NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

CZ 4034

Information Retrieval Assignment

Harry Potter and the Cursed Child Book

Amazon Reviews

Tutorial Group: CS4

Team Members:
Wilson Neo Yuan Wei (U1721538L)
Ng Chen Ee Kenneth (U1721316F)
Atluri Sai Mona (U1722267D)

Table of Contents

1 Introduction

J.K Rowling is one of the most popular writers of the present generation famously known for her Harry Potter series which was first published in 1997 and soon gained massive popularity and was expanded into a series of seven books and earned million in profit. The books' massive popularity also led to the eventual making of the Harry Potter film series that was based on the books. Although the first seven books and their respective films are a huge success J.K Rowling received massive criticism on her recent launch "Harry Potter and the cursed child".

Many members of the Harry Potter community were unhappy with the book and some even felt that the series would be better off without it. This project aims to help Harry Potter fans analyze what went wrong with J.K Rowling's "Harry Potter and the Cursed Child" by helping them to analyze the opinions of other readers and find aspects of the book that they agree or disagree with. The analysis also helps have a more detailed idea of the overall response readers have towards the book.

The analysis is based on the reviews of the books found on Amazon. The book has around 12,000 reviews and an overall rating of 3.9 which is relatively low compared to other books in the series which have a rating of at least 4.5. **An overview of the classification process can be found in the Appendix of the report.**

2 Question 1 - Crawling

Our group built our information retrieval system based on the customer reviews of the book, Harry Potter and the Cursed Child Part 1 and 2, on Amazon. We crawled for the available reviews from Amazon and performed sentiment analysis on the data collected to analyze how well-liked the book is.

2.1 How you crawled the corpus and stored them?

To acquire the data required for our analysis, we crawled the corpus, which is the Amazon review page of the book, Harry Potter and the Cursed Child Part 1 and 2. (Refer to Section 7: Video, Data and Source Codes links for the URL to the webpage). Our group used an Amazon Review Scraper, which uses a web scraper extension of the Google Chrome Internet Browser to crawl the designated corpus. In order to use this tool, the extension is required to be installed on a Google Chrome browser whose version is required to be 49 or newer. The web scraper extension can be downloaded and installed at the Chrome Web Store. (Refer to Section 5 for the download link of the extension)

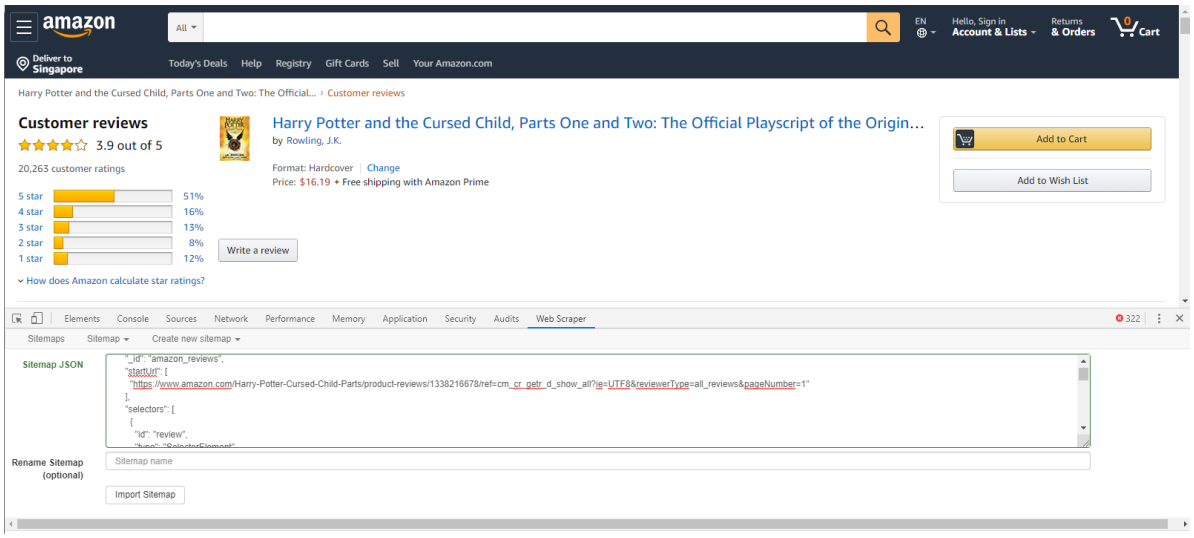
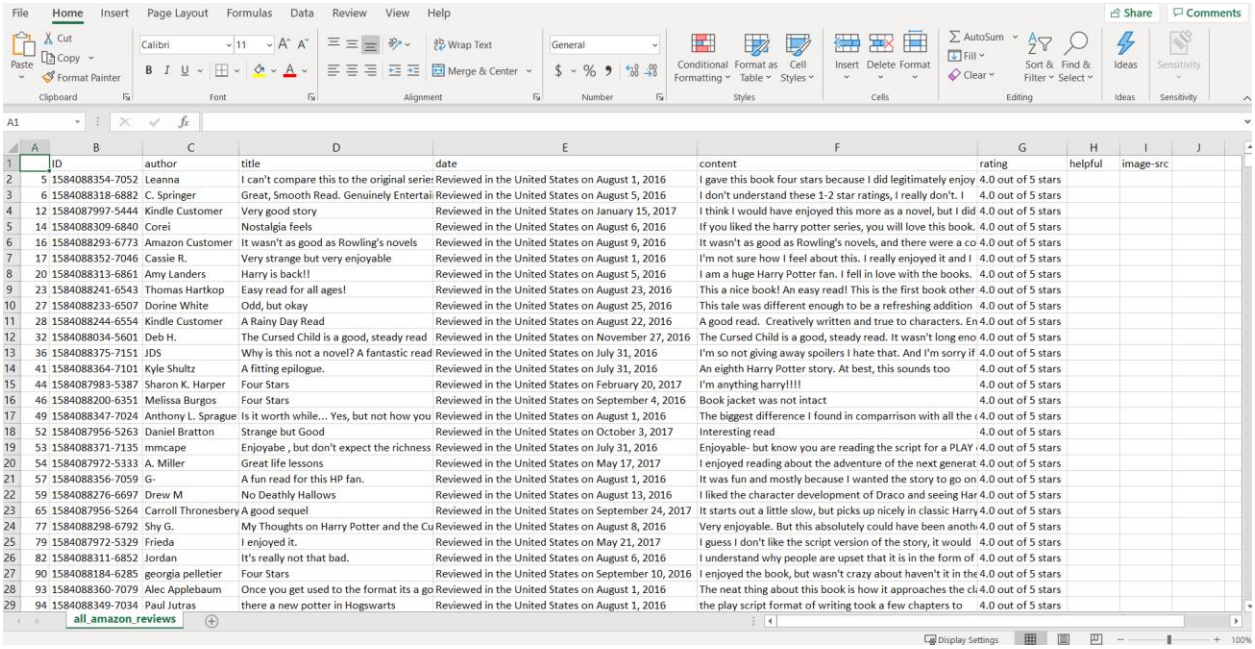


Figure 1: Amazon Review Scraper

Once the web scraper extension is downloaded and installed, we can create or import a sitemap that reveals how the website should be traversed and what data to be extracted as shown in Figure 1 above. In order to crawl information such as reviews from Amazon webpage, we will use a sitemap that is for Amazon webpage. By editing the sitemap JSON file, we will be able to choose specific data that we want to extract, such as Author, Title, Rating and Review comments. We then import the sitemap JSON file into the web scraper extension and have the extension crawl the corpus for the data we want which will be elaborated in Section 2.2.



	ID	author	title	date	content	rating	helpful	image-src
2	5 1584088354-7052	Leanna	I can't compare this to the original series	Reviewed in the United States on August 1, 2016	I gave this book four stars because I did legitimately enjoy	4.0 out of 5 stars		
3	6 1584088318-6882	C. Springer	Great, Smooth Read. Genuinely Entertai	Reviewed in the United States on August 5, 2016	I don't understand these 1-2 star ratings, I really don't. I	4.0 out of 5 stars		
4	12 1584087997-5444	Kindle Customer	Very good story	Reviewed in the United States on January 15, 2017	I think I would have enjoyed this more as a novel, but I did	4.0 out of 5 stars		
5	14 1584088309-6840	Corei	Nostalgia feels	Reviewed in the United States on August 6, 2016	If you liked the harry potter series, you will love this book.	4.0 out of 5 stars		
6	16 1584088293-6773	Amazon Customer	It wasn't as good as Rowling's novels	Reviewed in the United States on August 9, 2016	It wasn't as good as Rowling's novels, and there were a co	4.0 out of 5 stars		
7	17 1584088352-7046	Cassie R.	Very strange but very enjoyable	Reviewed in the United States on August 1, 2016	I'm not sure how I feel about this. I really enjoyed it and I	4.0 out of 5 stars		
8	20 1584088313-6861	Amy Landers	Harry is back!!	Reviewed in the United States on August 5, 2016	I am a huge Harry Potter fan. I fell in love with the books.	4.0 out of 5 stars		
9	23 1584088241-6543	Thomas Hartkop	Easy read for all ages!	Reviewed in the United States on August 23, 2016	This a nice book! An easy read! This is the first book other	4.0 out of 5 stars		
10	27 1584088233-6507	Dorine White	Odd, but okay	Reviewed in the United States on August 25, 2016	This tale was different enough to be a refreshing addition	4.0 out of 5 stars		
11	28 1584088244-6554	Kindle Customer	A Rainy Day Read	Reviewed in the United States on August 22, 2016	A good read. Creatively written and true to characters. En	4.0 out of 5 stars		
12	32 1584088034-5601	Deb H.	The Cursed Child is a good, steady read	Reviewed in the United States on November 27, 2016	The Cursed Child is a good, steady read. It wasn't long eno	4.0 out of 5 stars		
13	36 1584088375-7151	JDS	Why is this not a novel? A fantastic read	Reviewed in the United States on July 31, 2016	I'm so not giving away spoilers I hate that. And I'm sorry if	4.0 out of 5 stars		
14	41 1584088364-7101	Kyle Shultz	A fitting epilogue.	Reviewed in the United States on July 31, 2016	An eighth Harry Potter story. At best, this sounds too	4.0 out of 5 stars		
15	44 1584087983-5387	Sharon K. Harper	Four Stars	Reviewed in the United States on February 20, 2017	I'm anything harry!!!!	4.0 out of 5 stars		
16	46 1584088200-6351	Melissa Burgos	Four Stars	Reviewed in the United States on September 4, 2016	Book jacket was not intact	4.0 out of 5 stars		
17	49 1584088347-7024	Anthony L. Sprague	Is it worth while... Yes, but not how you	Reviewed in the United States on August 1, 2016	The biggest difference I found in comparrison with all the	4.0 out of 5 stars		
18	52 1584087956-5263	Daniel Bratton	Strange but Good	Reviewed in the United States on October 3, 2017	Interesting read	4.0 out of 5 stars		
19	53 1584088371-7135	mmcape	Enjoyabe , but don't expect the richness	Reviewed in the United States on July 31, 2016	Enjoyable- but know you are reading the script for a PLAY	4.0 out of 5 stars		
20	54 1584087972-5333	A. Miller	Great life lessons	Reviewed in the United States on May 17, 2017	I enjoyed reading about the adventure of the next generat	4.0 out of 5 stars		
21	57 1584088356-7059	G-	A fun read for this HP fan.	Reviewed in the United States on August 1, 2016	It was fun and mostly because I wanted the story to go on	4.0 out of 5 stars		
22	59 1584088276-6697	Drew M	No Deathly Hallows	Reviewed in the United States on August 13, 2016	I liked the character development of Draco and seeing Har	4.0 out of 5 stars		
23	65 1584087956-5264	Carroll Thronesbery	A good sequel	Reviewed in the United States on September 24, 2017	It starts out a little slow, but picks up nicely in classic Harry	4.0 out of 5 stars		
24	77 1584088298-6792	Shy G.	My Thoughts on Harry Potter and the Cu	Reviewed in the United States on August 8, 2016	Very enjoyable. But this absolutely could have been anoth	4.0 out of 5 stars		
25	79 1584087972-5329	Frieda	I enjoyed it.	Reviewed in the United States on May 21, 2017	I guess I don't like the script version of the story, it would	4.0 out of 5 stars		
26	82 1584088311-6852	Jordan	It's really not that bad.	Reviewed in the United States on August 6, 2016	I understand why people are upset that it is in the form of	4.0 out of 5 stars		
27	90 1584088184-6285	georgia pelletier	Four Stars	Reviewed in the United States on September 10, 2016	I enjoyed the book, but wasn't crazy about haven't it in the	4.0 out of 5 stars		
28	93 1584088360-7079	Alec Applebaum	Once you get used to the format its a go	Reviewed in the United States on August 1, 2016	The neat thing about this book is how it approaches the cl	4.0 out of 5 stars		
29	94 1584088349-7034	Paul Jutras	there a new potter in Hogswarts	Reviewed in the United States on August 1, 2016	the play script format of writing took a few chapters to	4.0 out of 5 stars		

Figure 2: Crawled Data in CSV file

When the crawling is completed, we export the extracted data in the form of a CSV file. As shown in Figure 2, the CSV file can be viewed with Microsoft Excel, which will display the reviews in rows while each column represents a type of information. For our CSV data, we crawled for the author, title, date, content and rating given of the reviews. We also additionally crawled for the number of people who find any specific review helpful to help us in conducting our analysis.

2.2 What kind of information users might like to retrieve from your crawled corpus?

This information retrieval system is created keeping Harry Potter fans in mind and consists of the reviews of the book "Harry Potter and the cursed child." Hence the possible type of information that is expected to be retrieved by the user are:

- The total number of positive and negative reviews for the book
- The number of reviews containing a particular keyword (For example the word 'Voldemort' can be used to find out the number of reviews that mentioned the word Voldemort)
- Reviews that are spoiler-free (For readers who do not want the plot hole to be revealed)
- Reviews that mention kids (For parents planning to gift the book to their kids)
- Reviews that mention J.K Rowling and
- Reviews that describe the quality of the book

2.3 The numbers of records, words and types in the corpus

Figure 3: Amazon Review Filters

Despite having a total of 12,697 customer reviews, Amazon displays only up to 5000 reviews per filter. As such, we crawled through each filter (1 to 5 stars, positive, critical and image reviews) to maximize the number of reviews that can be crawled from the product site.

Total No. of Records	10175
Total No. of Words in the Corpus	533637
Total No. of Unique Words in the Corpus	18322

3 Question 2 & 3 - Indexing and Querying

Indexing is a data structure technique that allows users to quickly retrieve records from a database file. For the scope of this project, the software Solr which is developed by the Apache software foundation is mainly used for indexing and querying. Solr is a very powerful, flexible, mature technology, and it offers not only powerful full-text search capabilities but also autosuggestion, advanced filtering, geocoded search, highlighting in text, faceted search, and much more. The main component in Solr is the Lucene library, a full-text search library written in Java. Solr uses inverted index to optimize the search process. Inverted indexing helps optimize the search process by enabling the user to retrieve all the documents containing a particular term.

Solr version 8.5.0 is used for the indexing of .csv file. Since the data being processed includes reviews, all the collected data is in the form of text that is sorted into different columns of the .csv files. Hence when deciding on the field type, “text_general” is chosen for all the columns. Amazon review id is chosen to be the default id as it is unique for each id.

3.1 Build a simple Web interface for the search engine

A web interface is created using php in combination with html. The web interface is hosted on the Apache HTTP Server by XAMPP. Attempt.php is used to get data from solr and homepage_modified.html is used to display the webpage that enables the user to make queries. Homepage_modified.html and search.php are used together to display the results of the query.

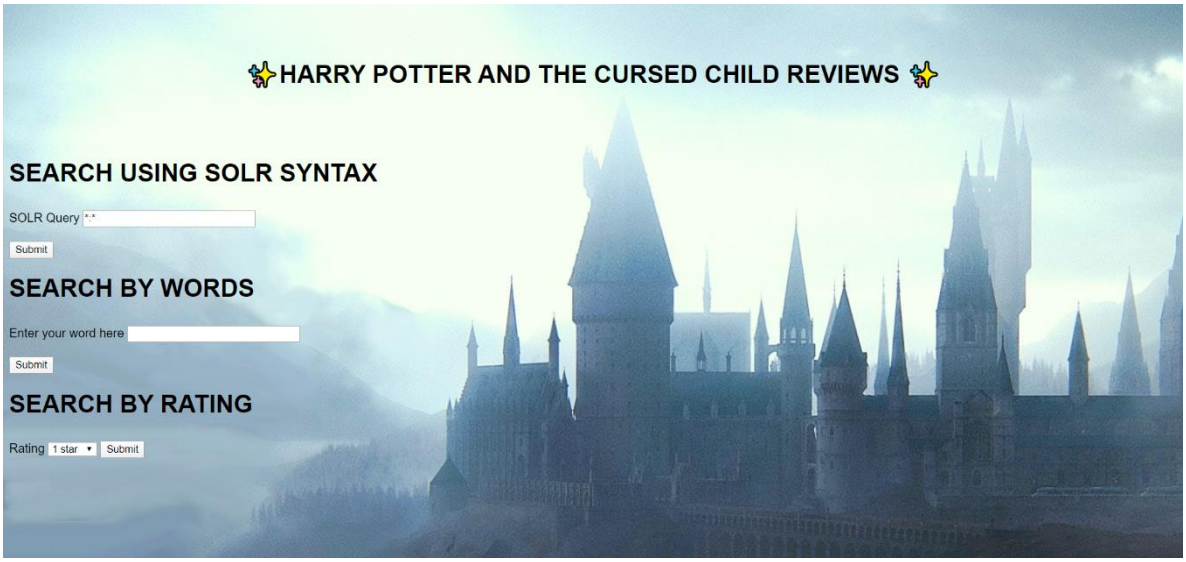


Figure 4: Search Engine Web Interface

In order to help ease the process of searching the user can search either by using specific syntax or by using words that they wish to find in the reviews. User can also choose to filter the existing reviews by their rating,

3.2 Write five queries, get their results and measure the speed of querying

For querying we decided to write five queries of five different types and measure the speed of querying.

Term query: A term query returns data that matches the specific term.

In this example term query is used to return all the reviews that have the phrase “harry” in their contents. The query used is:

```
content: harry.
```

Additionally, ‘Harry’ can also be entered in the search by words search bar to display the same reviews.

The resulting output as shown below is all the entries that have the term “harry” in the content field.

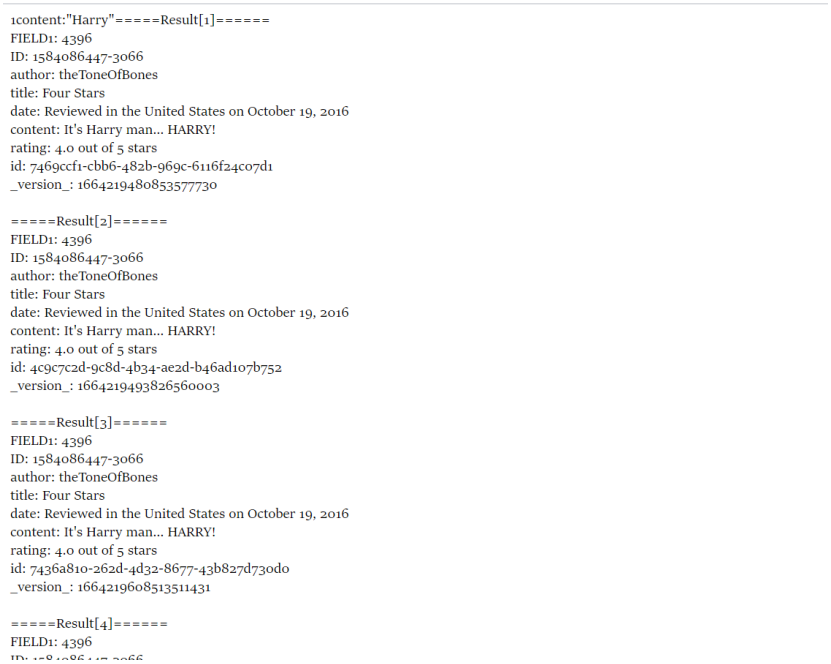


Figure 5: Term Query

Phrase query: Phrase query returns data that matches multiple terms in a sequence.

In this example phrase query is used to return all the reviews that have a rating of 3 stars. The query used is:

rating: "3.0 out of 5 stars"

Additionally, the user can also choose 3 stars in the drop-down menu in the search by rating

The resulting output as shown below is all the entries that have "3.0 out of 5 stars" in the field rating.

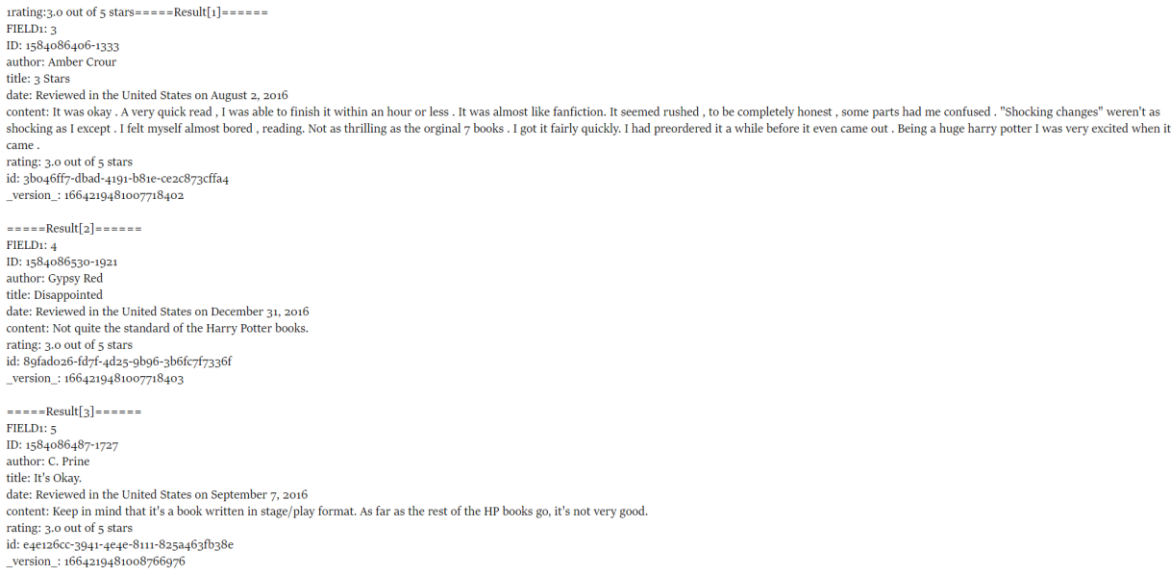


Figure 6: Phrase Query

Boolean query: A boolean query is used when a search contains multiple terms where each term can be either mandatory, or optional, or prohibited.AND, OR, and NOT are used to write a boolean query. Alternatively, +, - can also be used to indicate mandatory or prohibited respectively while no sign indicates optional.

In this example boolean query is used to return all the reviews that have a rating of 5 and that mentioned the term “Harry” or the term “Ron” in their review. The query used is:

(rating: 3.0 out of 5 stars) AND (content: Harry OR content: Ron)

The resulting output as shown below is all entries that have “3.0 out of 5 stars” in the field rating and at least one of the two terms “Harry” and “Ron” in the field content

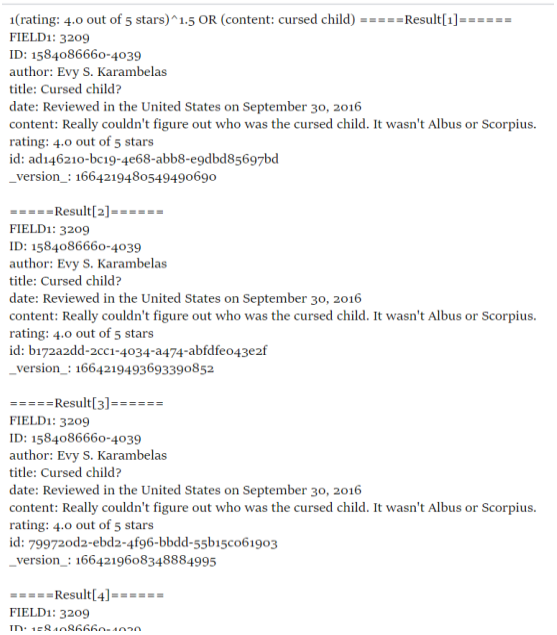
Figure 7: Boolean Query

Boost query: Boost query is used to specify which terms and clauses are more important. The importance is determined by the boost factor that is specified along with the query. The higher the boost factor, the more important is the term.

In this example, boost query is used to return all the reviews that have a rating of 4 or reviews that mentioned the term "cursed child" in their review. In order to give preference to reviews with rating 4, the boost factor for rating is chosen to be 2.0. The query used is:

(rating: 4.0 out of 5 stars)^1.5 OR (content: cursed)child)

The resulting output is shown below is all entries that have "4.0 out of 5 stars" in the field rating and the term cursed cursed child in the field content where the entries with a rating of 4 are given more importance.



1(rating: 4.0 out of 5 stars)^1.5 OR (content: cursed child) =====Result[1]======
FIELD1: 3209
ID: 1584086660-4039
author: Evy S. Karambelas
title: Cursed child?
date: Reviewed in the United States on September 30, 2016
content: Really couldn't figure out who was the cursed child. It wasn't Albus or Scorpius.
rating: 4.0 out of 5 stars
id: ad146210-bc19-4e68-abb8-e9dbd85697bd
version: 1664219480549490690

=====Result[2]======
FIELD1: 3209
ID: 1584086660-4039
author: Evy S. Karambelas
title: Cursed child?
date: Reviewed in the United States on September 30, 2016
content: Really couldn't figure out who was the cursed child. It wasn't Albus or Scorpius.
rating: 4.0 out of 5 stars
id: b172a2dd-2cc1-4034-a474-abfdfe043e2f
version: 1664219493693390852

=====Result[3]======
FIELD1: 3209
ID: 1584086660-4039
author: Evy S. Karambelas
title: Cursed child?
date: Reviewed in the United States on September 30, 2016
content: Really couldn't figure out who was the cursed child. It wasn't Albus or Scorpius.
rating: 4.0 out of 5 stars
id: 799720d2-ebd2-4f96-bbdd-55b15c061903
version: 1664219608348884995

=====Result[4]======
FIELD1: 3209

Figure 8: Boost Query

Proximity Query: Proximity query is used to find words that are in the specified proximity to each other. For example, in the word "Cup and cake" cup and cake are at a proximity of 1. ~x is used to define proximity where x is the proximity of the word.

In this example, proximity query is used to return all the reviews that have the terms "harry" and "child" at a proximity of 4. The query used is:

content: harry child~4

The resulting output as shown below is all entries that have the terms harry and child in the content field with a proximity of 4.

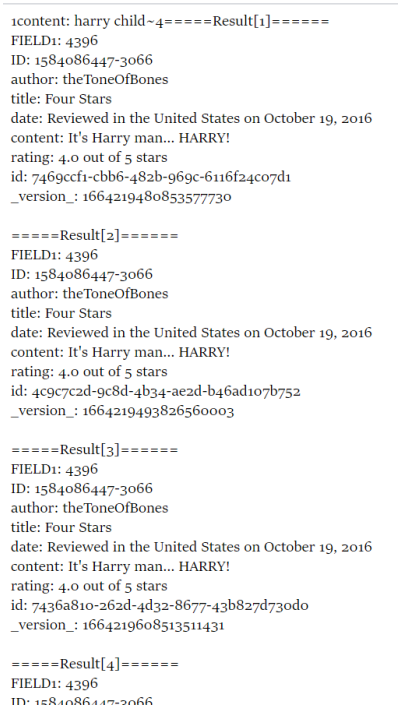


Figure 9: Proximity Query

Speed analysis for the queries:

Query	Speed of Querying (milliseconds)
content:harry	97
rating: “3.0 out of 5 stars”	18
rating: “3.0 out of 5 stars” AND (content: “Harry” OR content: “Ron”)	3
(rating: “4.0 out of 5 stars”)^1.5 OR (content: “cursed child”)	7
content: “harry child”~4	149

3.3 Innovations for enhancing indexing and ranking

As the data set for reviews is large it is important to make use of various enhancement techniques to ease the process of searching and querying. Some of the enhancement techniques that are useful in the context of this project are:

Spell checking and providing suggestions based on keywords

Using words that are spelled wrong during a search is one of the most encountered problems. In order to solve this problem, we can design the interface such that it would provide the correctly spelled versions of misspelled words in the form of suggestions.

This functionality can be implemented by using the 'suggest' component in Solr. This component provides users with automatic suggestions for query terms. The 'suggest' component makes use of Lucene's Suggester implementation and supports all of the lookup implementations available in Lucene. The "suggest" component function is activated by adding the search component solrconfig.xml.

```
<searchComponent name="suggest" class="solr.SuggestComponent">
  <lst name="suggester">
    <str name="name">mySuggester</str>
    <str name="lookupImpl">FuzzyLookupFactory</str>
    <str name="dictionaryImpl">DocumentDictionaryFactory</str>
    <str name="field">cat</str>
    <str name="weightField">price</str>
    <str name="suggestAnalyzerFieldType">string</str>
    <str name="buildOnStartup">false</str>
  </lst>
</searchComponent>
```

MoreLikeThis Search Component

Often a user who looks up for a particular query is also interested in other relevant queries. The *MoreLikeThis* search component enables users to query for documents similar to a document in their result list. The terms from the original document are used to find the relevant documents.

For the scope of this project, the more like this function is used as a request handler, that is similar results are displayed only when the user clicks on the link showing "similar results". The mlt query parser of Solr is used to implement this function and returns a list of results that exclude the results of the original query.

Parameter	Description
qf	Specifies the fields to use for similarity.
mintf	Specifies the Minimum Term Frequency, the frequency below which terms will be ignored in the source document.
mindf	Specifies the Minimum Document Frequency, the frequency at which words will be ignored when they do not occur in at least this many documents.
maxdf	Specifies the Maximum Document Frequency, the frequency at which words will be ignored when they occur in more than this many documents.
minwl	Sets the minimum word length below which words will be ignored.
maxwl	Sets the maximum word length above which words will be ignored.
maxqt	Sets the maximum number of query terms that will be included in any generated query.

maxntp	Sets the maximum number of tokens to parse in each example document field that is not stored with TermVector support.
boost	Specifies if the query will be boosted by the interesting term relevance. It can be either "true" or "false".

4 Question 4 - Classification

4.1 Motivate the choice of your classification approach in relation with the state of the art

The models we have chosen to use for classification are Naïve Bayes Classification, K-Nearest Neighbour Classification and Support Vector Machine Classification.

Naïve Bayes Classification

Naïve Bayes Classification is chosen as one of our models to conduct sentiment analysis on the Amazon reviews because using a Naïve Bayes classifier for the Amazon reviews sentiment analysis, which are textual data, will provide fruitful results. As the name of the classifier suggested, Naïve Bayes classifier uses the Bayes Theorem, which works on the probability that something will occur, given that something else has already occurred. Hence, it is very useful to help us in our sentiment analysis by identifying and classifying the Amazon customers' sentiment and emotions towards the Harry Potter and the Cursed Child book based on their reviews.

K-Nearest Neighbour Classification

The next classification model used in our project is the K-Nearest Neighbour Classification. This classification model classifies via the use of the K-Nearest Neighbour (KNN) algorithm, which is an algorithm that stores all cases and classifies them based on a similarity measure. For this KNN algorithm, to classify based on similarity is also to classify based on the distance functions between the available cases. Some examples of distance functions used are the Manhattan and Euclidean distance functions. As our crawled data are text data which are reviews by customers, there will be similarities in the words used to express their positive or negative sentiment. Therefore, we can use this classification model to calculate the similarity scores among the reviews' text data, identify the K-Nearest neighbours and classify them as positive or negative sentiment.

Support Vector Machine Classification

The third classification model we used is Support Vector Machine (SVM) Classification. This classifier is also another type of classifier that works well on our sentiment analysis as SVM works by drawing a hyperplane to classify our data. Since our sentiment analysis is on whether the Amazon customers' sentiment is positive or negative based on the Amazon reviews of the Harry Potter book, it is a linearly separable problem and hence be suitable to use a linear SVM to conduct sentiment analysis.

Decision Tree Classification

The last classification model we used is Decision Tree Classification. Decision Tree classification is one of the most common types of classification models. It is a supervised machine learning where data is continuously split based on a certain parameter or condition. Since we are analyzing the Amazon customers' reviews, the outcome will be a variable that is either positive or negative. This variable is discrete and hence, using the Decision Tree classification model would be ideal in our sentiment analysis.

4.2 Discuss whether you had to preprocess data and why

Prior to using the acquired data for classification and analysis, we must preprocess the data as there are various irregularities with the acquired data. Irregularities such as duplicated data or empty field data cells can affect our analysis, causing them to be biased unintentionally. Therefore, we must proceed to clean and process the acquired data before conducting the classification and analysis.

4.2.1 Merging CSVs

Firstly, we will import the CSV files, which contain our crawled data. As we have multiple CSV files, which each containing different filters of the Amazon website's review page, we will have to check through each CSV file to understand the structure of the data. For example, there are multiple filters for the ratings of the reviews, resulting in multiple CSV files for the different rating values. Once done, we proceed to merge the CSV files and any related labels into a single CSV file.

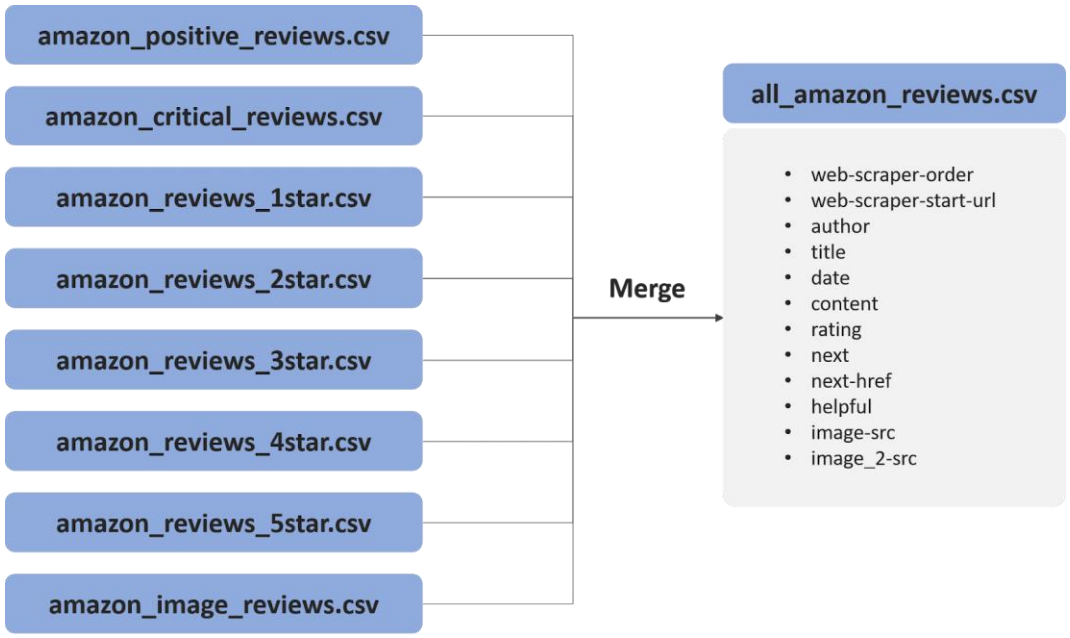


Figure 10: Merging CSVs

```
all_reviews_df.head()
```

	web-scraper-order	web-scraper-start-url	author	title	date	content	rating	next	next-href	helpful	image-src	im
0	1584086554-94	https://www.amazon.com/Harry-Potter-Cursed-Chi...	Scoe	some people might like it	Reviewed in the United States on April 20, 2018	I just could not get into this book. Too unha...	1.0 out of 5 stars	Next page→	https://www.amazon.com/Harry-Potter-Cursed-Chi...	NaN	NaN	
1	1584086949-1778	https://www.amazon.com/Harry-Potter-Cursed-Chi...	Amber Dews	Don't buy unless you want a play script!	Reviewed in the United States on August 1, 2016	If I could give zero stars...I would!!\nMy dau...	1.0 out of 5 stars	Next page→	https://www.amazon.com/Harry-Potter-Cursed-Chi...	NaN	NaN	
2	1584086610-324	https://www.amazon.com/Harry-Potter-Cursed-Chi...	jose garcia	One Star	Reviewed in the United States on January 17, 2017	I love harry potter but this book "the cursed ...	1.0 out of 5 stars	Next page→	https://www.amazon.com/Harry-Potter-Cursed-Chi...	NaN	NaN	
3	1584086676-600	https://www.amazon.com/Harry-Potter-Cursed-Chi...	Michael Warren	Cursed disappointment	Reviewed in the United States on October 7, 2016	Very poorly written a total insult to Rowlings...	1.0 out of 5 stars	Next page→	https://www.amazon.com/Harry-Potter-Cursed-Chi...	NaN	NaN	
4	1584086664-542	https://www.amazon.com/Harry-Potter-Cursed-Chi...	Allison Matthews	pathetic and sad	Reviewed in the United States on October 15, 2016	Characters are whiney, pathetic and sad. Not ...	1.0 out of 5 stars	Next page→	https://www.amazon.com/Harry-Potter-Cursed-Chi...	NaN	NaN	

Figure 11: Merged Data frame

4.2.2 Data Cleaning

Removal of irrelevant columns

After merging, we proceed to drop any columns that are redundant and irrelevant to our analysis. On closer inspection, we also realized that one of the columns is a unique ID for the web scraper to identify each crawled page and hence it is renamed to ‘ID’

Removal of duplicates

Next, we scan through the acquired data for data that are of exact duplicates from each other and remove them. We also scan through the data looking for any missing data or cells with NULL values to resolve such inconsistencies in the data.

Formatting columns

Next, the rating, helpful, date features are converted from a string to an integer or date format.

Feature	Before Cleaning	After Cleaning
rating	2.0 out of 5 stars	2
helpful	512 people found this helpful	512
date	Reviewed in the United States on August 5, 2016	2016-08-05

Sentiment Feature

The 4- and 5-star review ratings are grouped together as positive sentiment labelled as 1 while the 1- and 2-star review ratings are grouped as negative sentiment labelled as 0. The neutral rating of 3 is dropped from the data frame.

Case-folding, stripping of whitespaces and removal of empty strings

After cleaning the data, we proceed with formatting the content column. In the acquired data, the column called "Content" contains the reviews' comments by the author of the review. For us to conduct our analysis and classification without inconsistencies and errors, the next step will be to clean the text data within this "Content" column.

```
import re

def general_cleaning(x):
    pattern = '[^a-zA-Z0-9\ ]'
    x = str(x)
    x = re.sub(pattern,'',x)
    x = x.lower()
    x = x.strip()
    return x
```

```
comment_df = comment_df[comment_df['content'] != '']
comment_df['content'] = comment_df['content'].apply(general_cleaning)
comment_df.describe()
```

Figure 12: Review's Contents Data Preprocessing

We will first clean the content by performing case folding, removal of empty strings and removal of leading and trailing empty spaces. This will ensure that the text content will only consist of alphanumeric characters and the letters are entirely lowercase.

	ID	author	title	date	content	rating	helpful	image-src
0	1584088354-7052	Leanna	I can't compare this to the original series, b...	2016-08-01	I gave this book four stars because I did legi...	0	0	NaN
1	1584088318-6882	C. Springer	Great, Smooth Read. Genuinely Entertaining.	2016-08-05	I don't understand these 1-2 star ratings, I r...	0	0	NaN
2	1584087997-5444	Kindle Customer	Very good story	2017-01-15	I think I would have enjoyed this more as a no...	0	0	NaN
3	1584088309-6840	Corei	Nostalgia feels	2016-08-06	If you liked the harry potter series, you will...	0	0	NaN
4	1584088293-6773	Amazon Customer	It wasn't as good as Rowling's novels	2016-08-09	It wasn't as good as Rowling's novels, and the...	0	0	NaN

Figure 13: Result after preprocessing other columns

4.2.3 Stemming and Lemmatization

```
def word_pre(x):
    stemmer = PorterStemmer()
    x = word_tokenize(x)
    store = ''

    for i in x:
        store += stemmer.stem(i) + ' '

    return store
```

```
comment_df['content_stem'] = comment_df['content'].apply(word_pre)
comment_df.head()
```

Figure 14: Stemming

```
from nltk.stem import WordNetLemmatizer
def get_lemmatized_text(corpus):
    lemmatizer = WordNetLemmatizer()
    return [' '.join([lemmatizer.lemmatize(word) for word in review.split()]) for review in corpus]
```

```
comment_df['content_lem'] = get_lemmatized_text(comment_df['content'])
```

Figure 15: Lemmatization

```
from nltk.corpus import wordnet as wn
from nltk.stem.wordnet import WordNetLemmatizer
from nltk import word_tokenize, pos_tag
from collections import defaultdict
tag_map = defaultdict(lambda : wn.NOUN)
tag_map['J'] = wn.ADJ
tag_map['V'] = wn.VERB
tag_map['R'] = wn.ADV

def lemmatize_it(text):
  store = ''
  tokens = word_tokenize(text)
  lemma_function = WordNetLemmatizer()
  for token, tag in pos_tag(tokens):
    store += lemma_function.lemmatize(token, tag_map[tag[0]]) + ' '
  return str(store)
```

Figure 16: Lemmatization with POS tagging

Next, we proceed with stemming and lemmatization of the text contents to convert each word into its stem and lemma to assist us in our analysis and classification. We used the Wordnet lemmatizer from NLTK to group the different inflected forms of words. However, the default function in NLTK assumes that the words are nouns. As such, there is a need to represent adjectives, adverbs and verbs as part of speech (POS) tags before lemmatizing the words. In the Enhanced_Classification.ipynb file, we used figure 16's code snippet to lemmatize the document but it has proven to have limited effectiveness on the overall F1 score. Hence, it was not included in the Classification.ipynb file.

4.2.4 Removal of Stop Words

```
# Get english stopwords
stop = stopwords.words('english')
additional_stopwords = ["'s","...","'ve","``","''","'m",'--',"'ll","'d", "\n"]
stop = set(stop + additional_stopwords)
def remove_stop(x):
    x = word_tokenize(x)
    store = ''

    for i in x:
        if i not in stop:
            store += i + ' '

    return store
```

```
comment_df['content_stem_cleaned'] = comment_df['content_stem'].apply(remove_stop)
comment_df['content_cleaned'] = comment_df['content'].apply(remove_stop)
comment_df['content_lem_cleaned'] = comment_df['content_lem'].apply(remove_stop)
```

Figure 17: Removing Stop Words

We removed stop words to eliminate terms that are very common in the English language such as “and”, “this” and “the”. Additional stop words such as “’s ” and “’ve” were also removed.

	content	rating	sentiment	content_stem	content_lem	content_stem_cleaned	content_cleaned	content_lem_cleaned
9	a good read creatively written and true to characters enjoy reading this and watching it in your minds eye enjoy the rainy day	0	0	a good read creativ written and true to charact enjoy read thi and watch it in your mind eye enjoy the raini day	a good read creatively written and true to character enjoy reading this and watching it in your mind eye enjoy the rainy day	good read creativ written true charact enjoy read thi watch mind eye enjoy raini day	good read creatively written true characters enjoy reading watching minds eye enjoy rainy day	good read creatively written true character enjoy reading watching mind eye enjoy rainy day
10	the cursed child is a good steady read it wasnt long enough to suit me but then again its a play not a novel i like the characters the story was good if youre a consistent reader of the harry potter stories this is an essential part of it you will enjoy it	0	0	the curs child is a good steadi read it wasnt long enough to suit me but then again it a play not a novel i like the charact the stori wa good if your a consist reader of the harri potter stori thi is an essenti part of it you will enjoy it	the cursed child is a good steady read it wasnt long enough to suit me but then again it a play not a novel i like the character the story wa good if youre a consistent reader of the harry potter story this is an essential part of it you will enjoy it	curs child good steadi read wasnt long enough suit play novel like charact stori wa good consist reader harri potter stori thi essenti part enjoy	cursed child good steady read wasnt long enough suit play novel like characters story good youre consistent reader harry potter stories essential part enjoy	cursed child good steady read wasnt long enough suit play novel like character story wa good youre consistent reader harry potter story essential part enjoy
				im so not give away	im so not giving away			

Figure 18: Result after preprocessing text data

4.2.5 Word Count

As our data preprocessing progresses, the word count of our corpus are as follows:

```
There are 533637 words in the corpus.
There are 533802 words in the corpus after stemming.
There are 533637 words in the corpus after lemmatization.
There are 277840 words in the corpus after lemmatization and removal of stopwords.
There are 268819 words in the corpus after removal of stopwords.
There are 293184 words in the corpus after stemming and removal of stopwords.
```

Figure 19: Word Count

4.2.6 Vectorization and N-grams

```
from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer(ngram_range=(1,2))
comment_matrix = vectorizer.fit_transform(x_train['content'])

# from sklearn.feature_extraction.text import TfidfVectorizer
# vectorizer = TfidfVectorizer(use_idf=True, ngram_range=(1,2))
# comment_matrix = vectorizer.fit_transform(x_train['content'])
```

Figure 20: Count and TF-IDF Vectorizers and N-gram Code Snippet

Lastly, we transform the text data into vectors via the use of Python's Count Vectorization or TF-IDF Vectorization as shown in the code snippet above.

Count Vectorization provides a simple way for us to tokenize text documents and build a vocabulary of known words. It also allows us to encode new documents using this newly built vocabulary of known words. TF-IDF Vectorization is another form of vectorization, which uses the TF-IDF, an abbreviation for Term Frequency-Inverse Document Frequency. This TF-IDF is a numerical statistic that intends to reflect how important a word is in a document or a collection of documents.

N-grams of texts are extensively used in text mining and natural language processing tasks. An n-gram is a contiguous sequence of n items from a given sample of text or speech. In our model, we use a combination of bigrams and unigrams.

4.3 Build an evaluation dataset by manually labeling 10% of the collected data

Inter-annotator agreement is a measure of how well two (or more) annotators can make the same annotation decision for a certain category. As required, the 3 members of our group are the 3 judges who performed the labelling. In total, we manually labelled 1,001 records. The following code snippet shows the calculation of the Kappa score between 2 judges as well as between 3 judges.

```
judge_1 = labelled_df['Judge_1'].to_numpy()
judge_2 = labelled_df['Judge_2'].to_numpy()

from sklearn.metrics import cohen_kappa_score
kappa_boi = cohen_kappa_score(judge_1, judge_2)
print("The cohen's kappa score between 2 raters is {}".format(kappa_boi))
```

```
The cohen's kappa score between 2 raters is 0.9164047348358189
```

Figure 21: Cohen's Kappa Score Calculation

```
from nltk import agreement
Judge_1 = labelled_df['Judge_1'].to_numpy()
Judge_2 = labelled_df['Judge_2'].to_numpy()
Judge_3 = labelled_df['Judge_3'].to_numpy()

taskdata=[[0,str(i),str(Judge_1[i])]
          for i in range(0,len(Judge_1))]+[[1,str(i),str(Judge_2[i])]
          for i in range(0,len(Judge_2))]+[[2,str(i),str(Judge_3[i])]
          for i in range(0,len(Judge_3))]
ratingtask = agreement.AnnotationTask(data=taskdata)
print("The fleiss'kappa score between 3 raters is " + str(ratingtask.multi_kappa()))
```

```
The fleiss'kappa score between 3 raters is 0.9346946488517371
```

Figure 22: Fleiss' Kappa Score Calculation

Since the Kappa score is approximately 0.9347, which is above 0.8 (80%), our evaluation dataset is considered to have a good inter-annotator agreement.

"from a hp fan this book was not good erg i waited years for this its an ok read but not up to the original book standards"

In addition to calculating the Kappa score, there were also some reviews that were misclassified such as the above which is labelled as a negative review when the sentiment is clearly positive. This could potentially affect the training of the dataset, which prompted us to reclassify the sentiment label.

4.4 Provide evaluation metrics

With our processed data, we split it into 80% as training data and the other 20% as testing data. The training data will be used to train the chosen classifiers while the testing data will be used to evaluate the model performance. The following are the confusion matrix and evaluation of each model.

4.4.1 Naïve Bayes Classification

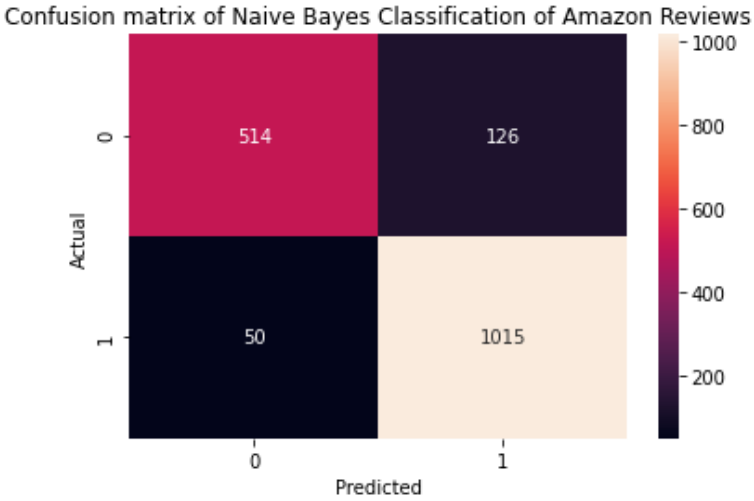


Figure 23: Confusion Matrix of Naive Bayes Classifier

4.4.2 Optimization of Naïve Bayes Classifier

Vectorizer	Preprocessing	F1 Score	Precision	Recall	Average precision-recall score	% change in F1 score from base vectorizer
Count	None	0.915	0.880	0.953	0.867	Base
Count	Stemming	0.908	0.871	0.949	0.858	-0.7
Count	Lemmatization	0.910	0.872	0.952	0.860	-0.5
Count	Stopwords	0.912	0.873	0.953	0.861	-0.3
Count	Stemming + Stopwords	0.902	0.863	0.945	0.850	-1.3
Count	Bigram	0.912	0.872	0.957	0.861	-0.3
Count	Unigram + Bigram	0.920	0.890	0.953	0.877	+0.5
Count	Unigram + Bigram + Trigram	0.918	0.888	0.950	0.875	+0.3
Count	Bigram + Trigram	0.915	0.874	0.960	0.864	0
TF-IDF	None	0.859	0.759	0.990	0.757	Base
TF-IDF	Lemmatization	0.858	0.758	0.990	0.756	-0.1
TF-IDF	Unigram + Bigram	0.839	0.726	0.995	0.725	-2
TF-IDF	Lemmatization + Stopwords	0.877	0.791	0.986	0.788	+1.8
TF-IDF	Lemmatization + Stopwords + sublinear tf scaling	0.880	0.794	0.986	0.792	+2.1

Figure 24: Comparison of preprocessing methods with Naïve Bayes Classifier

Several preprocessing methods were experimented to determine the best possible result. The above table summarizes that the best f1 score can be generated from using Count Vectorizer and unigrams and bigrams as the preprocessing method. Stemming, lemmatization and stop words removal are hardly effective as preprocessing methods in sentiment analysis. We can also infer that the Count Vectorizer performs much better than the TF-IDF Vectorizer although using the TF-IDF Vectorizer does induce a higher recall. (0.986-0.995) Count Vectorizer and Unigram and Bigrams are used as the default preprocessing methods for the models, K-Nearest Neighbors, Support Vector Machines, Decision Tree and Random Forest.

4.4.3 K-Nearest Neighbour Classification

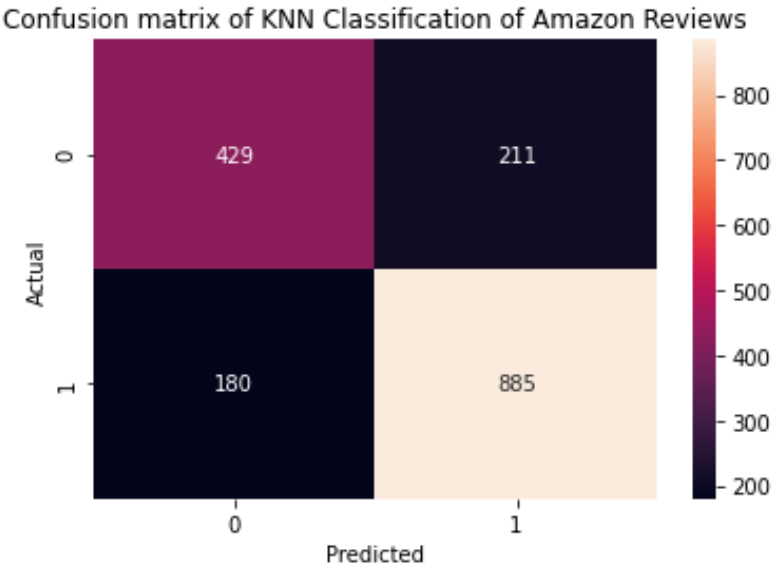


Figure 25: Confusion Matrix of K-Nearest Neighbour Classification

4.4.4 Support Vector Machine Classification

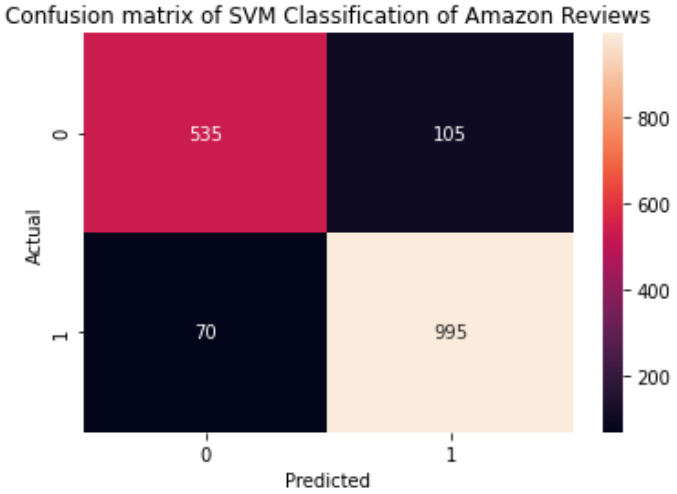


Figure 26: Confusion Matrix of SVM Classification

4.4.5 Decision Tree Classification

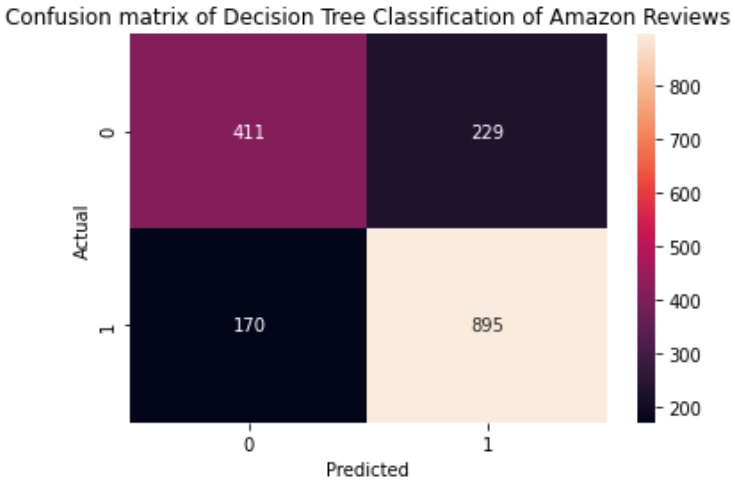


Figure 27: Confusion Matrix of Decision Tree Classification

4.4.6 Summary and Discussion of Evaluation Results

Model	F1 Score	Precision Score	Recall Score	CV Folds
Naïve Bayes Classification	0.863	0.901	0.830	5
K-Nearest Neighbor Classification	0.754	0.813	0.705	5
Linear Support Vector Machine Classification	0.873	0.849	0.899	5
Decision Tree Classification	0.782	0.737	0.812	5

After optimizing the model by performing the best preprocessing methods and choosing the best hyperparameters, we perform a 5-fold cross validation to ensure the validity of our results. The Linear SVM model generates the highest F1 score among the 4 models at 0.873 with Naïve Bayes classifier coming in close at 0.863.

4.5 Discuss performance metrics

4.5.1 Average Review Length

The average number of words in a review is 62.62.

4.5.2 Records classified per second

Based on the evaluation results above, we picked the top 2 models that perform the best and calculated the number of records classified per second and the time taken to train the model using the magic command timeit in Jupyter Notebook. From the below table, we can infer that although SVM performs better than

the Naïve Bayes classifier in terms of F1 score, the time taken to train the model and the speed of classifying a record was significantly slower for SVM than the Naïve Bayes classifier. The Naïve Bayes classifier would be preferred over SVM if we were to train a large dataset.

Method	Records classified per second	Time to train Model	F1 Score
Support Vector Machines	856.281	10.7 seconds	0.873
Naïve Bayes Classifier	856281.407	0.0178 seconds	0.863

4.5.3 Optimizing SVM for speed

To enhance the classification speed of SVM and reduce the time needed to train the model, we can perform stop words removal on top of count vectorizing and using n-grams. Although there is a slight decrease in the F1 score, there is a significant increase in the number of records classified per second as well as a large decrease in training time. Thus, this makes SVM a viable option for classifying and training large datasets.

Method	Records classified per second	% increase in records classified	Time to train model	% decrease in training time	F1 Score	Precision	Recall
SVM	856.281	Base	10.7s	Base	0.8733	0.8489	0.8995
SVM + Stop words removal	1331.25	55.4%	6.5s	39.2%	0.8731	0.8227	0.9305

4.6 Visualizing Classified Data

4.6.1 Word Cloud

A word cloud is a collection, or cluster, of words depicted in different sizes. Figure 28 and 29 displays the top 100 words for positive and negative reviews.

Figure 28: Word Cloud of Positive Reviews

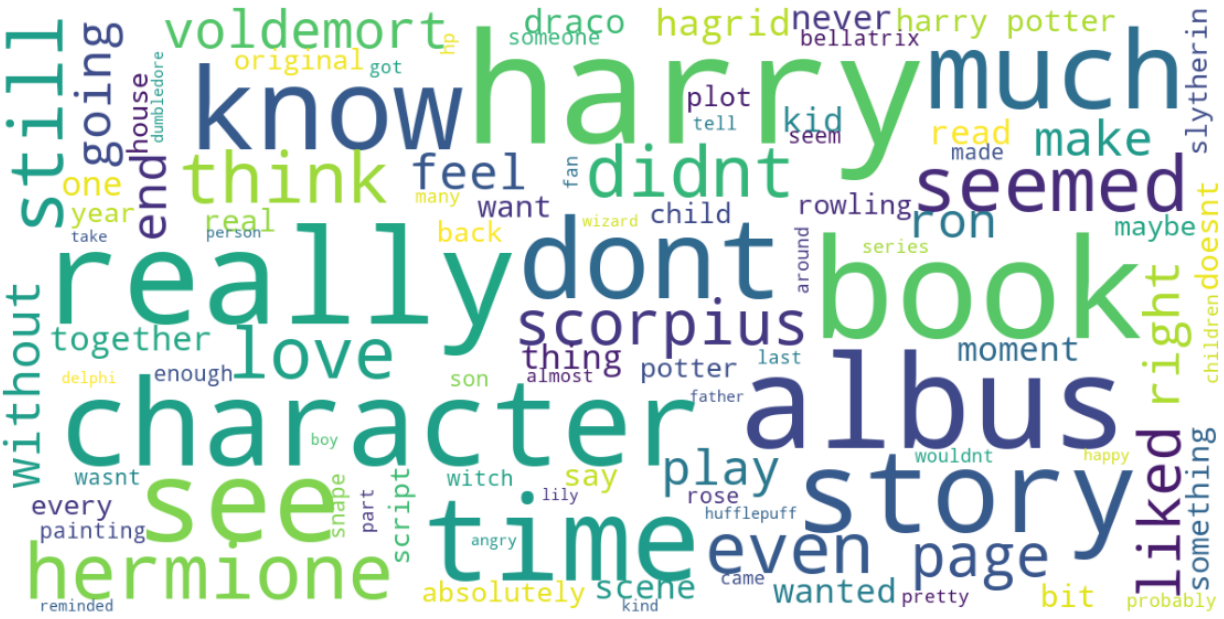


Figure 29: Word Cloud of Negative reviews

4.6.2 Topic Modeling with Latent Dirichlet Allocation

Topic Modeling

To group features that tend to co-occur in the same reviews we use Latent Dirichlet Allocation (LDA), a topic modeling algorithm. LDA is a probabilistic distribution algorithm which uses Gibbs sampling to assign topics to documents. In LDA a topic is a probabilistic distribution over words and each document is modeled as a mixture of topics. This means that each review can be associated to different topics and that topics are associated to different words with a certain probability

To prepare good segregation topics, the data was preprocessed into tokens and lemmatization and stop words removal were then performed. Words that have fewer than 4 characters are removed.

A dictionary was then created from the data and converted to a bag-of-words corpus. The pyLDAvis library was used to analyze and discover useful insights based on the visualization created.

Interpreting the Visualization

LDAvis is a library which extracts information from a topic model and creates a visualization where users can interactively explore the model. Below is a guide on how to interpret the visualization.

1. The red bars represent the frequency of each word given a topic.
2. The blue bars represent the overall frequency of the word in the corpus.
3. The area of the circle represents the topic prevalence.
4. The distance between topics is the approximation of the semantic relationship between the topics. When two topics are close to each other, they are semantically related.
5. By decreasing the relevance metric (lambda) on the right corner, we put more weight to the frequency of the topic as compared to the overall frequency of the word. The lower the lambda value, the easier it is to what the topic is about.
6. By setting the lambda value to 0, we can see words that are exclusive to the topic
7. By hovering over a specific word, we can observe where the word is used in other topics. It can be used to see how words are applied in different contexts.

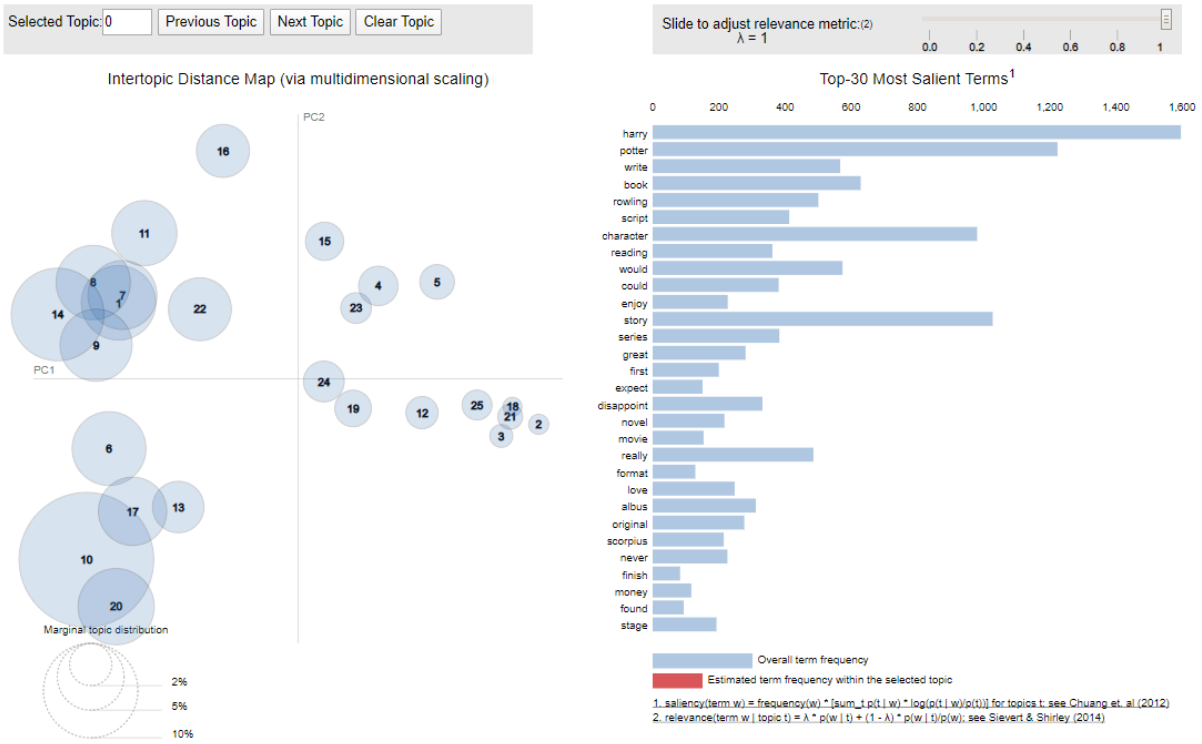


Figure 30: LDA Visualization

The above shows 25 topics that are defined on an intertopic distance map. The right shows the top 30 most salient terms in the corpus.

Insights

For the remaining part of this section, we will be exploring and analyzing possible reasons why people disliked or loved the book as well as some topics that were brought up by the reviewers.

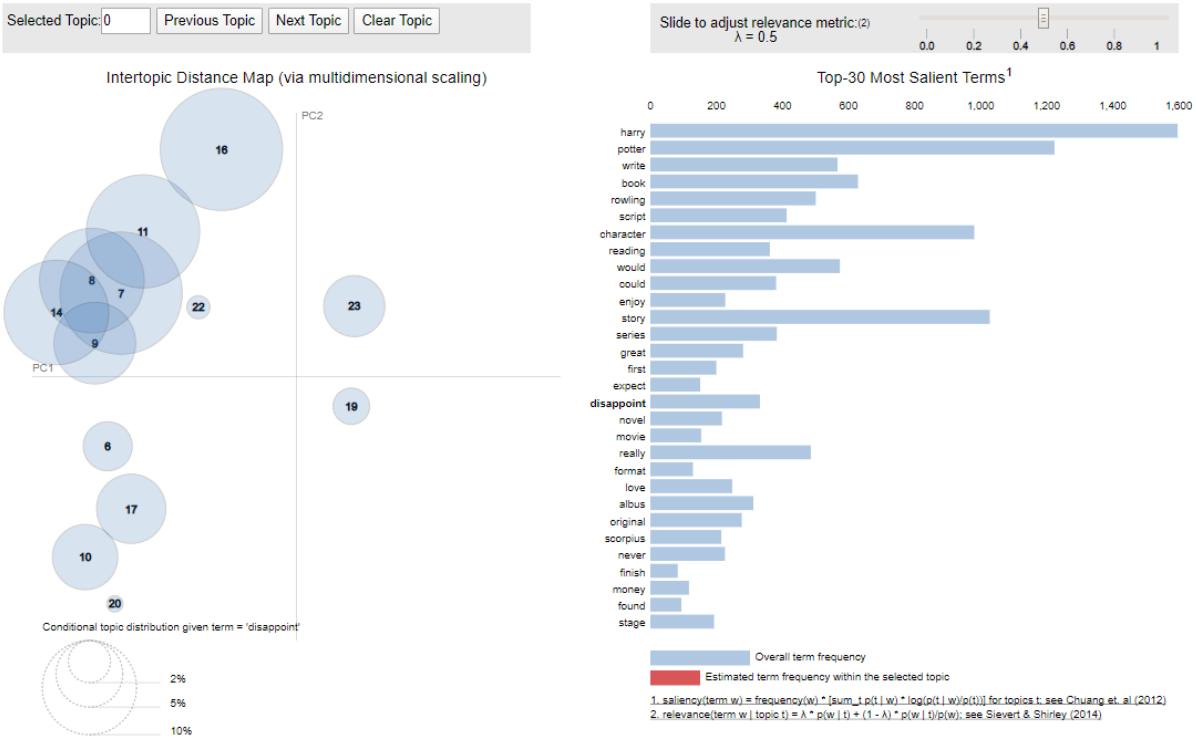


Figure 31: Term ‘disappoint’

Among the reviews, there were numerous reviewers who indicated their disappointment in the eighth book of the series. By hovering over the disappoint term, we can observe that there are a substantial number of topics relating to the word ‘disappoint’.

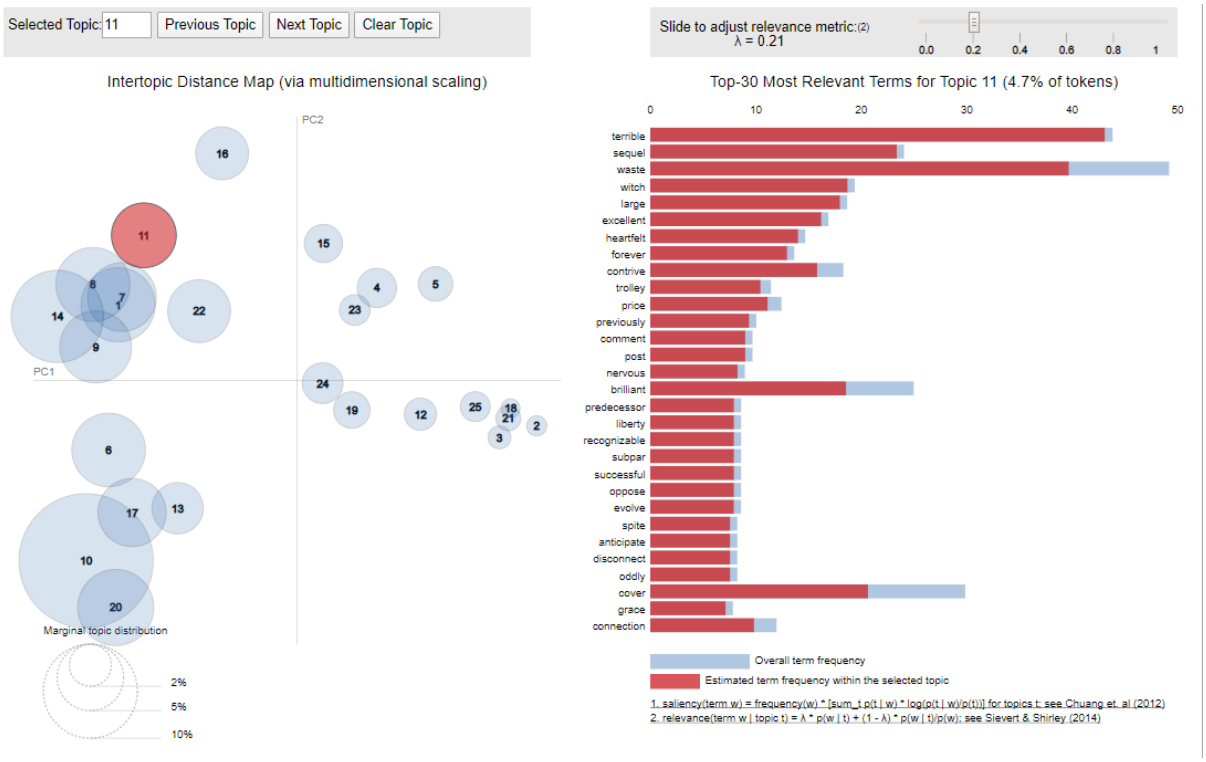


Figure 32: Topic 11 related to term 'disappoint'

There are some reviewers mentioning that the book is 'terrible', perhaps for a 'sequel' and how it could be a 'waste' of time and a 'disconnect' from the past seven books. However, there were also people mentioning positive words such as 'heartfelt', 'successful' and 'excellent'.

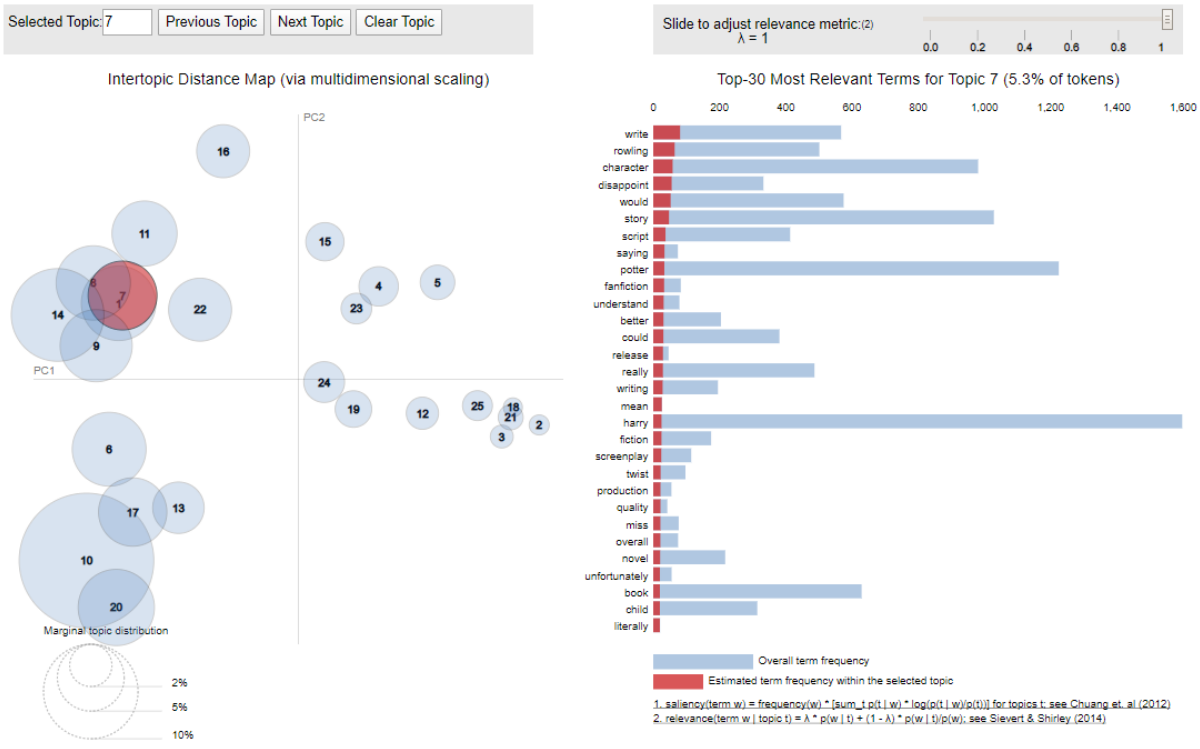


Figure 33: Topic 7 related to term 'disappoint'

A similar topic to topic 11 also mentioned words such as 'script', 'screenplay' and 'production', possibly hinting about the connection with the official play in London. The word 'fanfiction' was also brought up, possibly comparing the book to reading a fanfiction.

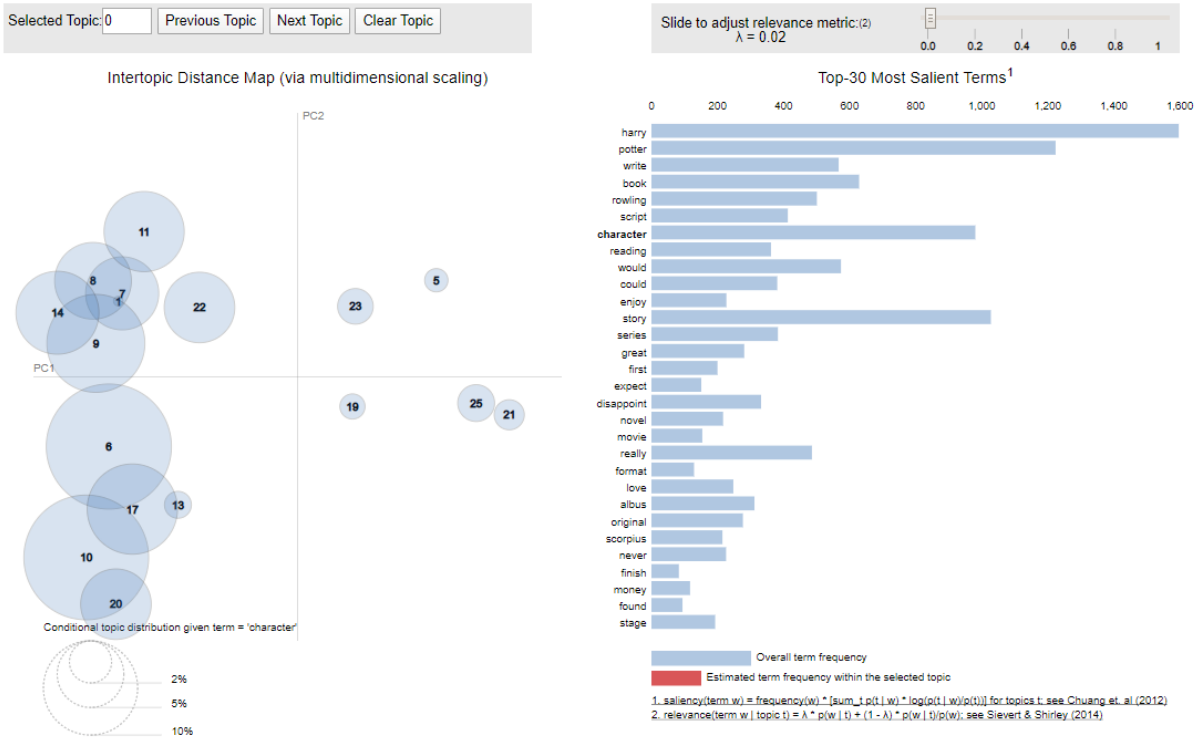


Figure 34: Term 'character'

The term 'character' is also one of the top 30 most salient terms.

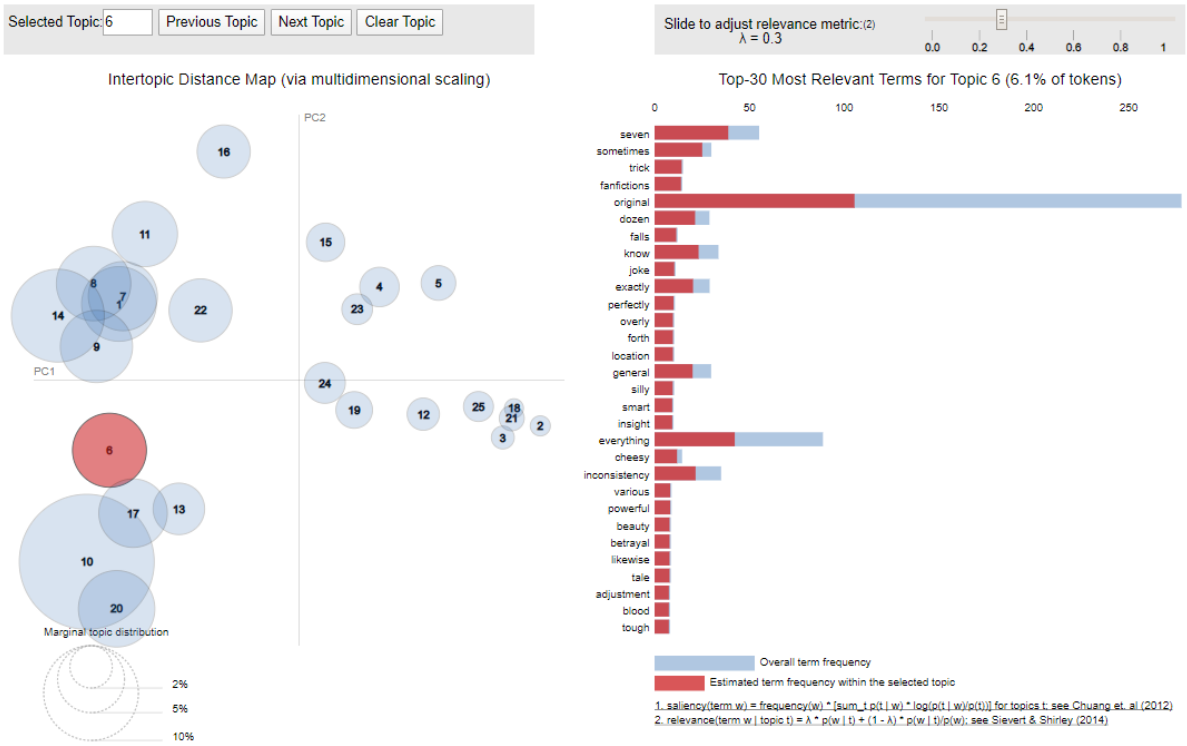


Figure 35: Topic 6 relating to term 'character'

From the above, we can infer that the reviewers made a comparison between the 'original' book written by J. K. Rowling and the current book written by Jack Thorne. Readers also made a comparison to fanfiction. Words such as 'cheesy' and 'inconsistency' were also brought up to describe the book.

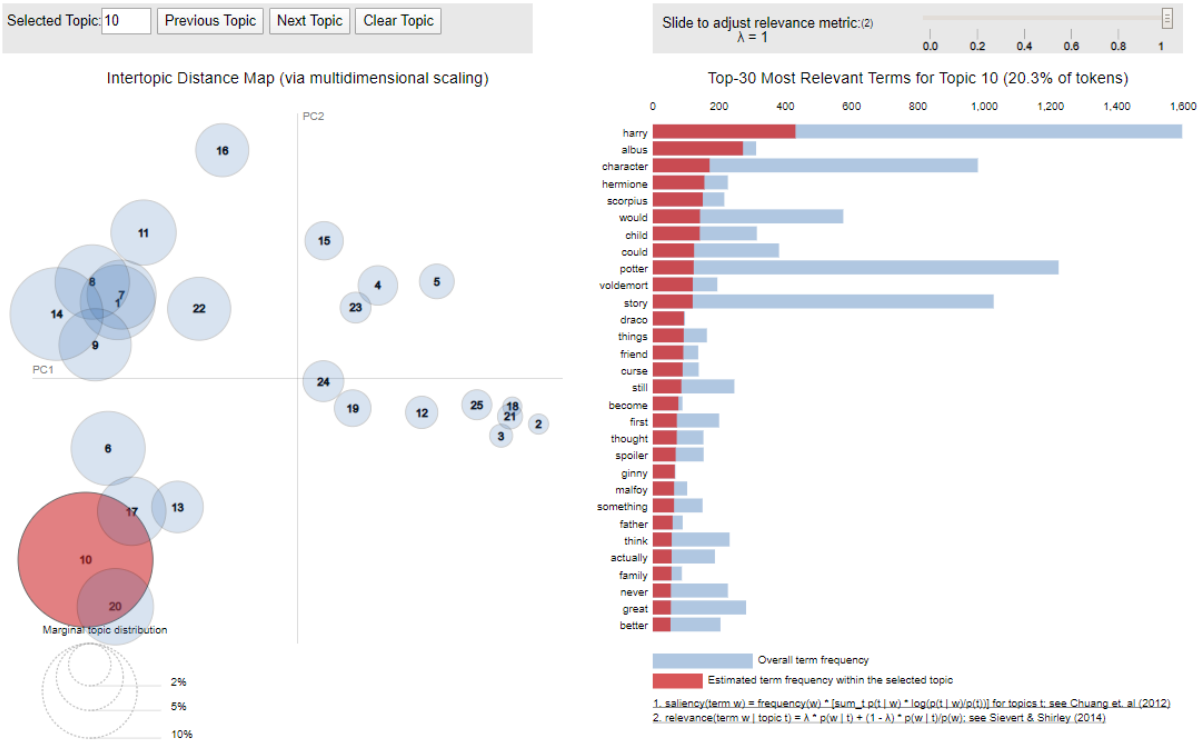


Figure 36: Topic 10 relating to term 'character'

From the above, we can also infer that reviewers mentioning character talked most about Albus, the main character of the book.

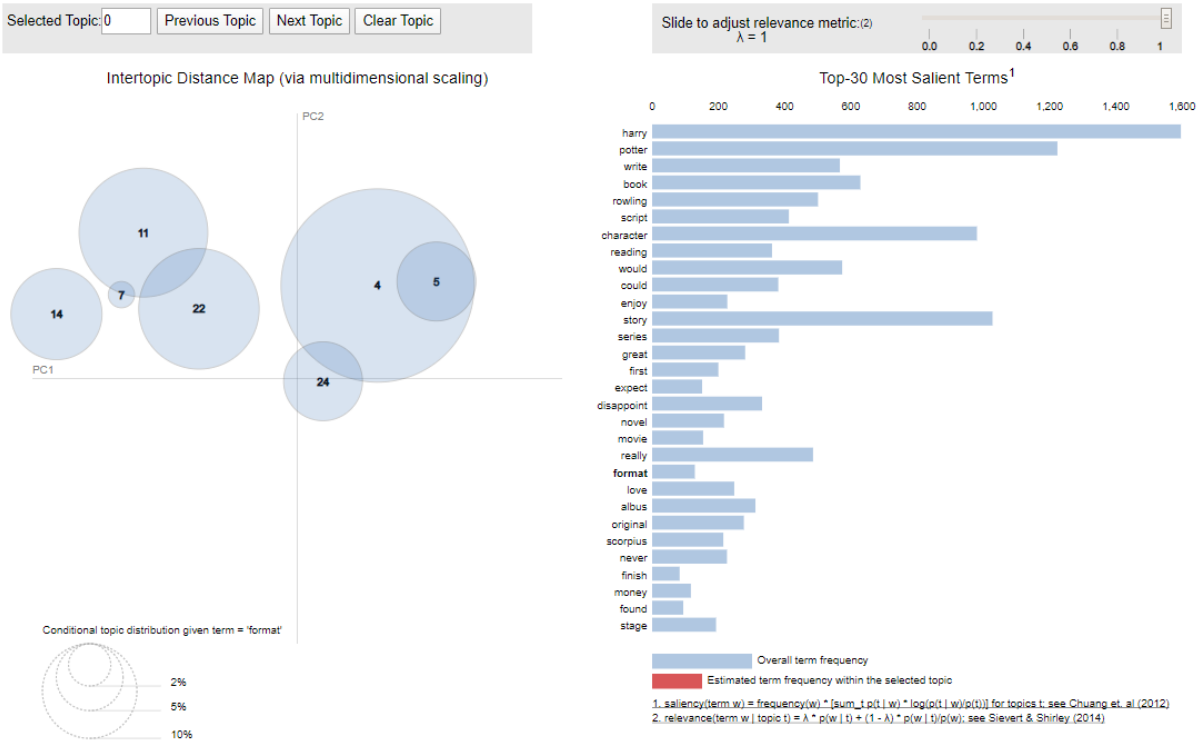


Figure 37: Term 'format'

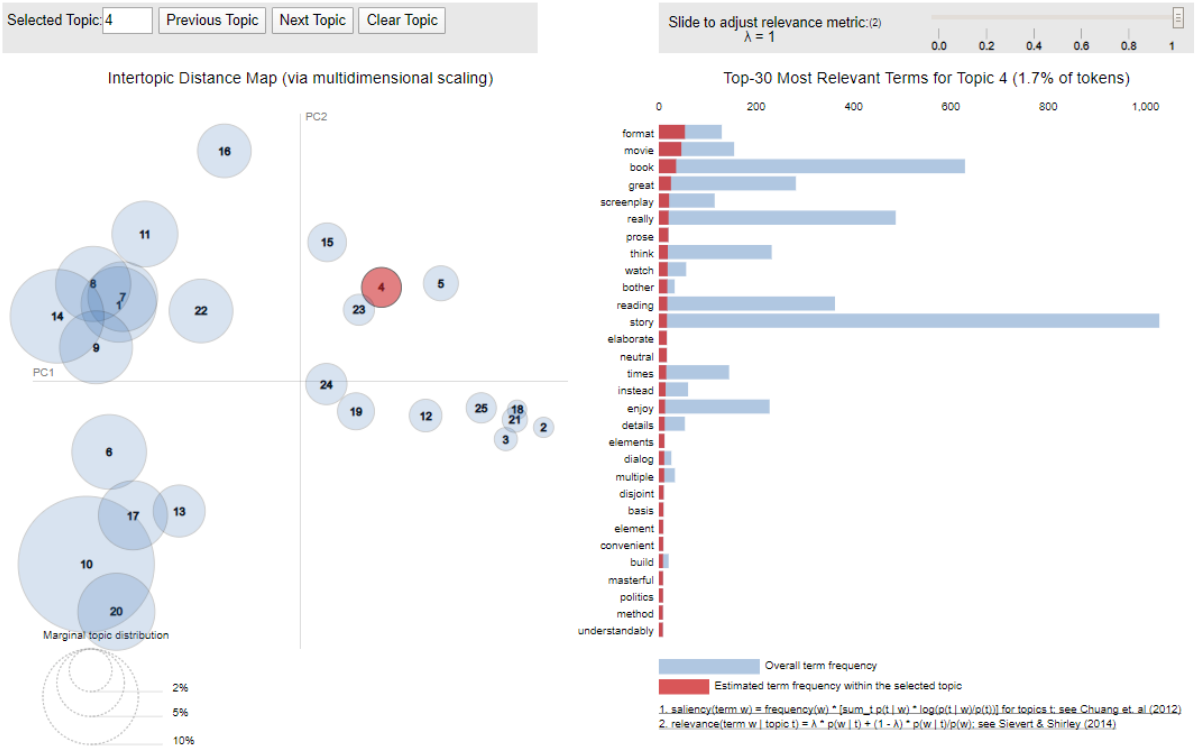


Figure 38: Topic related to term ‘format’

From the words appearing for topic 4 relating to the term ‘format’, we can roughly tell that most of the people in this topic found it to be ‘great’ and ‘masterful’. Words such as ‘movie’, ‘prose’, ‘screenplay’, ‘dialog’ could possible indicate that people were generally content with the format of the book and how it could be adapted to a movie.

5 Question 5 - Innovations for enhancing classification

5.1 Ensemble Classification – Random Forest Classification

Some of the main causes for errors in the classification models are due to noise, variance and bias. In order to reduce and minimize such factors, ensemble models are often used instead of the normal models. Unlike normal models, ensemble models combine the decisions from multiple models to improve its overall performance and results. One such ensemble classification model is the Random Forest Classification model.

The Random Forest Classification model is an ensemble model that works by constructing multiple decision trees, which uses the Decision Tree Classification, during training and output the class that is the mode of the classes. Using multiple random decision trees instead of a single decision tree will correct the decision trees' habit of overfitting to their training dataset, resulting in a better performance and result. As such, we decided to test out this classification model to compare with the other models, especially the Decision Tree Classification model.

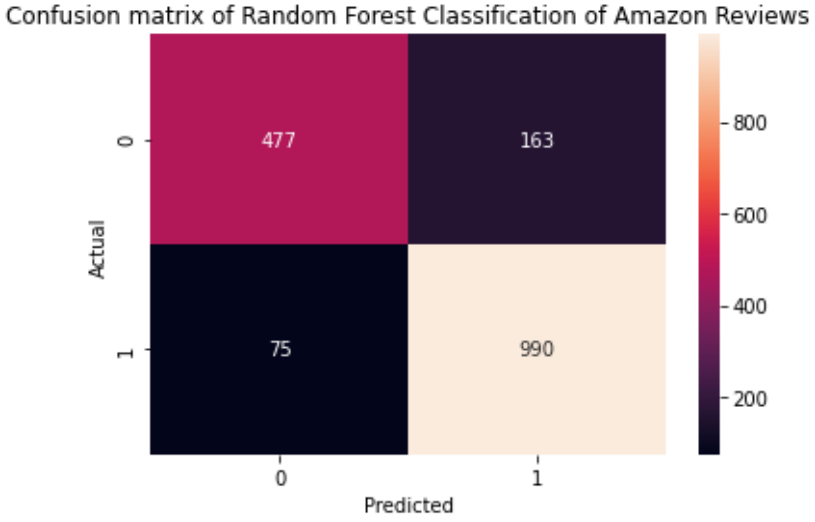
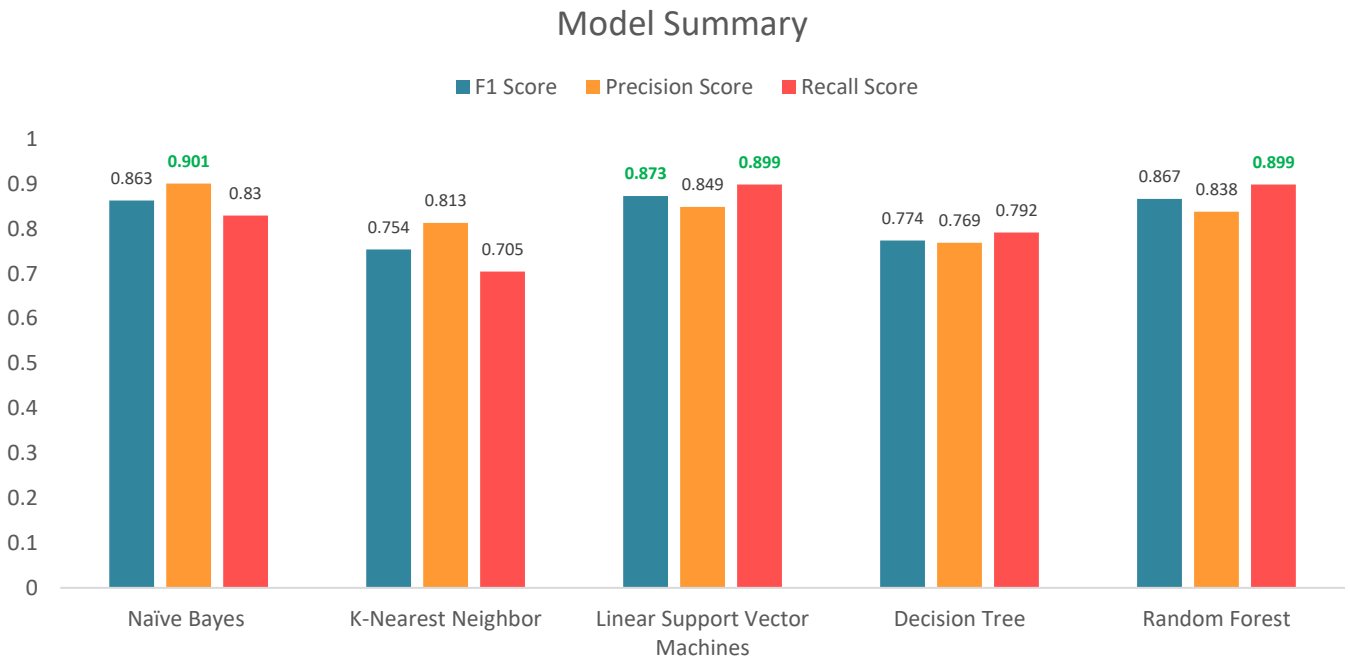


Figure 39: Confusion Matrix for Random Forest Classification

Model	F1 Score	Precision Score	Recall Score	CV Folds
Naïve Bayes Classification	0.863	0.901	0.830	5
K-Nearest Neighbor Classification	0.754	0.813	0.705	5
Linear Support Vector Machine Classification	0.873	0.849	0.899	5
Decision Tree Classification	0.774	0.769	0.792	5
Random Forest Classification	**0.867**	**0.838**	**0.899**	5



As shown, the evaluation results of Random Forest classification model are higher than the results of Decision Tree classification model even though both are based on the Decision Tree classifier. The F1 score of the Random Forest classification model also surpassed the Naïve Bayes classification model, which initially has the 2nd highest score.

5.2 k-Fold Cross Validation

To better enhance our classification, we should avoid possible overfitting issues. One such method is we used is the k-Fold Cross Validation, which refers to validating our evaluation results using different training and testing datasets within the original processed dataset and repeating this process *k* number of times. For enhancing our classification, we decided on a 5-Fold Cross Validation.

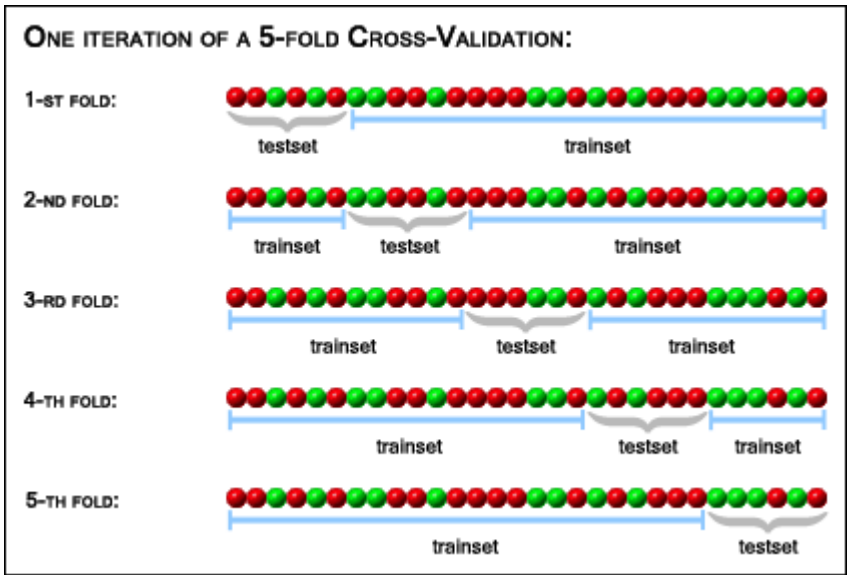


Figure 40: 5-Fold Cross Validation

As shown above, by using different training and testing datasets, that are parts of the original processed dataset, we will acquire multiple results for each evaluation metrics. In doing so, we will be able to compare

and check how accurate the initial results were and how consistent the results are when the training and testing datasets change.

We then calculate the mean value of each evaluation metric to acquire a less biased result that is better able to generalize new data. The table below shows the results for each evaluation metrics of each model used before and after 5-Fold Cross Validation.

Model	F1 Score		Precision Score		Recall Score	
	Before	**After**	Before	**After**	Before	**After**
Naïve Bayes Classification	0.920	**0.863**	0.890	**0.901**	0.953	**0.830**
K-Nearest Neighbour Classification	0.819	**0.754**	0.807	**0.813**	0.831	**0.705**
Linear Support Vector Machine Classification	0.919	**0.873**	0.905	**0.849**	0.934	**0.899**
Decision Tree Classification	0.796	**0.774**	0.787	**0.769**	0.805	**0.792**
Random Forest Classification	0.895	**0.867**	0.859	**0.838**	0.935	**0.899**

5.3 GridSearchCV

GridSearchCV is a function in scikit-learn that allows you to determine the best hyperparameters for a model by passing in a range of parameters' values. We performed grid search on all models used to ensure that the results of the model can be compared equally while also performing cross validation to avoid overfitting.

K-Nearest Neighbours

One of the most important hyperparameters of the k-nearest neighbours' model is the metric parametric which decides how distances are calculated in space. Upon further testing some of the distance metrics such as Euclidean, Manhattan and Cosine distance. Cosine distance proves to be the best distance metric. Cosine distance measures the cosine of the angle between two vectors and is highly advantageous for measuring the similarity of documents. A list of values ranging from 1 to 50 for the number of neighbours and the leaf size were also passed into the function to find the best number of clusters and the best leaf size.

Support Vector Machines

For support vector machines, a total of 4 hyperparameters were manipulated, namely C, gamma, kernel type and degree. C is the penalty parameter of the error term; it controls the tradeoff between smooth decision boundary and classifying the training points correctly. Gamma is a parameter that measures the radius of the area of influence of the support vectors. It is imperative to use a gamma level that allows for generalization while capturing the complexity of the data. Kernel parameters selects the type of hyperplane used to separate the data. Using 'linear' will use a linear hyperplane while 'rbf' and 'poly' uses a non-linear hyperplane. Degree is a parameter used for the polynomial kernel type and indicates the degree of the polynomial to be used. After performing grid search, it was concluded that a gamma level of 0.2, C value of 1, and a poly kernel with a degree of 1 is to be used.

Decision Trees

When predicting using the decision trees, the quality of the split can be measured by gini for gini impurity or entropy for information gain. This can be manipulated under the hyperparameter criterion in the DecisionTreeClassifier() function. Other hyperparameters tested include splitter which is the strategy used to choose the split at each node and the maximum number of features to consider when looking for the best split.

Random Forests

Random Forest classifier works in a similar manner to the Decision Tree classifier. For tuning of hyperparameters in Random Forests, we decided to tweak the maximum number of features and the number of decision trees, represented by n_estimators, that is used to predict the test set.

5.4 Error Analysis

After generating the confusion matrix, the incorrectly classified text was also analyzed to further enhance the classification of the model.

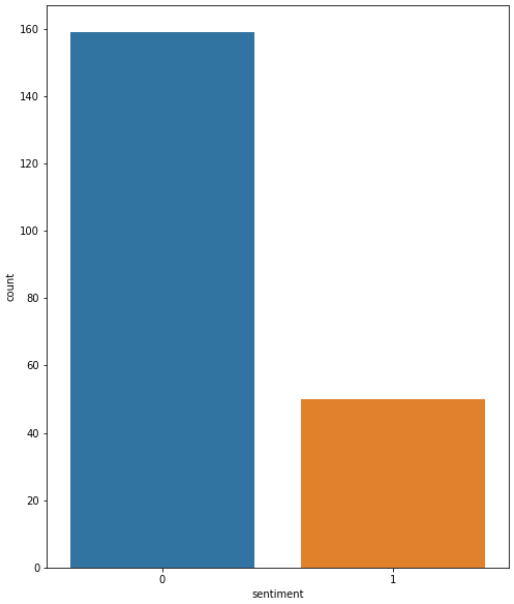


Figure 41: No. of false positives and negatives based on sentiment

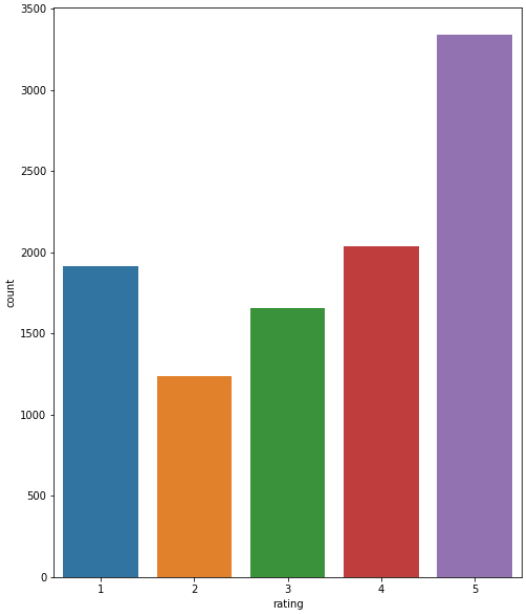


Figure 42: No. of reviews based on rating

As mentioned earlier, a sentiment value of 0 represents a negative rating of 1 or 2 stars while a sentiment value of 1 represents a positive rating of 4 or 5 stars. In the above figure, we can discern that most of the incorrectly classified data belongs to the negative reviews. This could likely be due to the imbalanced proportion of positives and negatives in the dataset which will lead to a different number of errors when tested against the true number.

After carefully analyzing the errors by viewing the errors directly (can be viewed in wrongly_classified.csv), the source of errors can be categorized into 3 forms, user error, indirect expression and algorithm limitation.

User Error

"maybe it works as a play its aweful as a book soppy and overly melodramatic"

Some of the words were given the wrong scores as there were spelling mistakes or grammatical errors. The

word “awful” and “sloppy” is misspelled as “aweful” and “soppy”, misclassifying the review as positive when it is in fact, negative.

Indirect Expression

“I’m a huge harry potter fan but i honestly can’t say anything good about this wish i could get a refund”

“the story is interesting but i do not like reading a script”

Some reviews expressed their opinion on how they enjoyed the Harry Potter Series but claimed that they did not enjoy the book. Considering words such as “huge harry potter fan” and “story is interesting”, the output becomes positive instead of negative because of the negation.

Algorithm Limitations

The majority of the misclassifications are due to the algorithm being unable to handle complex expressions of sentiment in texts. Some of the lengthier reviews may proof to be difficult to classify even for a human reader.

5.5 Future Considerations

```
def custom_preprocessing(x):
    x = re.sub('(https?://[\S]+)', ' URL ', str(x))
    x = x.replace("xmass", "christmas")
    x = x.replace("...", " ")
    x = x.replace("'ll", "will")
    x = x.replace("tewwible", "terrible")
    x = x.replace("frekking", "freaking")
    return x
```

Figure 43: Custom Preprocessing

In light of the errors made during classification, the above figure shows ways that we could further reduce the misclassification rate. One example would be replacing links starting with https with a string ‘URL’. Spelling mistakes can be corrected by using an external library as well as replacing emojis with the relevant equivalent term before training the data or predicting the sentiment of a review.

6 Conclusion

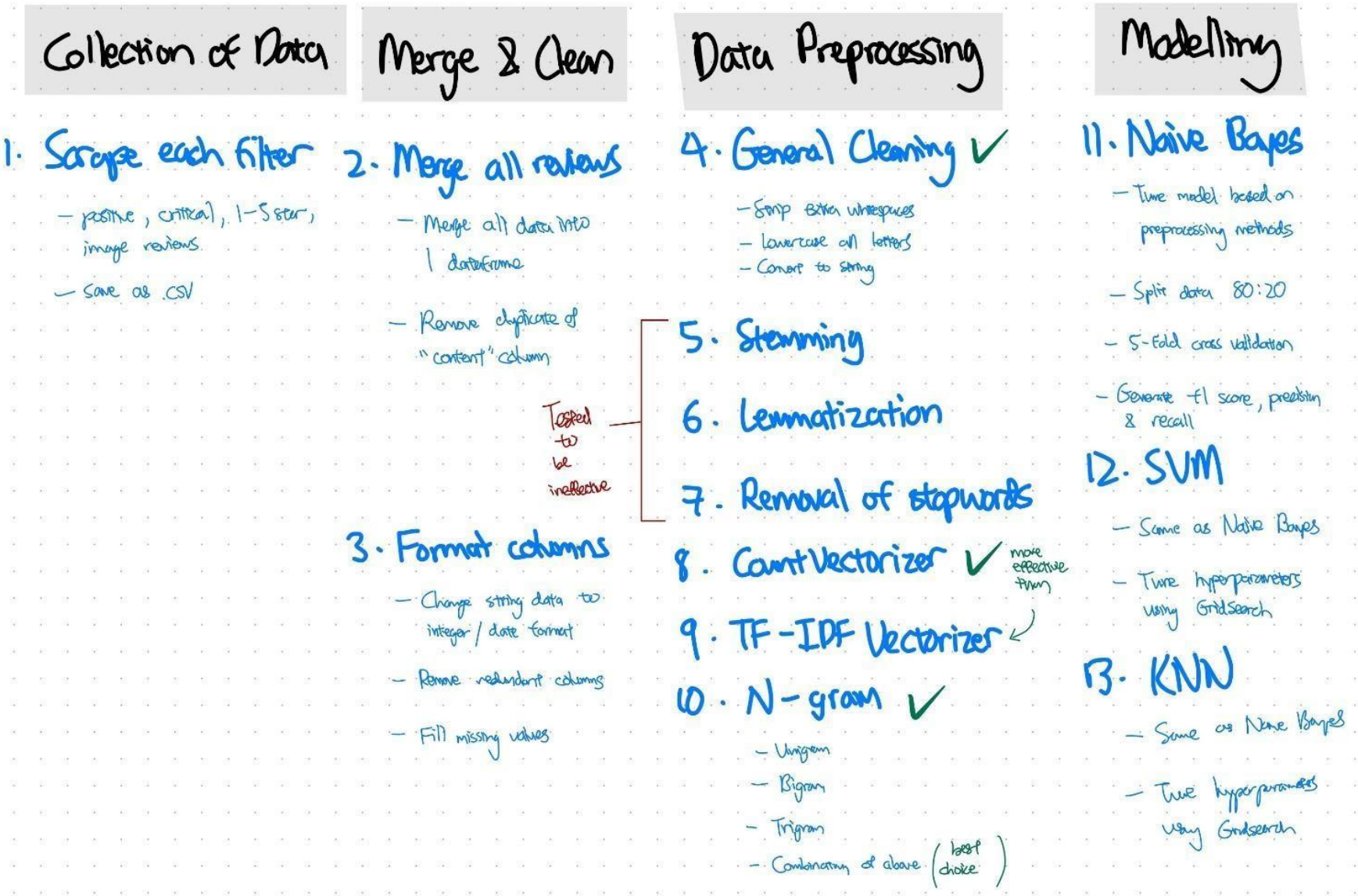
Through our project, our group has crawled for the latest customers’ reviews from the various filters and conditions available at the Amazon Customer Reviews webpages via the Google Chrome extension, Amazon Web Scraper. We then created a web interface, consisting of HTML and PHP files, and integrated them with the functionalities of the indexing software, Solr, where the required data manipulation, indexing and selection are carried out. After which, we proceed with data pre-processing using techniques such as merging and formatting the multiple CSV files of crawled data, removal of duplicated data, stemming, lemmatization, stop words removal and count vectorization. Following that, we performed our sentiment prediction using various classification models such as Naïve Bayes classifier, K-nearest Neighbour and Support Vector Machine. The models are then evaluated using evaluation metrics such as F-measure, precision and recall values and the results are compiled and analyzed. The powerful LDAvis library is used to visualize text data without difficulty. Lastly, we explored some innovations to enhance our classification such the use of GridSearch and k-fold Cross Validation and the use of ensemble classification model like Random Forest to compare with the models’ results.

7 Video, Data and Source Codes links

The following table will consist of the URL to the various resources used as well as the Youtube video link and the link to the Google Drive which holds our analysis data, analysis results and source codes.

Name	URL
Corpus Webpage	https://www.amazon.com/Harry-Potter-Cursed-Child-Parts/product-reviews/1338216678/ref=cm_cr_getr_d_show_all?ie=UTF8&reviewerType=all_reviews&pageNumber=1
Amazon Review Scraper Usage Guide	https://www.scrapehero.com/amazon-review-scraper/
Web Scraper Extension	https://chrome.google.com/webstore/detail/web-scraper/jnhgnonknehpejjnehehllkliplmbmhn/related
Cross Validation Illustration	https://www.google.com/url?sa=i&url=https%3A%2F%2Fgenome.tugraz.at%2Fproclassify%2Fhelp%2Fpages%2FXV.html&psig=AOvVaw1cE5VbaCX9h-mpP6CX3pxy&ust=1585054919794000&source=images&cd=vfe&ved=0CAIQjRxqFwoTCOiAhfDTsOgCFQAAAAAdAAAAABAH
Youtube Video	https://www.youtube.com/watch?v=SD77bGp-U0g&feature=youtu.be
Google Drive	Google Drive Link: https://drive.google.com/drive/folders/1ajbRtrn8waqA7XmWZnQd7CrPAt8-SZof?usp=sharing Directory of Folder: https://docs.google.com/document/d/12U1ISqKn5TwVq-aWQ1pCeL1dwmot0XRmFjr1Z7upSGA/edit?usp=sharing

8 Appendix



Overview of Data Analysis Process