Packt Group Project

Aim:

To propose a system to analyze individual print book performance based on product, title, information and feedback publicly gathered from Amazon.

Datasets used:

Two datasets were used in the project, both of which were scraped from Amazon.
The first dataset had the following columns:

1. Book name
2. Book author
3. Rating (number of stars out of 5)
4. Number of reviews
5. Price
6. Discount percentage
7. Date of Publication

This dataset was primarily used for EDA and determining the factors responsible for a book to be successful in the market.

The second dataset contains the book information, book reviews, and details of the reviewer. This dataset was used for review sentiment analysis, EDA of reviews, and intent classification.

Tools and libraries used in the project

Programming language used: Python
Environment: Jupyter notebook
Python libraries used:

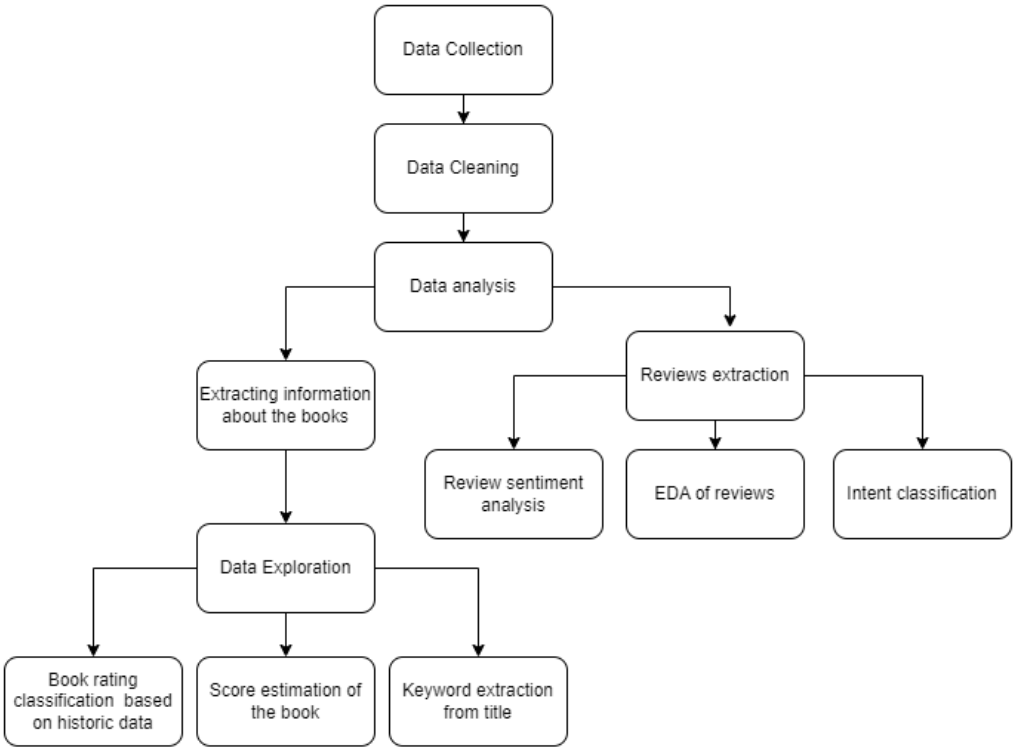
1. Pandas
2. Numpy
3. Matplotlib
4. Seaborn
5. Scikit-learn
6. NLTK

Web scraping tool used to scrape the data: Parsehub

Project Approach

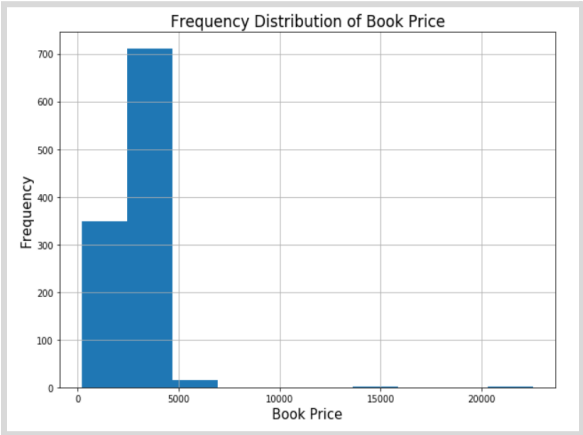
The following steps were followed to achieve the aim of the project:

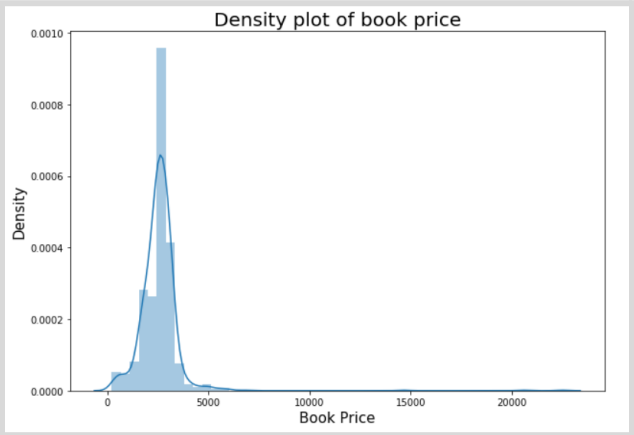
1. Data Scraping: The datasets scraped from the web using Parsehub
2. Data Cleaning: The scraped data was cleaned and brought into the desired format, dealing with the missing values and outliers.
3. Exploratory Data Analysis: Data Visualizations were made from the cleaned data to derive insights
4. Feature engineering: New features were generated from the existing features and data visualizations were made from the new features as well
5. Creating new score variable to determine the success of a book
6. Machine Learning model: A machine learning model was built to predict the score of a book on the basis of the book features
7. Sentiment and intent analysis: Sentiment and intent analysis of book reviews was done to predict the tone of the text and the sentiment it portrays.



Initial Data Visualizations

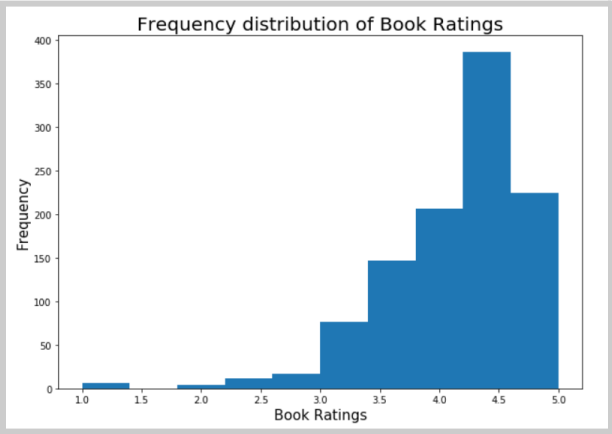
Univariate analysis of the Price column

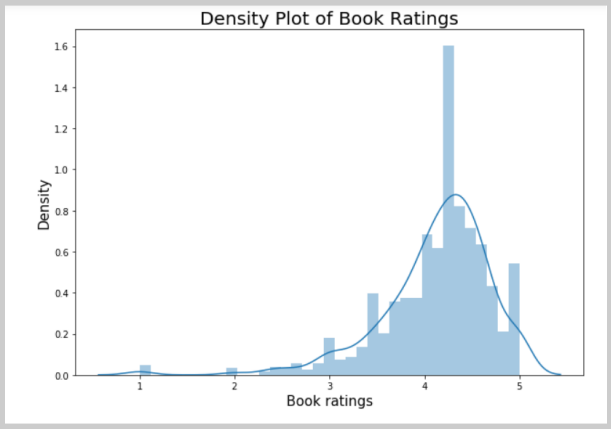




According to the histogram and density plot of book prices, most of the books are priced less than 5000 rupees. There are a very few books priced above 5000, and also a few exceptional books which cost greater than 10000.
The mean of the prices column was found to be 2572.41 rupees, while the median price is 2466 rupees.

Univariate analysis of Book ratings

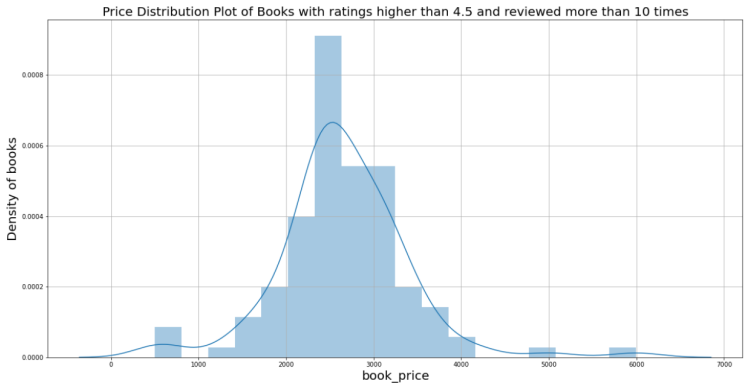




According to the Histogram and density plot of book ratings, the book ratings are left skewed. There are a few books rated below 3 stars, while the majority of the books are rated above 3.5.

The mean of all book ratings is 4.125 while the median is 4.2.

Price Distribution Plot of Books with ratings higher than 4.5 and reviewed more than 10 times

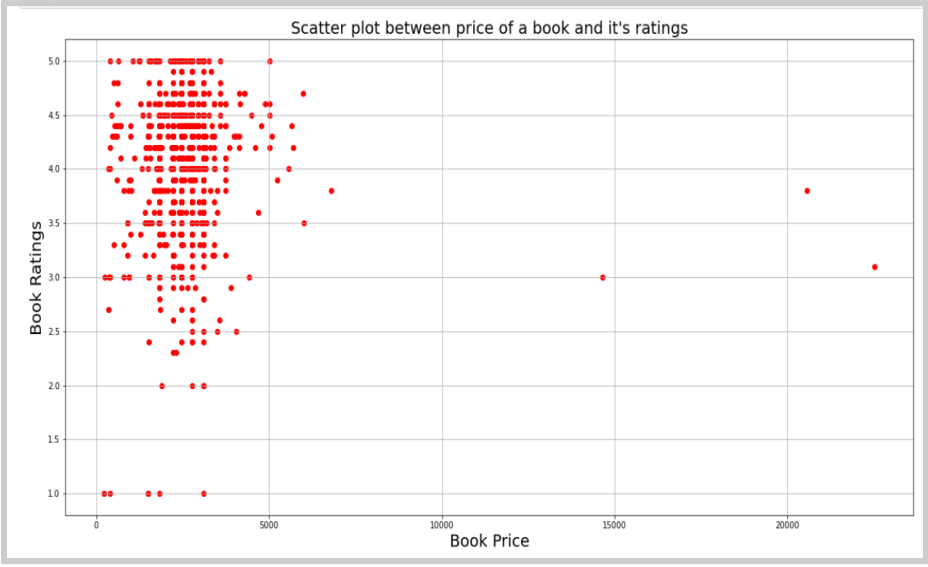


The price distribution of books with ratings higher than 4.5 and reviews more than 10 shows a pretty much normal distribution of data (slightly skewed to the left though).

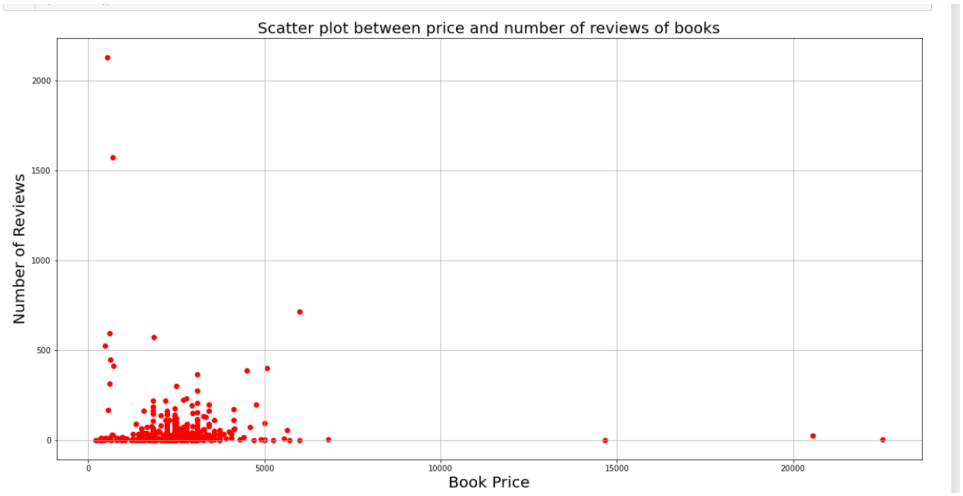
- It reflects that most of the books falling under this filter are priced between Rs. 2000-3000.
- Just a few books that are priced more than Rs. 3000 tend to perform well amongst the readers.

The arithmetic mean of price of books with ratings > 4.5 and more than 10 reviews is - 2652.269565217391
The median of price of books with ratings > 4.5 and more than 10 reviews is - 2466.0

Plot between the book price and book ratings



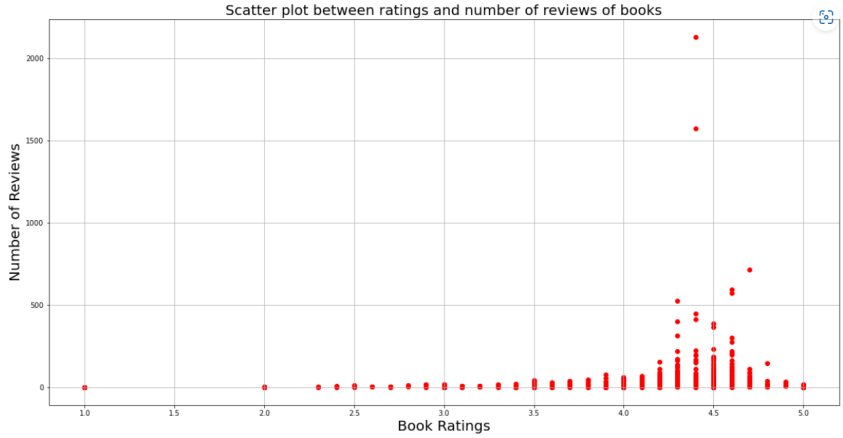
Plot between book price and number of book reviews



This scatter plot between the price and number of reviews of books shows that:

- Most of the people prefer reading the books that are not too expensive and since more people read the books in the price range of Rs. 2000-4000. they have the a consistent number of reviews.
- The books that are priced more than 5000 do not have many readers, thus not many reviews either.

Scatter plot between ratings and number of reviews of books

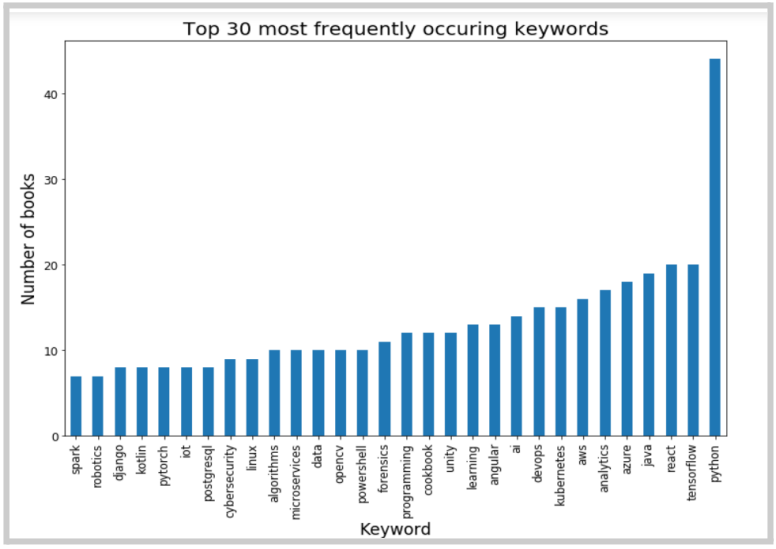


The scatter plot between ratings and number of reviews of a book goes onto tell you that -

- When a book has higher ratings, it is actually been reviewed by a significant number of people, and liked by them.
- The books with less reviews has less ratings, which reflects that people do not review a book when they don't like it.

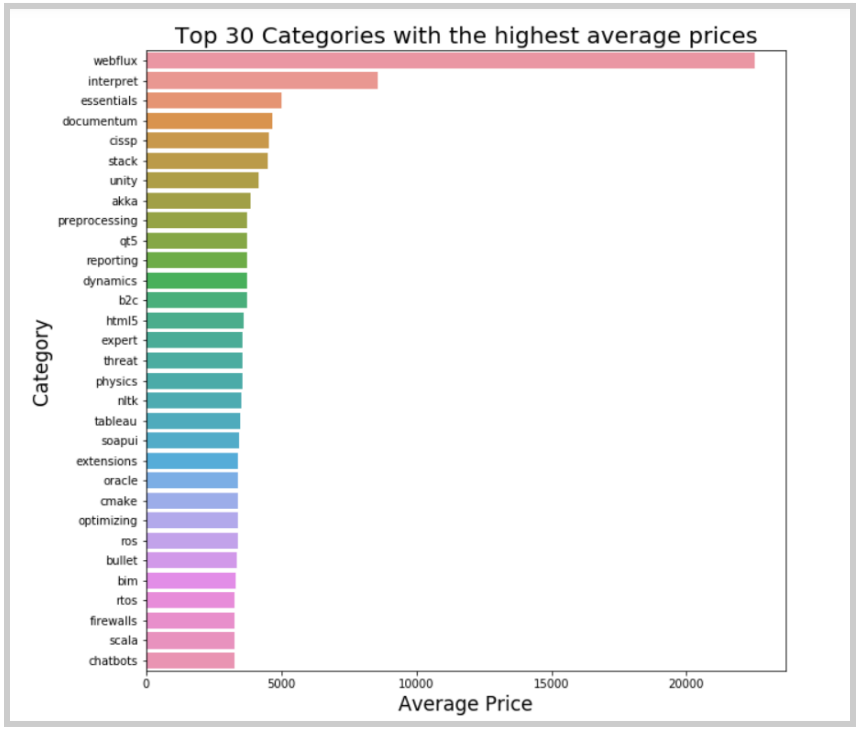
EDA based on keywords found in the book title

Top 30 most frequently occurring keywords in the book titles



- Python is the most frequently occurring keyword with more than 40 instances.
- Python, ReactJs, Tensorflow, Java, and Azure are trending topics among publications

Top 30 categories with highest average prices

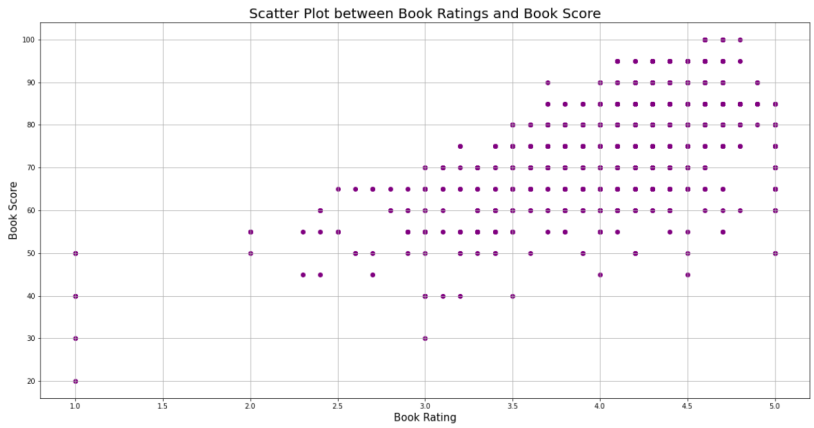


The average price of books on ‘webflux’ is the highest among all the categories.

EDA based on book score

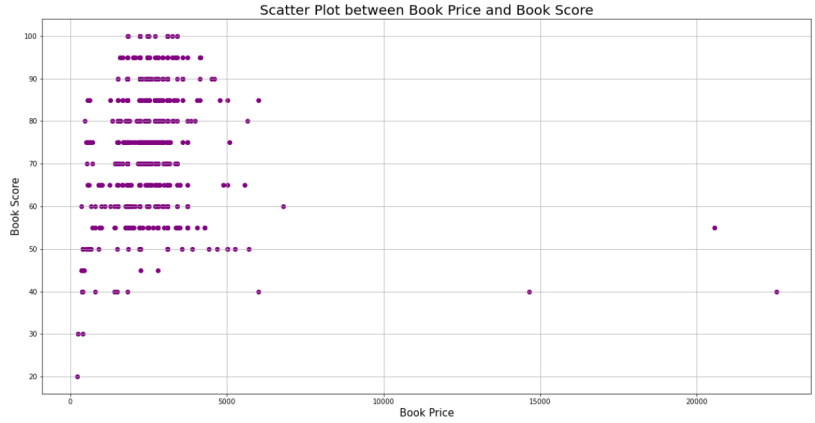
Score is an attribute which was incorporated with the information about the book. The score of a book varies on a scale of 0-100, reflecting the success of the book proportionally. This score was the result of the performance of the book over the period of time. Entities like ratings, number of reviews, number of years since it's released, and the price of the book are the key features, which have a great impact on the performance score of the book.

Scatter Plot between Book Ratings and Book Score



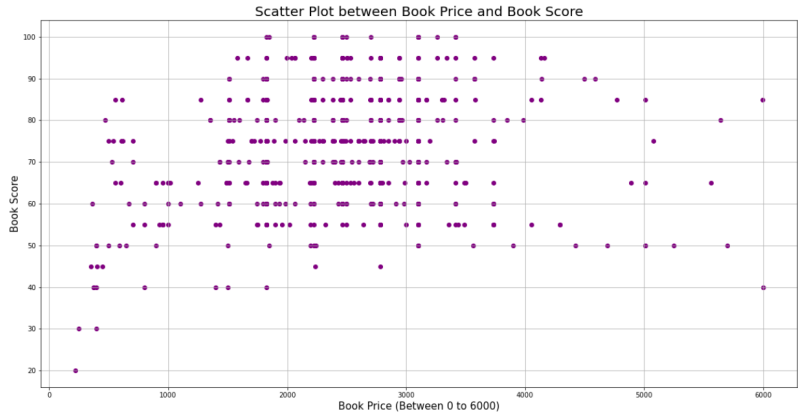
- This scatter plot clearly shows that the book ratings and directly proportional to the book score, higher the rating, higher the book score.

Scatter Plot between Book Price and Book Score



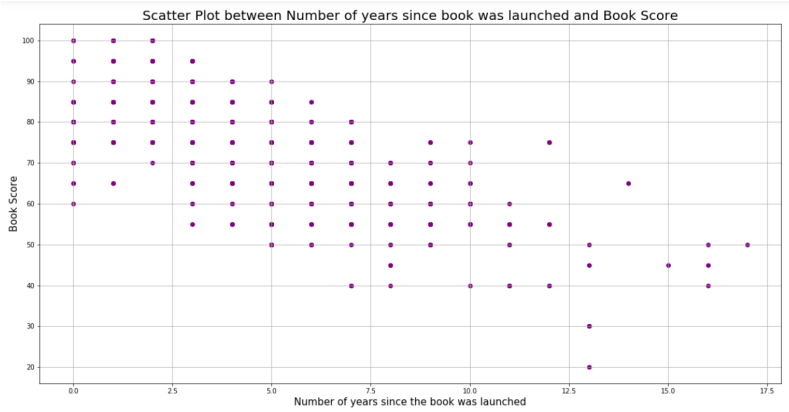
- The book price and book score are inversely proportional to each other; higher the book price, lower the book score.

Scatter Plot between Book Price and Book Score



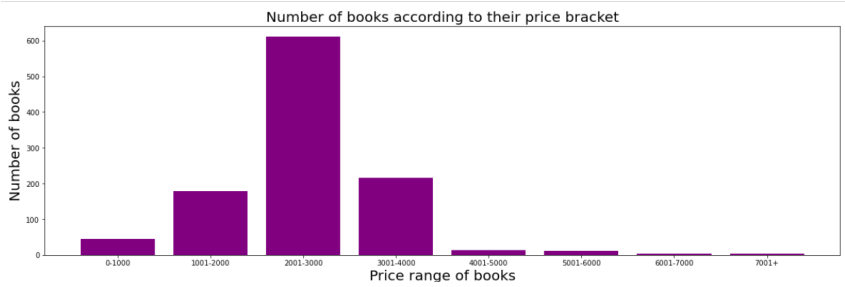
- The scatter plot between book prices and book score reflects that the books in the price range of Rs. 2000-3000 tend to have a higher book score compared to the other price ranges.

Scatter Plot between Number of years since book was launched and Book Score



- The above scatter plot between the number of years since the book is released and the book score reflects an inversely proportional relationship between the two entities.

Number of books according to their price bracket



- This bar plot shows that out of all the books available in the dataset, the most number of books have been priced in the range of Rs. 2000-3000 followed by the price bracket of Rs. 3000-4000 and then closely followed by Rs. 1000-2000 range.

EDA of books that have performed well amongst readers (with more than 10 reviews and ratings > 4.5)

Scatter plot between ratings and number of reviews of books with more than 10 reviews and ratings > 4.5

This scatter plot between ratings and number of reviews of books with ratings > 4.5 and reviewed more than 10 times reflects that -

- Most of the books with ratings at 4.6 have been reviewed about 90-100 times.
- Books that have been reviewed more than 50 times tend to have a rating lesser than those that have been reviewed less than 50 times.

Scatter plot between price and ratings of books with more than 10 reviews and ratings > 4.5

This scatter plot between price and ratings of books rated higher than 4.5 and reviewed more than 10 times shows that -

- The books priced between Rs. 2000-4000 have a rating significantly higher than the mean rating of the dataset(Mean : 4.125).
- It shows an almost normal distribution.

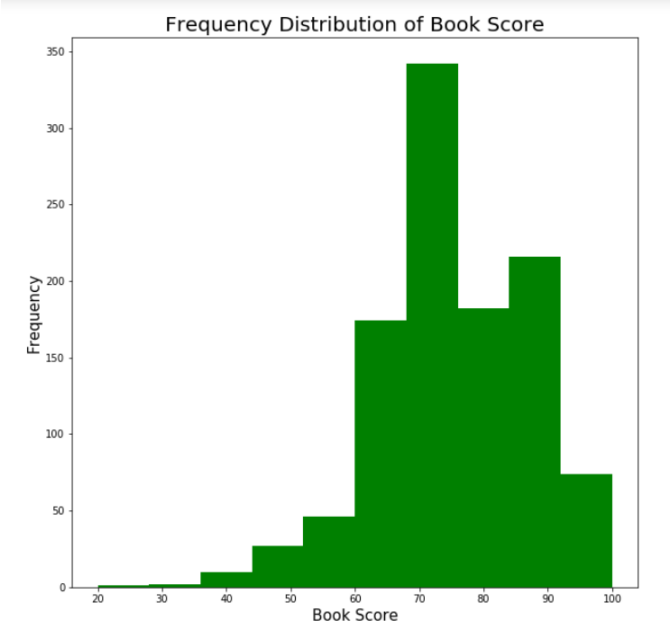
Scatter plot between price and number of reviews of books with more than 10 reviews and ratings > 4.5

- Books that are priced between Rs. 2000-4000 have been reviewed more compared to those priced outside of this price bracket.

Number of books according to their price bracket

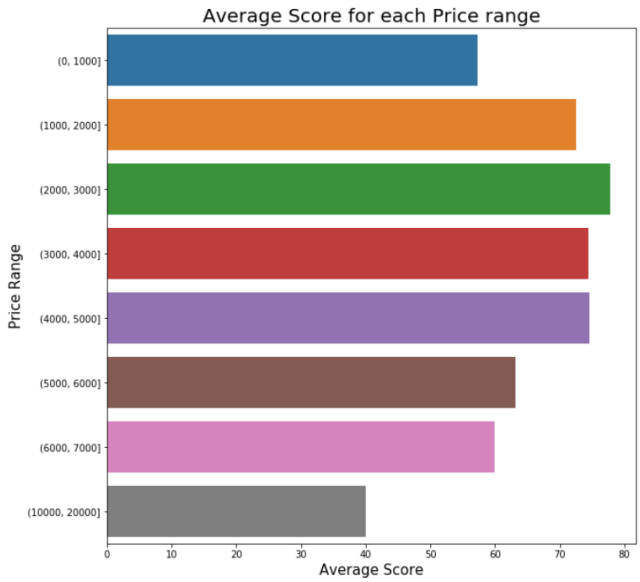
- The bar plot above shows that majority of books that have ratings > 4.5 and have been reviewed more than 10 times have been priced in the range of Rs. 2000-4000

Frequency Distribution of Book Score



- Most of the books have a score between 70-75, followed by 85-90 and 75-80, which is closely chased by 65-70.

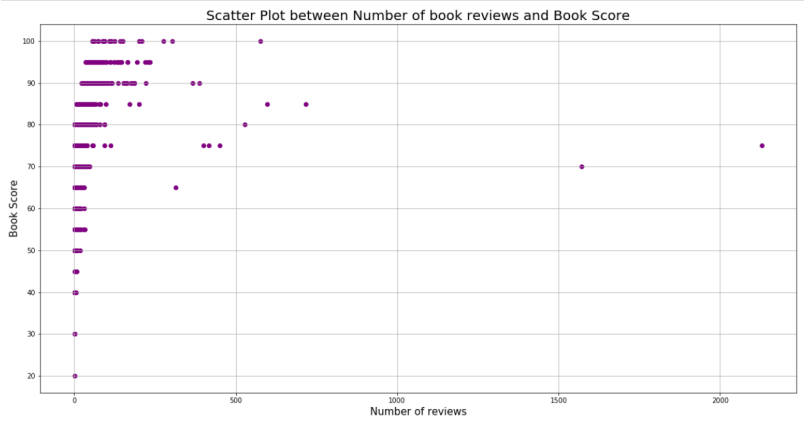
Average Score for each Price range

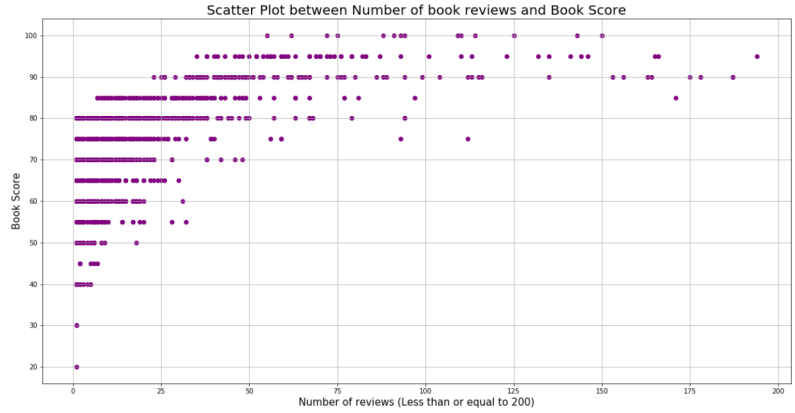


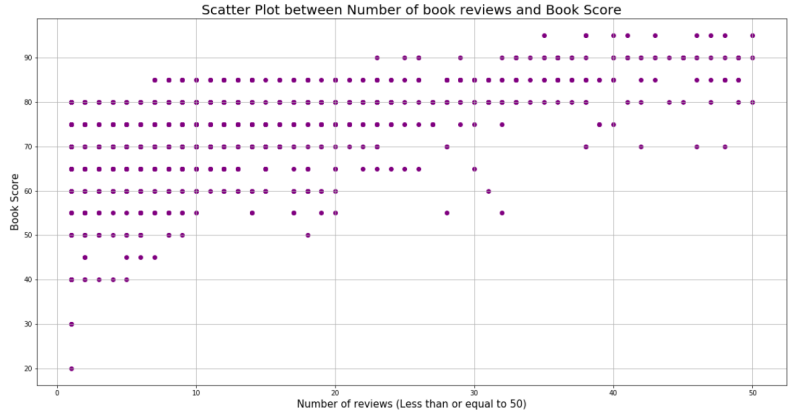
The average book score of price ranges is shown in the graph above.

- The average book score of books priced between Rs. 2000-3000 is higher than the rest of the price brackets.
- Books priced above Rs. 5000 tend to have low book scores.

Scatter Plot between Number of book reviews and Book Score







The three visualizations above (generally, less than 200 reviews and less than 50 reviews) reflect the fact that the books that have been reviewed more tend to have a higher book score.

Conclusions and Results

EDA of entire data

- Most people prefer reading books that are not too expensive.
- The books that are priced more than 5000 do not have many readers, thus not many reviews either.
- When a book has higher ratings, it is actually been reviewed by a significant number of people, and liked by them.
- The books with less reviews has less ratings, which reflects that people do not review a book when they don't like it.
- The ratings distribution is skewed to the right, which means most of the books have good ratings.
- The price distribution is skewed to the left, with most of the books priced between Rs, 1000-4000.
- Most books have been priced in the range of Rs. 2000-3000.

EDA of books performing well

- Mean Price - 2652.269565217391
- Median Price - 2466.0
- It reflects that most of the books falling under this filter are priced between Rs. 2000-3000.
- Books that have been reviewed more than 50 times tend to have a rating lesser than those that have been reviewed less than 50 times.
- Books priced between Rs. 2000-4000 have been reviewed more, and have significantly higher ratings compared to the mean of the entire dataset.
- Most books under this filter are priced between Rs. 2000-3000.

EDA involving book score

- Book ratings and book score are directly proportional to each other.
- Book price and book score are inversely proportional to each other.
- Books priced between Rs. 2000-3000 have a higher book score than the other books.
- There is no relationship between the book score and discount applicable on it.
- Book score is inversely proportional to the number of years since the book is released.
- Most of the books have a book score between 70-75.
- Books priced above Rs. 5000 have a less book scores, and thus, can be considered overprized.
- The number of reviews on a book is directly proportional to their respective book scores.

EDA involving keywords

- Python, React.js, Tensorflow, Java and Azure are trending topics amongst publications.
- Powerpoint, MySQL have the highest average ratings amongst the keywords, which means many publications are writing books about these topics.
- The average price of books about 'webflux' is the highest amongst all other categories

Extracting Keywords from Title:

Since the amount of features we could extract from web scraping were limited. It was imperative to find features that could prove useful. The core idea was to find the overall topic of the book and at the same time divide the books into segments which would facilitate further analysis on these segments.

This is achieved using natural language processing Tool known as KeyBERT (https://github.com/MaartenGr/KeyBERT) , This helps to extract keyword from a given text. KeyBERT is a minimal and easy-to-use keyword extraction technique that leverages BERT embeddings to create keywords and keyphrases that are most similar to a document.
First, document embeddings are extracted with BERT to get a document-level representation. Then, word embeddings are extracted for N-gram words/phrases. Finally, we use cosine similarity to find the words/phrases that are the most similar to the document. The most similar words could then be identified as the words that best describe the entire document.
Keywords with one and two entities were extracted, but finally we chose to go with the single keyword as it had more instances of a given category

keyword	keyword2
['python object oriented']	['python']
['learning rstudio statistical']	['rstudio']
['real time web']	['socket']
['robot framework test']	['automation']
['digital painting essentials']	['sketchbook']
['node js']	['node js']
['runescape gold strategy']	['runescape']
['data science python']	['jupyter']
['devops docker']	['docker']
['symfony web application']	['symfony']
['enterprises lambda architecture']	['lambda']

Title performance

The first thing we notice while searching for a book is the title of the book, so the title has a lot of impact on the book, and on its sales. The analysis of the title is also a crucial aspect along with others, while analyzing the books. In this model, a function is created to estimate the range of ratings a book would get based on the title, and the authors of the book. A classifier is trained to predict the range of rating a title could get by taking text input of the title, and author together. An algorithm is developed using the technique of vectorization of tokens in the text, and machine learning algorithms together. The output of this algorithm is a single integer, which suggests the open interval rating range between the previous, and output integer. For instance, if the output is 3 then the (2,3] is the rating range the function is expected for the title of the book.

```
pipe_nb.score(test_text,test_class)
```

```
0.6850393700787402
```

Estimating the score of the book

The score of the book is one of the features, which can give the viewer an overall insight about the performance since the release of the book. The score of the book is calculated by analyzing different features like the price, rating, number of reviews since the launch of the book. Score is an important feature to understand the performance of the book than other metrics of the book, because other features can give misinterpretation about the book. For example, two books of the same concept from the same publisher have ratings of 4.7 and 4.2, but the first one has only 10 overall ratings and the second one has 150 ratings which make the second book with 4.2 rating more reliable. Considering all these scenarios the score was used to judge overall achievement of the book. The function to predict the score of the book was created with aid of machine learning regression algorithms. The algorithm was trained with the several data points, where price, rating, number of ratings, and number of years since its release as independent variables, and score as target variable. Different algorithms like linear regression, decision tree regressor, random forest regressor, and SVM regressor were used in validation. And the score, the output is a float variable returned from the score predicting function varying between 0 and 100.

```
print('r2 socre is',r2_score(y_test,y3_prediction))
print('mean_absolute_error',mean_absolute_error(y3_prediction,y_test))
```

```
r2 socre is 0.9553721062886857
mean_absolute_error 1.0774691358024686
```

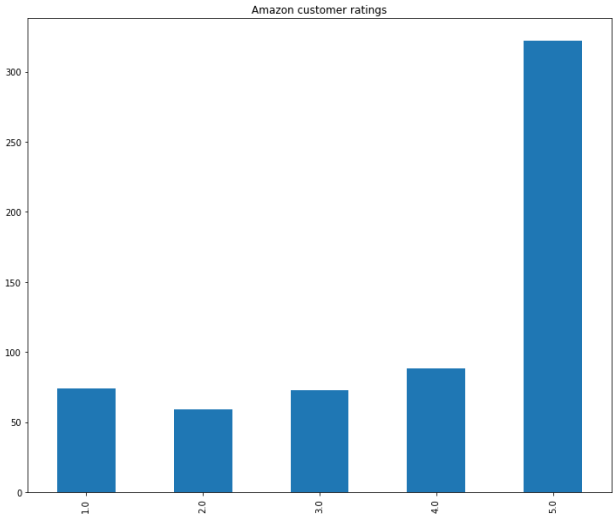
Packt product's amazon review analysis:

Parse hub was used to scrape packt product review data from 15 consecutive amazon pages following this page : https://www.amazon.in/s?k=packt&i=stripbooks&rh=n%3A976389031%2Cp_n_binding_browse-bin%3A1318375031%7C1318376031%7C1318377031%7C1634951031&dc&crid=N53G9MJAW569&qid=1664807339&rnid=1318374031&sprefix=packt+%2Caps%2C301&ref=sr_pg_1

From every product page the top reviews, that is the reviews with the maximum 'is helpful' rating were selected. This led us to a total of 604 reviews and a total of 63 books. The data extracted consists of the following fields

	book_name	book_reviewer_name	book_reviewer_rating	book_reviewer_title	book_reviewer_info	book_reviewer_reviewText
0	Getting Started with Google BERT: Build and tr...	Mohd Azam	5.0	Highly recommended	Reviewed in India IN on 25 May 2022	This is best book for advance NLP tasks. BERT ...
1	Getting Started with Google BERT: Build and tr...	MRS IYENGAR	5.0	A pitch-perfect book in regards to getting sta...	Reviewed in India IN on 7 March 2021	The amount of in-depth information compiled ab...
2	Getting Started with Google BERT: Build and tr...	Chandrakant Kantilal Bhogayata	4.0	Transformer revolution in Natural Language Pro...	Reviewed in India IN on 16 April 2021	I like the excellent explanation of the BERT m...

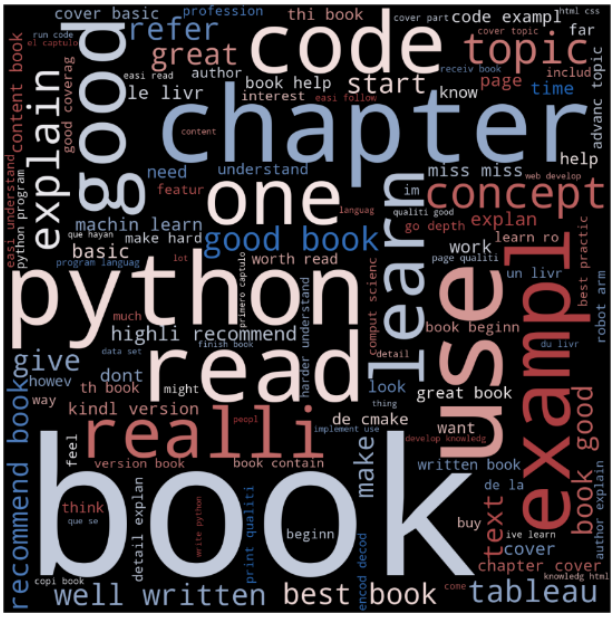
The frequency of the each individual rating in amazon reviews is represented in the bar plot below:



For further analysis a cleaned review text was created to find most commonly used words in the reviews. To clean the text we remove all the stopwords present, removing punctuations, converting to lowercase and also stemming. This data is referred to as cleaned reviews

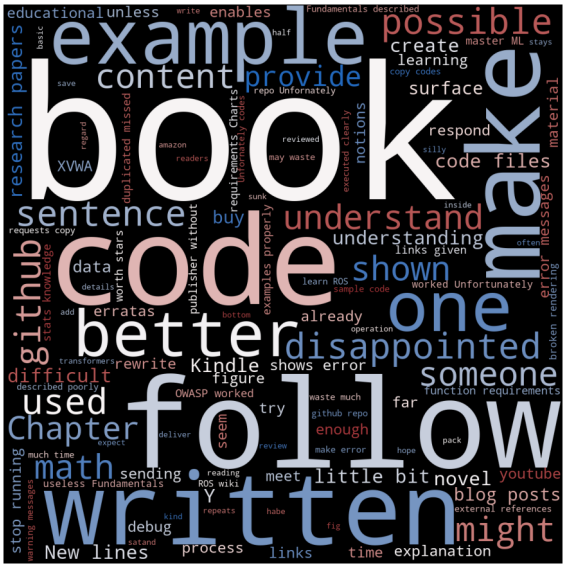
Word Cloud:

Below is a word cloud of the most frequent words used in the reviews:



Wordcloud - all reviews

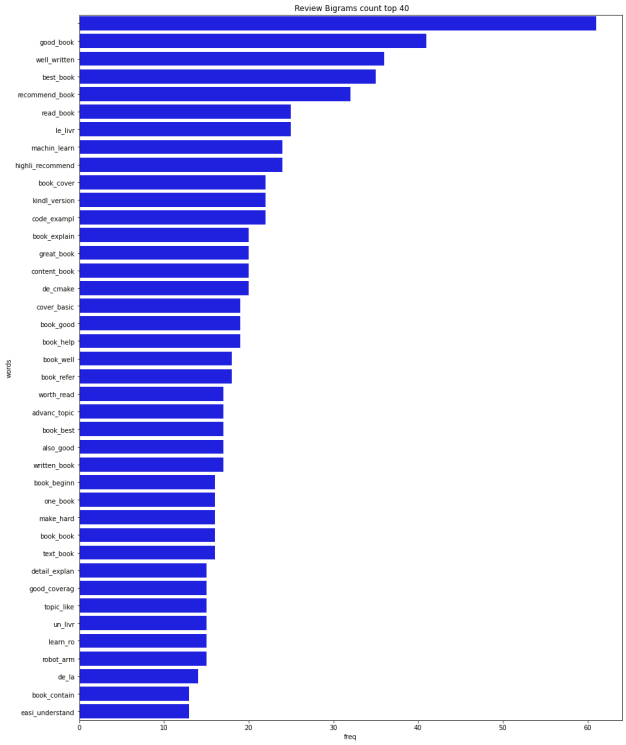
reviews with rating = 5



Reviews with rating = 1

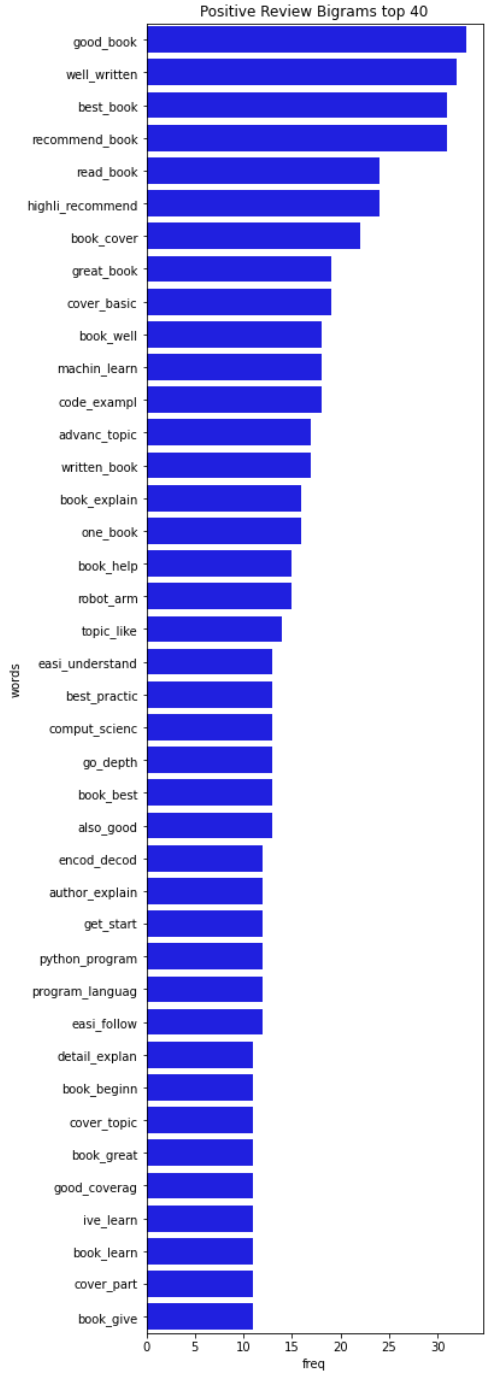
Bigrams and trigrams :

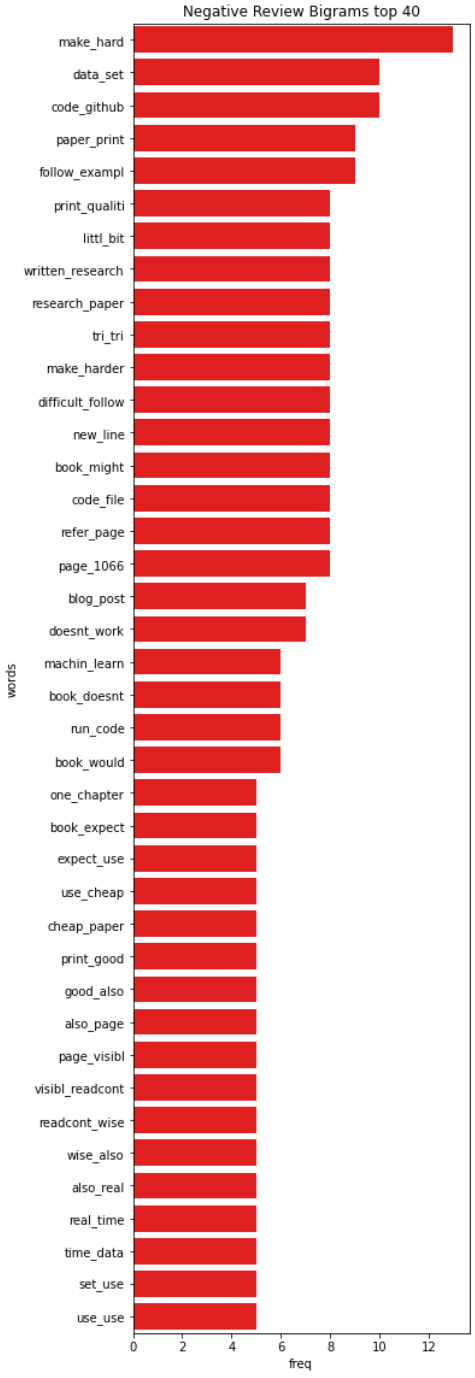
Below is the frequency bar plot that depicts the top 40 bigrams in the all cleaned reviews. That represents the frequently used phrases from the reviews. ‘Good book’, ‘best book ’ and ‘well written ’ are amongst the top three most used bigrams.



Comparing bigrams of positive and negative reviews:

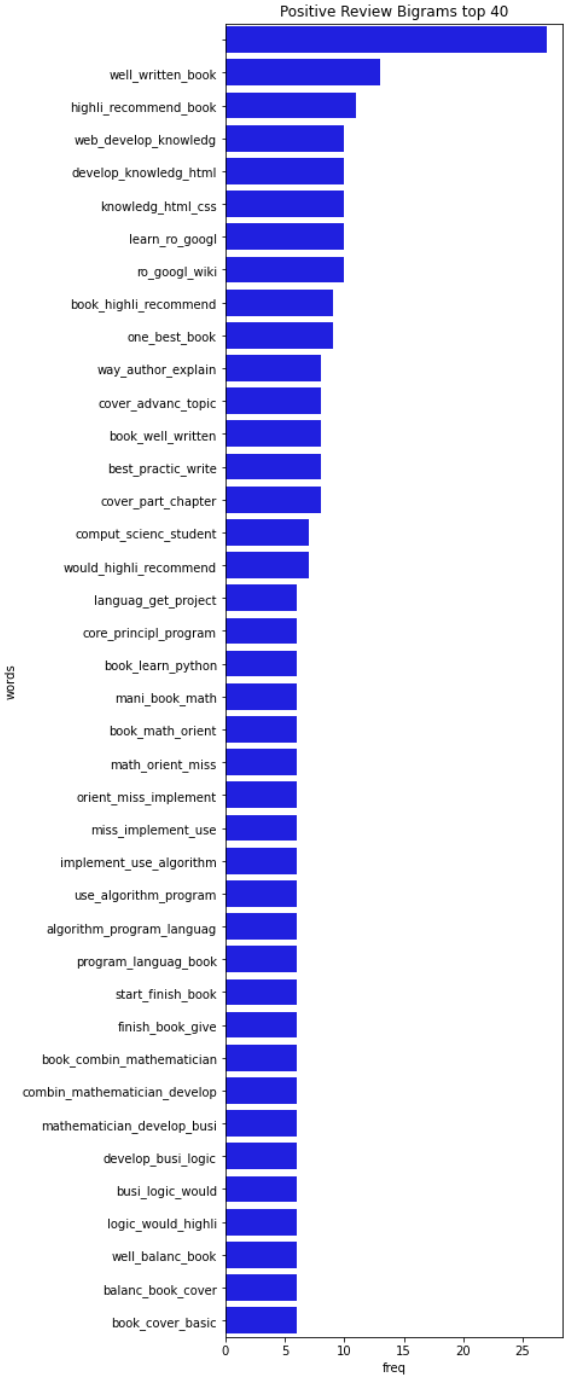
In this context positive reviews refer to the reviews that consist of a rating greater than 3 and negative reviews have a rating of less than 3. The most frequently used bigrams are contrasted

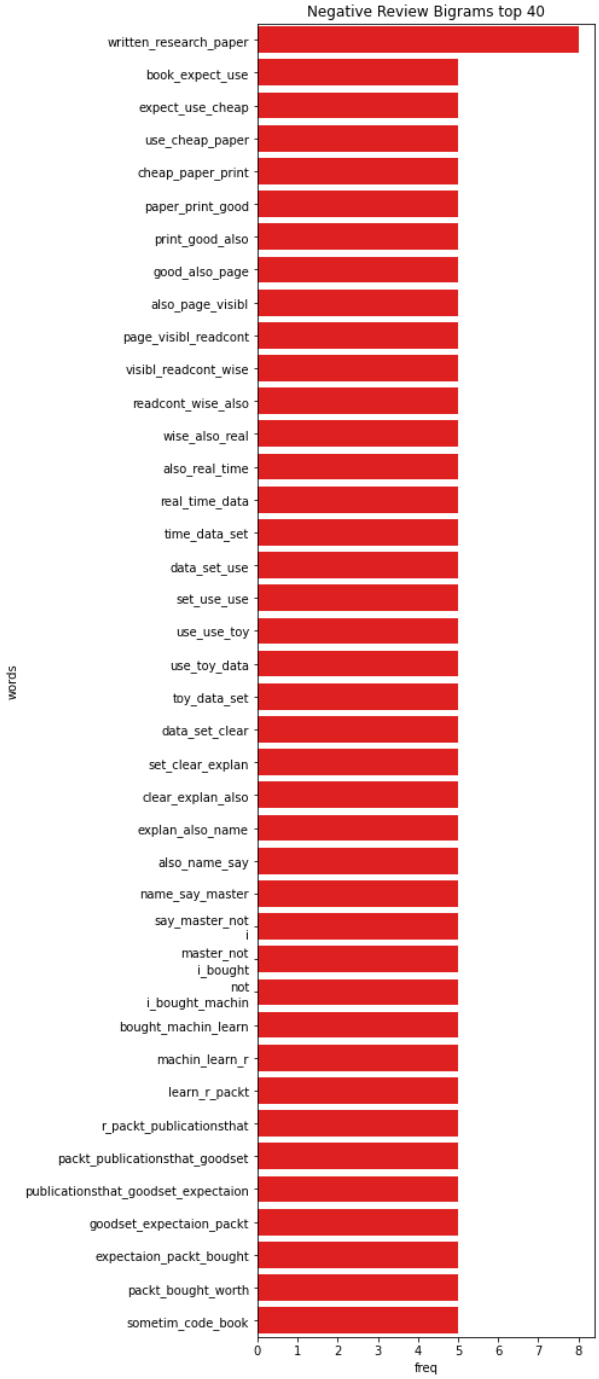




Comparing trigrams of positive and negative reviews:

In a similar fashion as before the most frequently used trigrams are contrasted





Editorial Feedback :

Performing Intent classification and sentiment analysis :

A preliminary proof of concept

Intent classification: Intent classification is the process of classifying the customer's intent by analyzing the language they use. This method was designed for chat bots to understand the intent of the user. Thus it can be further used to relay this information and generate a response This comprises use of a natural language understanding. However we can make use of the classifying method to help extract feedback in a more structured manner. We can use these intent classes as segments to subset the textual information in the review.

sentiment analysis: Sentiment analysis, also referred to as opinion mining, is an approach to natural language processing (NLP) that identifies the emotional tone behind a body of text. This is a popular way for organizations to determine and categorize opinions about a product, service, or idea. The output from this can also be used as segments of overall data to filter and analyze in detail.

The classical approach is to apply these methods to the entire review. **However**, we chose to first **sentence tokenize** the reviews using NLTK sentence tokenizer. This will leave us with multiple sentences belonging to the same review text. So that we can now **segregate further** on the basis of the intent classes and sentiments.
The motivation behind the decision is to extract individual praise and criticism and different intent classes from the same review text so that they can be further evaluated and also used for statistics.

Intent classification method : Time being a huge factor in our decision making we chose to go we a pretrained model from hugging face :
bespin-global/klue-roberta-small-3i4k-intent-classification
- Dataset for fine-tuning : 3i4k
- Train : 46,863
- Validation : 8,271 (15% of Train)
- Test : 6,121

The following intent classes were used to divide the sentences into segments for further analysis :
Label info:
- 0: "fragment"
- 1: "statement"
- 2: "question"
- 3: "command"
- 4: "rhetorical question"
- 5: "rhetorical command"
- 6: "intonation-dependent utterance"

Considering the time and resource constraint this model seemed like a good option to prove the concept. However, better models can be trained by fine tuning the data, training on a custom dataset that better represents our use case with better fitting clases.

The model provides the confidence score to show the confidence on the prediction :

	label	score
0	fragment	0.000248
1	statement	0.985341
2	question	0.000511
3	command	0.002226
4	rhetorical question	0.009262
5	rhetorical command	0.002199
6	intonation-dependent utterance	0.000213

Sentiment analysis method : Based on the constraints mentioned before, a similar approach was adopted leading to the selection of a pretrained model for sentiment analysis from hugging face:

SiEBERT - English-Language Sentiment Classification

This model ("SiEBERT", prefix for "Sentiment in English") is a fine-tuned checkpoint of RoBERTa-large (Liu et al. 2019). It enables reliable binary sentiment analysis for various types of English-language text. For each instance, it predicts either positive (1) or negative (0) sentiment. The model was fine-tuned and evaluated on 15 data sets from diverse text sources to enhance generalization across different types of texts (reviews, tweets, etc.).

```
text = "It covers many things but none in detail except a few. Page quality is good, write style is great.But highly overpriced and disappointed."
sentiment_analysis(text)
```

```
[{'label': 'NEGATIVE', 'score': 0.9994895458221436}]
```

This model is used in tandem with the intent classifier :

```
text = "It covers many things but none in detail except a few. Page quality is good, write style is great.But highly overpriced and disappointed."
clas, confc, sentiment0, confs = classify_sentence(text)
print(clas, confc, sentiment0, confs)
```

```
statement 0.9853407740592957 NEGATIVE 0.9994895458221436
```

Examples from the amazon review dataset itself can be used to further illustrate the use of both of these classification :

```
text = "is this book for beginners..?"
clas, confc, sentiment0, confs = classify_sentence(text)
print(clas, confc, sentiment0, confs)
```

```
rhetorical question 0.48029017448425293 NEGATIVE 0.9971852898597717
```

```
text = "good job"
clas, confc, sentiment0, confs = classify_sentence(text)
print(clas, confc, sentiment0, confs)
```

```
fragment 0.9804899096488953 POSITIVE 0.9987636804580688
```

The model understands the use of rhetoric to emphasize points used by the reviewer and also that the implications of the statement are negative.

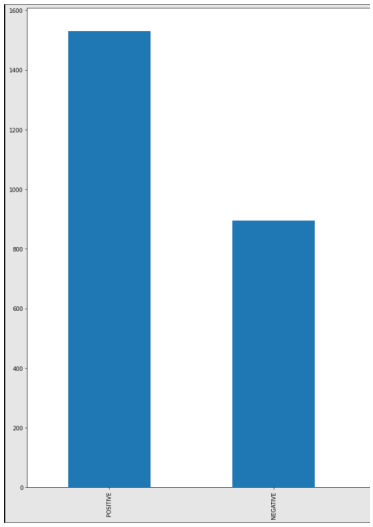
```
text = "A Perfect treat to AWS Cloud Lovers!"
clas, confc, sentiment0, confs = classify_sentence(text)
print(clas, confc, sentiment0, confs)
```

```
statement 0.9897257685661316 POSITIVE 0.9988704323768616
```

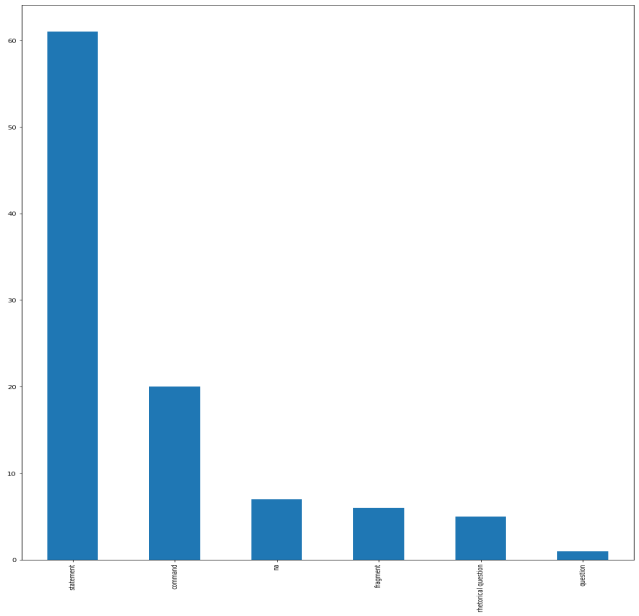
These two classifications were used in tandem and the classes and confidence values are attached to the dataset after this processing the dataset looks like this :

	book_name	book_reviewer_name	book_reviewer_rating	book_reviewer_title	book_reviewer_info	book_reviewer_reviewText	sentences	clas	confc	sentiment0	confs
84	Mastering SaltStack -: Use Salt to the fullest	Amazon Customer	3.0	As expected	Reviewed in the United States us on 1 November...	As expected	As expected	fragment	0.547011	POSITIVE	0.997447
117	LEARNING AWS 2/ED: Design, build, and deploy r...	Sarat Chandra Dash	5.0	Good one	Reviewed in India IN on 9 May 2019	Good one	Good one	fragment	0.896900	POSITIVE	0.998824
144	Programming with CodeIgniterMVC	Amazon Customer	1.0	One Star	Reviewed in India IN on 29 October 2017	Useless	Useless	fragment	0.995300	NEGATIVE	0.999297

The following is the frequency count of positive and negative reviews:



The following is the frequency count of individual intent class for the first hundred :



Filtering involves subsetting the data to find valuable

```
packt_reviews_sent[packt_reviews_sent.book_reviewer_rating == 4][packt_reviews_sent.sentiment0 == 'NEGATIVE'].iloc[10].sentences
```

With that said, so far, the biggest challenge getting through this book as a beginner has been keeping myself motivated when the reading makes such drastic shifts from difficult but approachable topics, to suddenly very dense and mathematical concepts.

This is an example of such extracted text. The proposed method does involve a bit of human involvement but the intention is to make sifting through tons of review easier for the handler. And a bunch of analytics can also be performed on the data like no. of questions or rhetoric questions that are used to express criticism. To get further insight.

The idea is for the analyser to focus on a subset of the reviews. filtering according to the requirement and focusing on those aspects.

sentences	clas	confc	sentiment0	confs	book_reviewer_rating
This is best book for advance NLP tasks.	statement	0.97262585	POSITIVE	0.99878424	5
BERT is not more complex â€¦ recommended for NLP developers	statement	0.71662933	POSITIVE	0.99847919	5
The amount of in-depth information compiled about BERT and its varients in this book is quite astonishing	statement	0.95994586	POSITIVE	0.99877983	5
The content is top-notch with great visual explanations.	statement	0.99441999	POSITIVE	0.99891734	5
Encoders and decoders are very hard to explain.	statement	0.91956586	NEGATIVE	0.99947506	5
In this book encoder and decoder chapters are peaches, they are explained with absolute perfection.	statement	0.98429894	POSITIVE	0.99887997	5
The range of BERT variants in this book is mind-blowing, the book contains BERT, ALBERT, RoBERTa, Dist	statement	0.77928525	POSITIVE	0.99806827	5
My favorite BERT models are XLM-R, Sentence BERT, and VideoBERT.	statement	0.98397034	POSITIVE	0.99828136	5
The author explains each and every concept with ease.	statement	0.98373002	POSITIVE	0.99890399	5
I personally felt very happy to have purchased and read the book.	statement	0.97866493	POSITIVE	0.9989202	5
I thank the author for writing such a gem of a book.	statement	0.94334567	POSITIVE	0.99888486	5
If you looking for a book to strengthen your knowledge of NLP, Deep Learning, BERT this is the go-to book	statement	0.62426895	POSITIVE	0.9988116	5

Above are all the sentences from a review with rating 5. That makes it an excellent rating on the reviewers end however. And if we perform overall sentiment analysis. It will return positive value however, A positive reviewer might have some constructive criticism. This can be extracted by filter for reviews with high rating and a negative sentiment attached to the sentences

Doing the same on the above review we can extract:
Encoders and decoders are very hard to explain. (constructive criticism from the above example)

Grow up dude.	rhetorical command	0.97596663	NEGATIVE	0.99849045	2
The book was good but the packaging was terrible.	statement	0.97707659	NEGATIVE	0.99887198	2
Maybe that's the last time I use your platform?	rhetorical question	0.98017085	NEGATIVE	0.99851936	2

Grow up dude.
The book was good but the packaging was terrible. Maybe that's the last time I use your platform?

This way the intent classification can also be used to filter reviews.