want to consider the frequent words like is,an,the in our text corpus while implementing Bag of words to find most frequent 4000 words.

Now everytime the user inputs we have to fetch its author, genres and description, and combine them to form a tag. Do the pre-processing in the similar way defined above and compare it with the cosine similarity matrix of dataset and display the most similar book's title, its similarity percentage(cosine similarity multiplied by 100), synopsis (description), sentiment(as calculated by BERT) and the similar keywords(as derived by BERTopic).

A prototype of the app is displayed below along with the link to access it.

New York Times Book recommendation

Takes a book that user has read as a input and gives recommendation based on 2023 NYT bestsellers.

Enter the book you have read

The Alchemist

Recommend

Book recomended-Dog Man: Grime and Punishment: A Graphic Novel (Dog Man #9): From th

Similarity percentage=26.79

Sentiment-positive

Synopsis

The mayor has had enough of Dog Man's shenanigans in the ninth book from worldwide bestselling author and artist Dav Pilkey. Dog Man's really done it this time! He hands over his badge and clears out his desk, but while he may be out of a job, he's not yet out of hope. With his friends at his side, can Dog Man dig himself out of this hole and paw his way back onto the force? Dav Pilkey's wildly popular Dog Man series appeals to readers of all ages and explores universally positive themes, including empathy, kindness, persistence, and the importance of doing good.

Keywords

dog,dav,pilkey,underpants,guys,man,captain

Link to the interface - https://huggingface.co/spaces/sameerrawat07/NYTbestseller

Link to hugging face space (files) -

https://huggingface.co/spaces/sameerrawat07/NYTbestseller/tree/main

3. Conclusion

To sum everything up this project was a great mix of modelling and analysis. Creation of the dataset from scratch was challenging and exciting at the same time. In the sentiment analysis we saw that readers prefer to read the stories with positive sentiments attached to them. This also correlates with our finding in themes generated by topic modelling where we saw the words such as hope and love emerging as popular themes. We saw people in 2023 are still liking the novels by author Franz Kafka. In genres we saw fantasy and crime, thriller being liked by people. It was through surprising to find that kid books with the stories about mermaid and unicorns being in the bestsellers.

Overall the books that are being read shows the overall mood and mindset of today's generation. Also it shows which topics and themes are currently being talked about and are being given attention to by the public. In a way it also helps us to see towards the future like in which direction we are heading to. This if implemented in a big enough dataset from various bestsellers and with more refined way can give us various insights about the society.

4. References

Sentiment Analysis - *https://www.analyticsvidhya.com/blog/2021/06/nlp-sentiment-analysis/*

Topic Modelling using BERTopic - *https://hackernoon.com/nlp-tutorial-topic-modeling-in-python-with-bertopic-372w35l9*

Content Based Recommendation System - *https://medium.com/@prateekgaurav/step-by-step-content-based-recommendation-system-823bbfd0541c#:~:text=Content%2Dbased%20recommendation%20systems%20are,their%20viewing%20and%20purchasing%20history.*