Mari Kristine Knutsen ECON4170 20.11.2021

Text mining and text classification

TABLE OF CONTENTS

Introduction

In this project I used a data set of text and labels. I did some data wrangling and text mining. I also tried out some text analysis methods and multi-class classification of text data.

Disclaimer 1: This project has suffered from a wish to try a range of things, instead of doing one thing perfectly.

Disclaimer 2: I have found a lot of help with the coding in forums etc., some are copied, some are just the inspiration, and some are made by using the Tidyverse's cheat cheets.

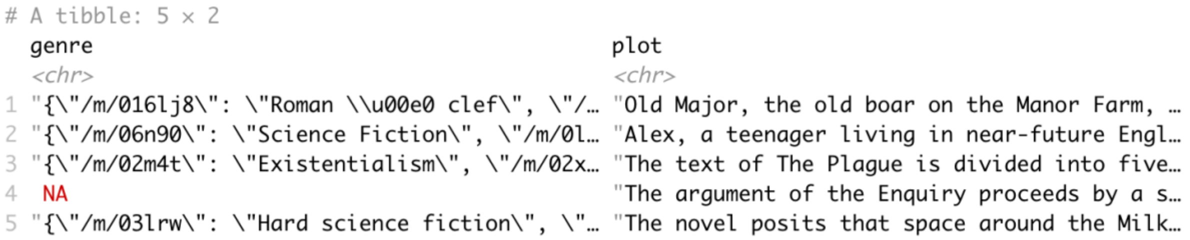
The data

Source:
I found the data through Kaggle, and then downloaded it from the original source:
https://www.cs.cmu.edu/~dbamman/booksummaries.html

Description of the data:
Plot summaries of 16,559 books extracted from the November 2, 2012 dump of English-language Wikipedia. Tab-separated; columns:

#	Description
1	Wikipedia article ID
2	Freebase ID
3	Book Title
4	Book Author
5	Publication date
6	Book genres (Freebase ID: name tuples)
7	Plot summary

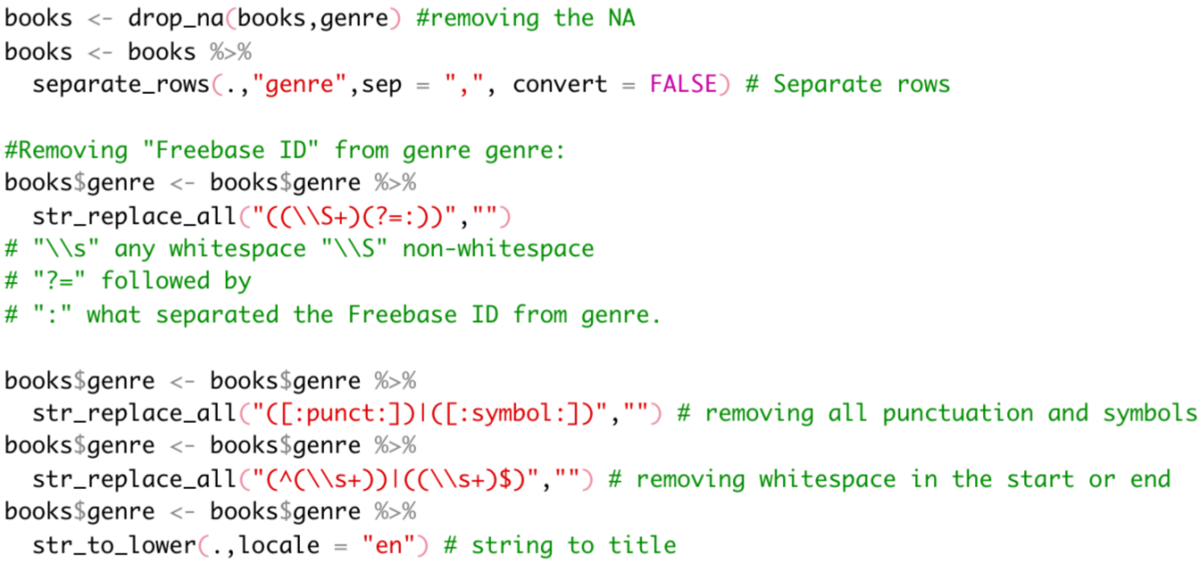
In this project "genre" and "plot" are the ones I will use. Under is a print of some of the data frame, and here we can see that genre is quite messy.



```
# A tibble: 5 × 2
  genre                                       plot
  <chr>                                       <chr>
1 "{\"/m/016lj8\": \"Roman \\u00e0 clef\", \"/… "Old Major, the old boar on the Manor Farm, …
2 "{\"/m/06n90\": \"Science Fiction\", \"/m/0l… "Alex, a teenager living in near-future Engl…
3 "{\"/m/02m4t\": \"Existentialism\", \"/m/02x… "The text of The Plague is divided into five…
4  NA                                          "The argument of the Enquiry proceeds by a s…
5 "{\"/m/03lrw\": \"Hard science fiction\", \"… "The novel posits that space around the Milk…
```

Cleaning

Method
To clean the genre column, I used methods from the packages Stringr (part of Tidyverse). Where I identified patterns and then removed them (replaced it with "").
Code

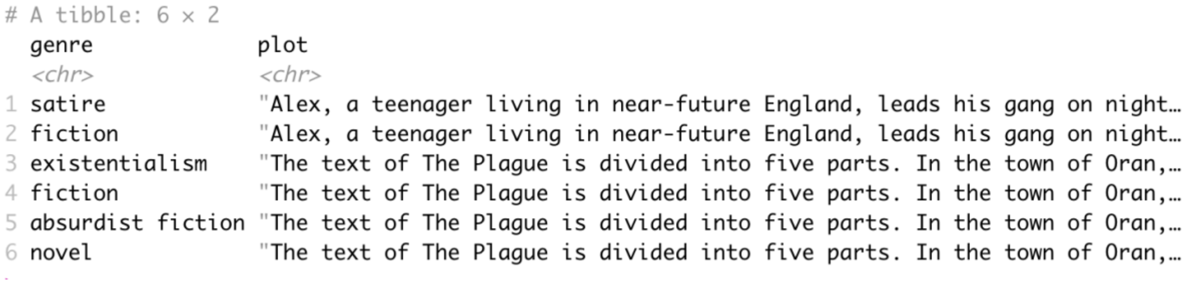


```
books <- drop_na(books,genre) #removing the NA
books <- books %>%
  separate_rows(.,"genre",sep = ",", convert = FALSE) # Separate rows

#Removing "Freebase ID" from genre genre:
books$genre <- books$genre %>%
  str_replace_all("((\\S+)(?=:))","")
# "\\s" any whitespace "\\S" non-whitespace
# "?=" followed by
# ":" what separated the Freebase ID from genre.

books$genre <- books$genre %>%
  str_replace_all("([:punct:])|([:symbol:])","") # removing all punctuation and symbols
books$genre <- books$genre %>%
  str_replace_all("(^(\\s+))|((\\s+)$)","") # removing whitespace in the start or end
books$genre <- books$genre %>%
  str_to_lower(.,locale = "en") # string to title
```

Result

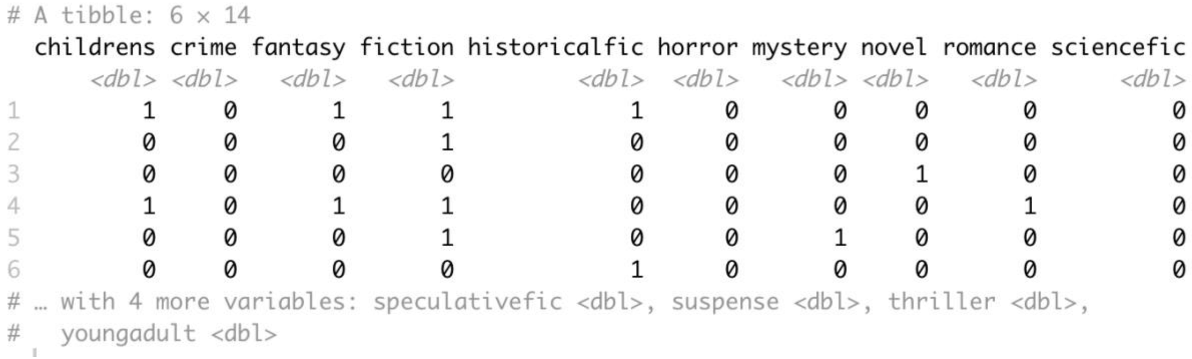


```
# A tibble: 6 × 2
  genre             plot
  <chr>             <chr>
1 satire            "Alex, a teenager living in near-future England, leads his gang on night…
2 fiction           "Alex, a teenager living in near-future England, leads his gang on night…
3 existentialism    "The text of The Plague is divided into five parts. In the town of Oran,…
4 fiction           "The text of The Plague is divided into five parts. In the town of Oran,…
5 absurdist fiction "The text of The Plague is divided into five parts. In the town of Oran,…
6 novel             "The text of The Plague is divided into five parts. In the town of Oran,…
```

Now we have each genre clean and separated. As we can see that two plots in this slice have more than one genre labeled.

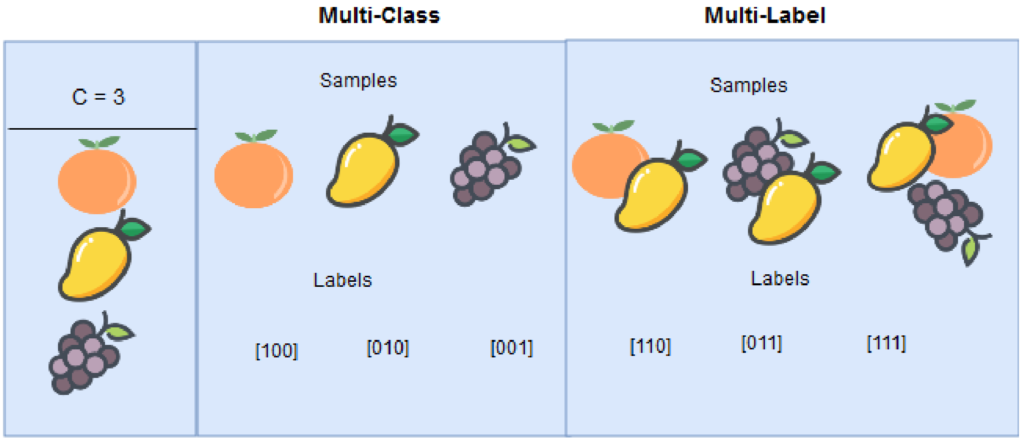
Multi-class VS. multi-label

Under I transformed the data frame, so that each line is one plot, and each column is a genre. 0 means the plot is not labeled with that genre and 1 means it is.



```
# A tibble: 6 × 14
  childrens crime fantasy fiction historicalfic horror mystery novel romance sciencefic
      <dbl> <dbl>   <dbl>   <dbl>         <dbl>  <dbl>   <dbl> <dbl>   <dbl>      <dbl>
1         1     0       1       1             1      0       0     0       0          0
2         0     0       0       1             0      0       0     0       0          0
3         0     0       0       0             0      0       0     1       0          0
4         1     0       1       1             0      0       0     0       1          0
5         0     0       0       1             0      0       1     0       0          0
6         0     0       0       0             1      0       0     0       0          0
# … with 4 more variables: speculativefic <dbl>, suspense <dbl>, thriller <dbl>,
#   youngadult <dbl>
```

This data is a multi-label dataset. The difference between multi-label and multi-class is best explained by a figure.



From: https://cynthiachuang.github.io/Difference-between-Multiclass-Multilabel-and-Multitask-Problem/

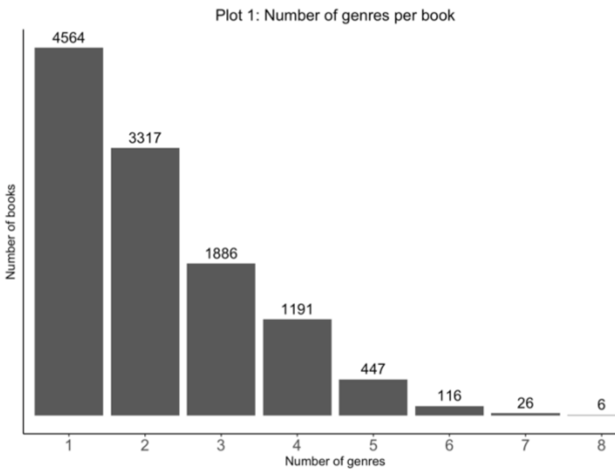
In the picture above the difference is illustrated by three different fruits. In the multi-class case we ask which fruit, and a fruit can have only one label. E.g., An orange can only be labeled as an orange, and not as Orange *and* grape. In the case of multi-label on the other hand, the observation might have multiple labels that fit. The question then might be, what fruit are in the photo, and here the answer might include more than one fruit.

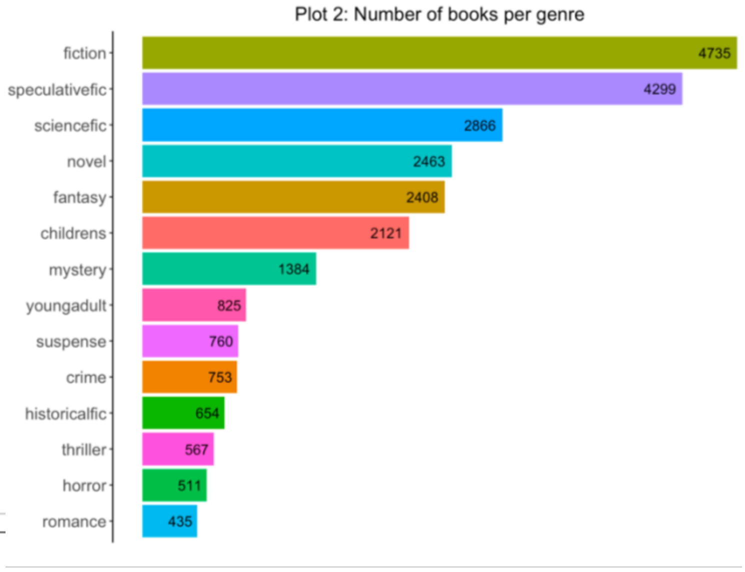
Visualization

Method

I wanted to explore how many books with 1,2,3… N genres labeled, and the number of books per genre. First, I counted the number of different genres, and found out that it was more than 200. Second, I filtered the genres that were labeled in more than 400 plots, and then I removed the rest from the data frame. I also shortened some of the names of the genres for faster coding and no whitespace-confusion.
To get the number of genres per book, I summarized each row, and then counted the frequency of the row values.

Results


Plot 1: Number of genres per book


Plot 2: Number of books per genre

In Plot 1 we see that 4564 plots had only 1 label and that 8 genres were the highest number of labels, which only 6 plots had.
In Plot 2 the number of books per genre is sorted from the most frequent genre, *fiction*, to the least frequent, *romance*.

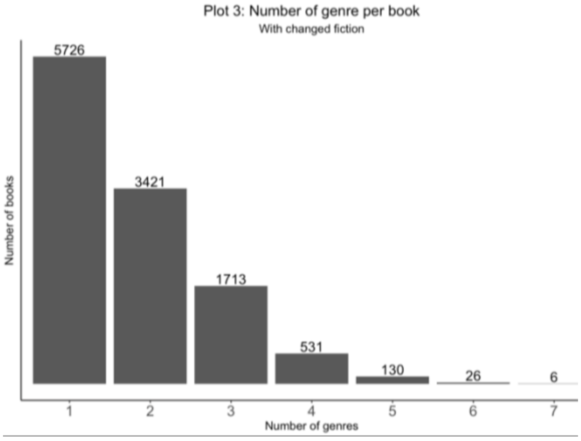
Transforming the data to multi-class

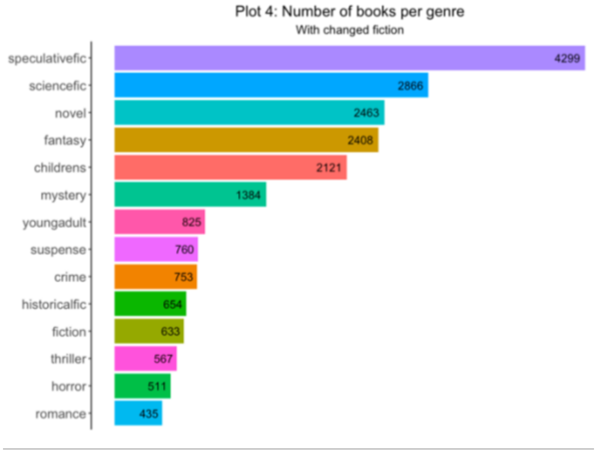
I tried to classification as a multi-label text classification, but it was too complex for a first-time machine learning project. I found a lot of help on how to do it in Python. I tried and failed to use some packages in R that were for multilabel classification. It was a lot of work to get the data into the right format, and if something was wrong it was hard to find information that would help. In this subject we have learned about Tidymodels and Caret, which there are a lot of information about. Therefore, for further text analysis and classification I transformed my data to multi-class.

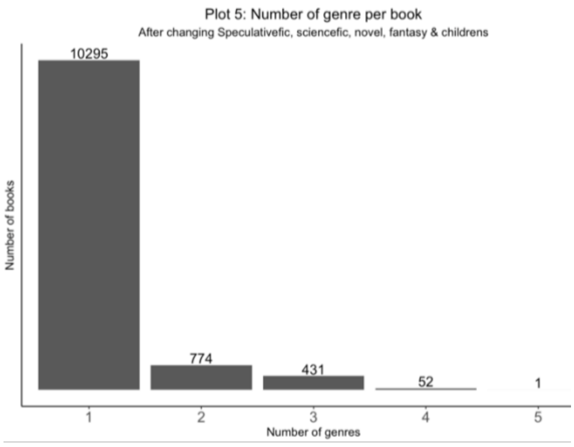
Method

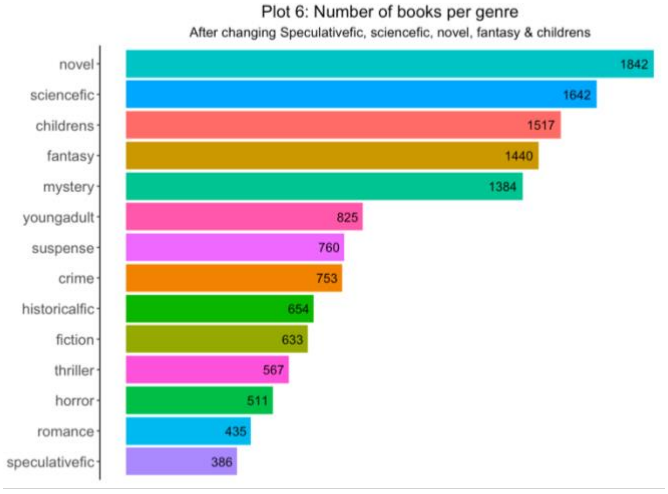
I first sorted out "fiction only" from the data set. That is, I removed fiction as a label on the plots that already had a different label. However, I kept fiction as a label where fiction was the only one. I Also did this to speculative fiction, science fic., novel, fantasy & children's lit.

Intermediate results


Plot 3: Number of genre per book
With changed fiction


Plot 4: Number of books per genre
With changed fiction

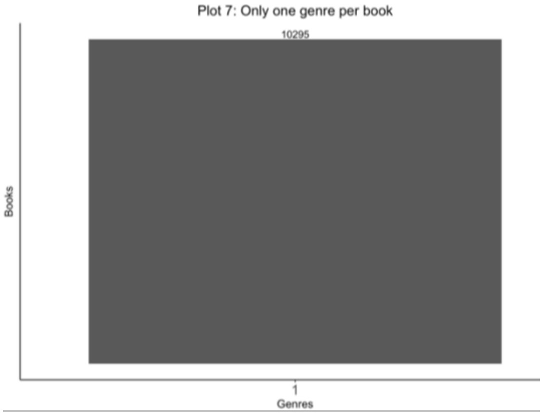

Plot 5: Number of genre per book
After changing Speculativefic, sciencefic, novel, fantasy & childrens

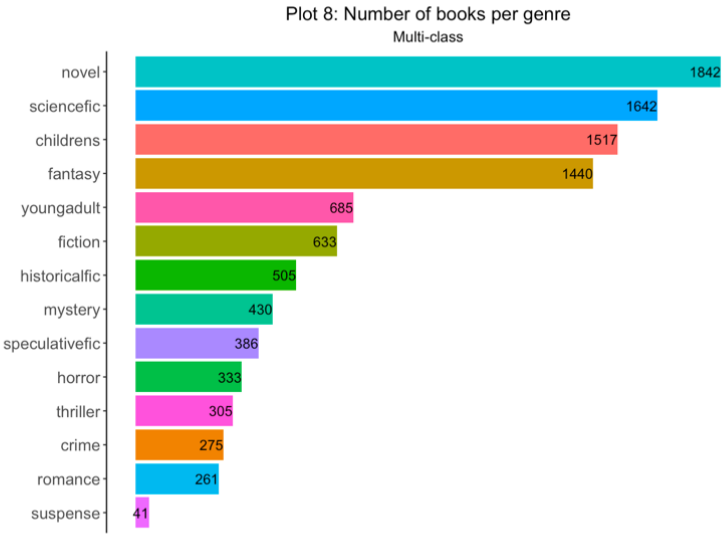

Plot 6: Number of books per genre
After changing Speculativefic, sciencefic, novel, fantasy & childrens

Plot 2 and Plot 3, show the changes in the dataset after the change of fiction labeling. Where Plot 2 show the number of genres per book, and plot 3 show the number of books per genre. Plot 5 and Plot 6 show the same but after all the changes. Plot 5 also illustrates that the books with more than one label are now a small minority.

Further Methods
Lastly, I removed the plots with more than one label as the remaining data were still a substantial size.

Results



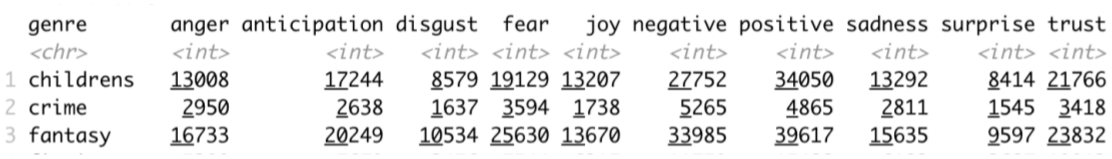


Plot 7, included for symmetry, confirm that only one genre is labeled per book. Plot 8 displays the new distribution of book per genre.

Sentiments

Sentiment lexicons are used to give a deeper meaning to the words in a text. This is a very useful tool when analyzing user reviews, comments, or other texts that expresses feelings or opinions (Silge & Robinson, 2021).
This might not be so decisive in book summaries, which are objective, however I wanted to examine whether there were changes in sentiments between plot summaries in genres like romance and horror.

Method
I followed the steps in the book "Text Mining in R: A tidy approach" by Silge & Robinson. Where I chose the sentiment dictionary "ncr" and combined the sentiments to the words in the plots. The slice under shows the number of sentiments of some of the genres. This dictionary uses categorical sentiments, but there are also sentiment dictionaries that assign a score as the sentiment.

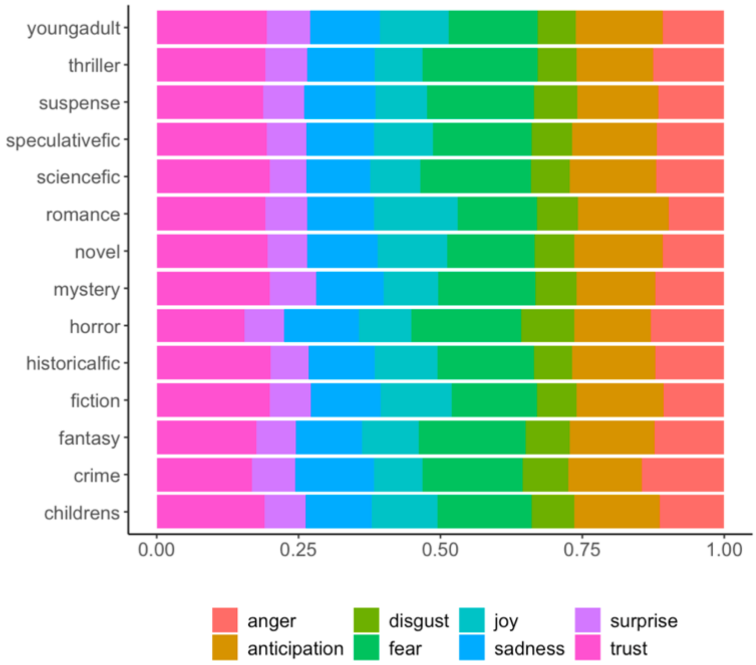


```
  genre      anger anticipation disgust  fear   joy negative positive sadness surprise trust
  <chr>      <int>        <int>   <int> <int> <int>    <int>    <int>   <int>    <int> <int>
1 childrens  13008        17244    8579 19129 13207    27752    34050   13292     8414 21766
2 crime       2950         2638    1637  3594  1738     5265     4865    2811     1545  3418
3 fantasy    16733        20249   10534 25630 13670    33985    39617   15635     9597 23832
```

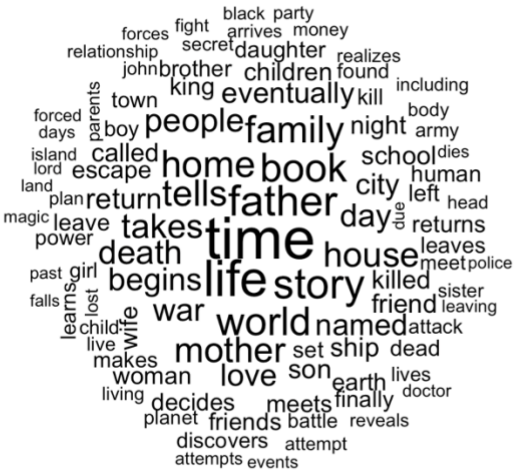
Result

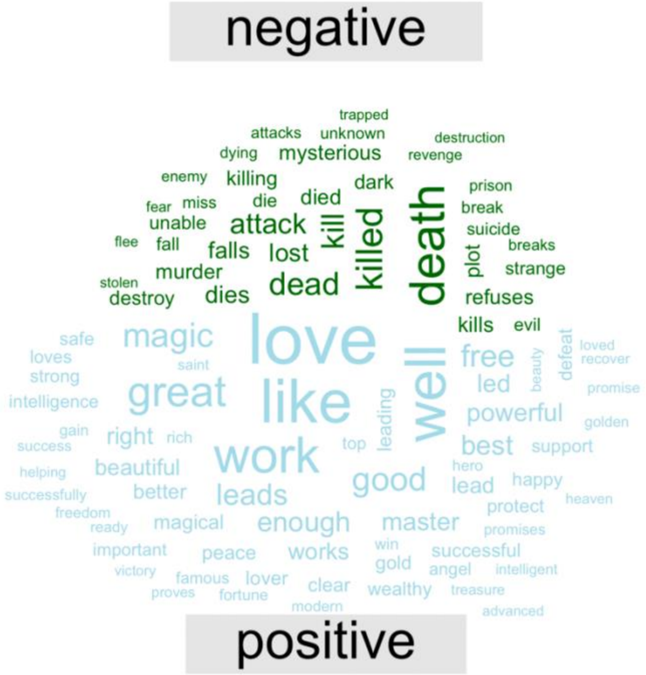
The genres had very different amounts of the sentiments, due to the large variation in number of books per genre. To compare between the genres I used position=”fill” in geom_bar. Plot 9 that det differences between the sentiments in the genres are small. There are however some slight variations in the sentiments *trust*, *joy* and *fear*. Where the plots with the label *horror* had less trust and more *fear* than *romance*. It also had slightly more *disgust* than the others. While *romance* had more *joy* and less *fear*, than the others.

Plot 9: Share of different sentiments in the genres



Wordclouds





The wordcloud on the left shows the most frequent words in the dataset (in the most frequent genres), where the most frequent words are the largest.

The wordcloud on the right shows the most frequent words with sentiments negative and positive, from the dictionary "bing". In this case the size is reflecting the frequency of a word within their sentiment. (Silge & Robinson, 2021).

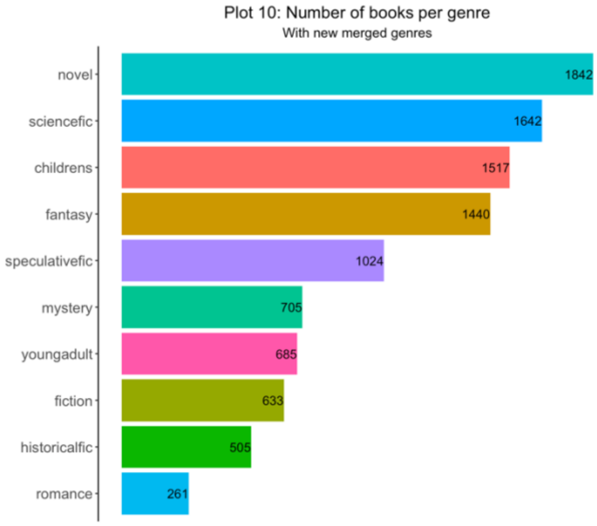
Merging genres

Method

I filtered out the 100 most used words in the plots of mystery, crime, speculative fiction, horror, and thriller. I then calculated the length of the intersection between two and two of the genres.

Results

I found that the genres with the most words in common where horror and thriller with speculative fiction, and between crime and mystery. Plot 10 illustrates the new distribution of genres.



TF-IDF –Term frequency -Inverse document frequency

Term frequency

TF is the number of times a word is used divided by the total number of words. The most common are usually words like *a*, *the*, *is, my, there* etc. These types of words are called Stopwords. These are usually not very important for the analysis as they are usually the most frequent words in all text, and therefore not helpful when we want the distinctive features of a text.

Inverse document frequency

IDF decreases the importance of frequent words and increases the importance of the words that are not much used in a document.

TF-IDF measures then how important a word is in a document collection of documents. In this case the importance of a word in plots that are of the genre *fantasy* in the collection of all the book plots (Silge & Robinson, 2021).

Method

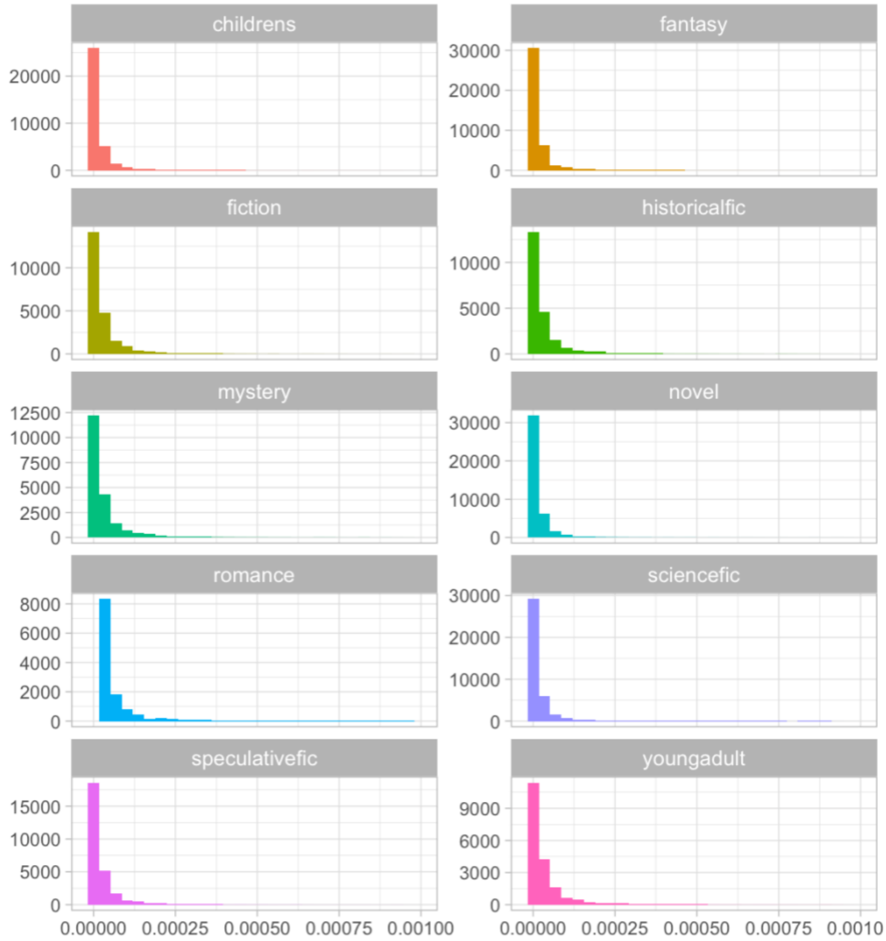
I used the code in the book “Text Mining in R: A tidy approach” by Silge & Robinson. But I used the function bind_tf_idf, to find both Term frequency and TF-IDF.

Results

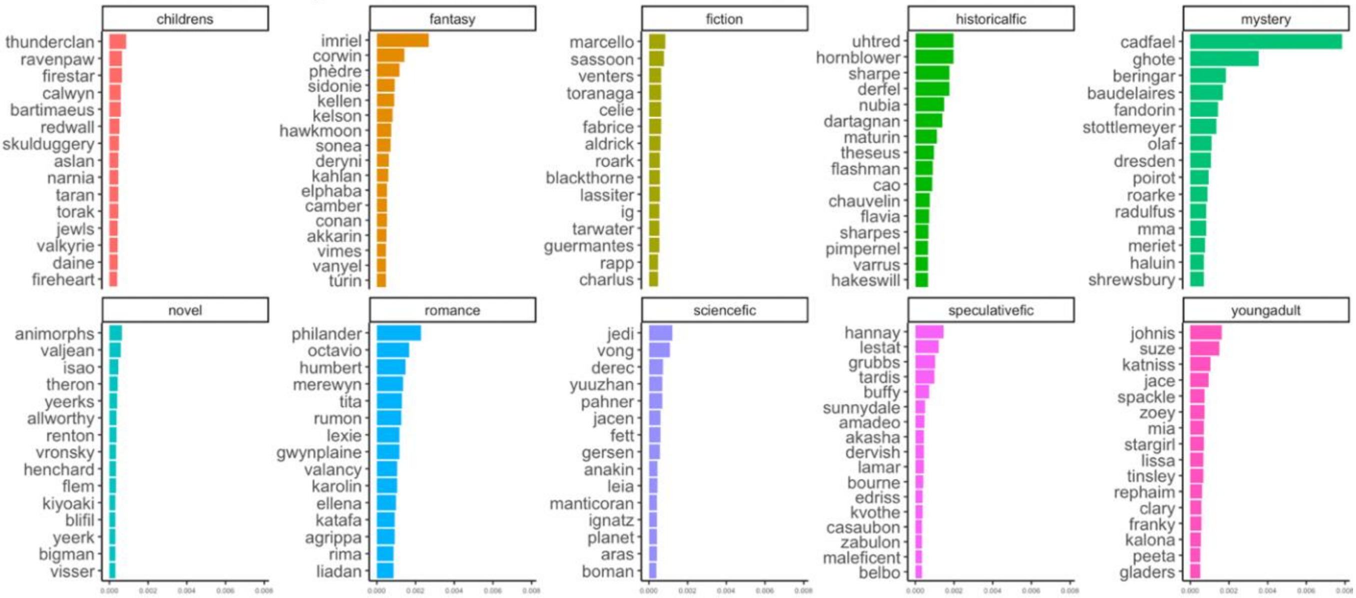
In Plot 11, the y-axis is the count of words, and the x-axis is the term frequency.

In Plot 12, the 15 words in each genre with the highest TF-IDF.

Plot 11: Term frequency



Plot 12: TF-IDF in each genre



Text classification

Given these plots will a model be able to predict the right genre?

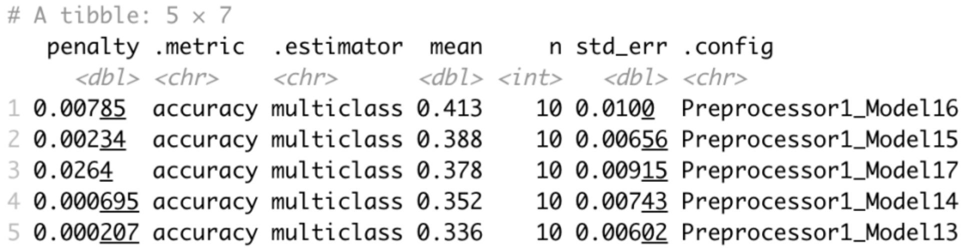
Method

I imitated the methods from a youtube video and Rmd by Andrew Couch, and "Supervised Machine Learning for Text Analysis in R" by Emil Hvitfeldt and Julia Silge.

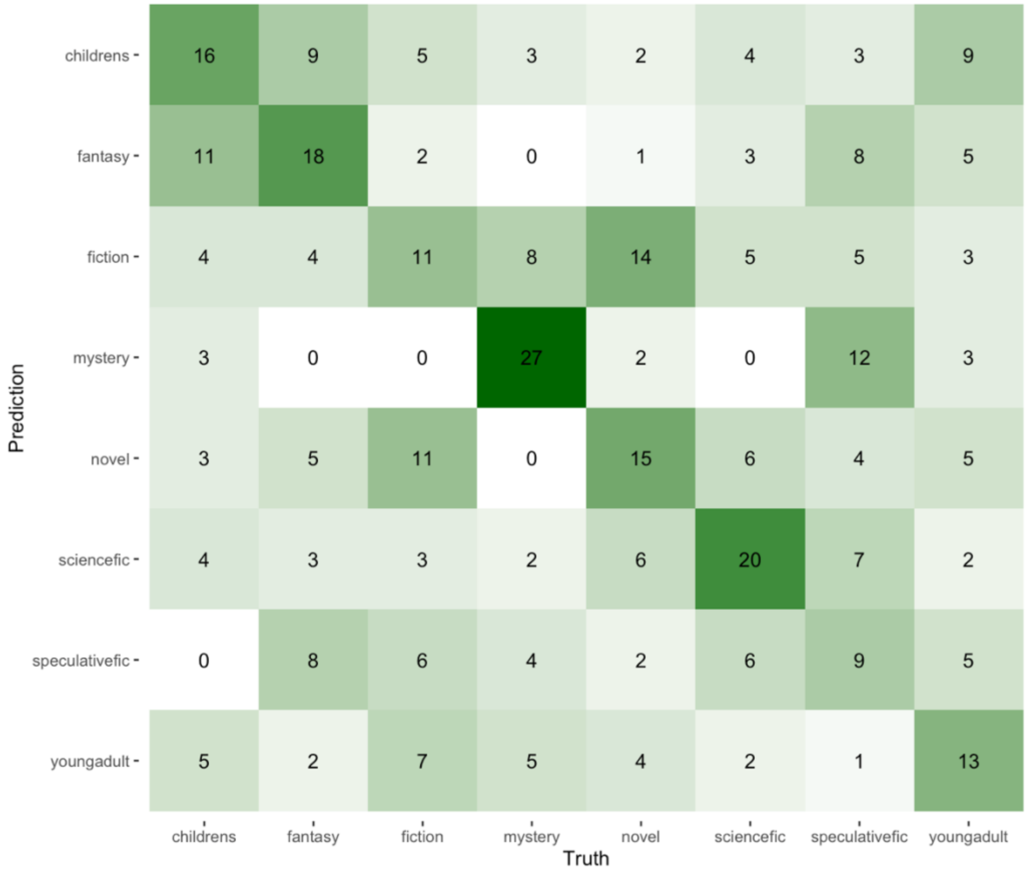
Where I initially used Couch's method with three models, lasso, knn and tree, with the text measured with hash, however this gave average accuracy under 30 percent in all the models. I tried SVM and naive bayes too without any improvement.

In the end I changed from hash to tf-idf and only used lasso like Hvitfeldt and Silge, this gave a slight improvement in average mean, but was still very low.

First results



```
# A tibble: 5 × 7
   penalty .metric  .estimator  mean     n std_err .config
     <dbl> <chr>    <chr>      <dbl> <int>   <dbl> <chr>
1 0.00785  accuracy multiclass 0.413    10 0.0100  Preprocessor1_Model16
2 0.00234  accuracy multiclass 0.388    10 0.00656 Preprocessor1_Model15
3 0.0264   accuracy multiclass 0.378    10 0.00915 Preprocessor1_Model17
4 0.000695 accuracy multiclass 0.352    10 0.00743 Preprocessor1_Model14
5 0.000207 accuracy multiclass 0.336    10 0.00602 Preprocessor1_Model13
```



	childrens	fantasy	fiction	mystery	novel	sciencefic	speculativefic	youngadult
childrens	16	9	5	3	2	4	3	9
fantasy	11	18	2	0	1	3	8	5
fiction	4	4	11	8	14	5	5	3
mystery	3	0	0	27	2	0	12	3
novel	3	5	11	0	15	6	4	5
sciencefic	4	3	3	2	6	20	7	2
speculativefic	0	8	6	4	2	6	9	5
youngadult	5	2	7	5	4	2	1	13

How to read the predictions to one of the genres in the confusion matrix:

Other genres	True Negatives	False Negatives	True Negatives
genre	False Positives	True Positive	False Positives
Other genres	True Negatives	False Negatives	True Negatives
	Other genres	genre	Other genres

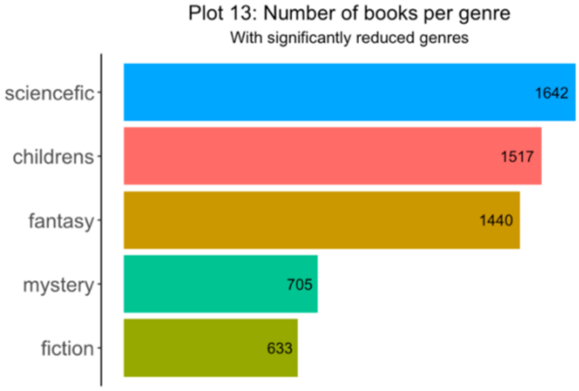
(Devopedia. 2019)

Changes

Trying a lot of different models, without improvement. I decided to remove some genres and try again.

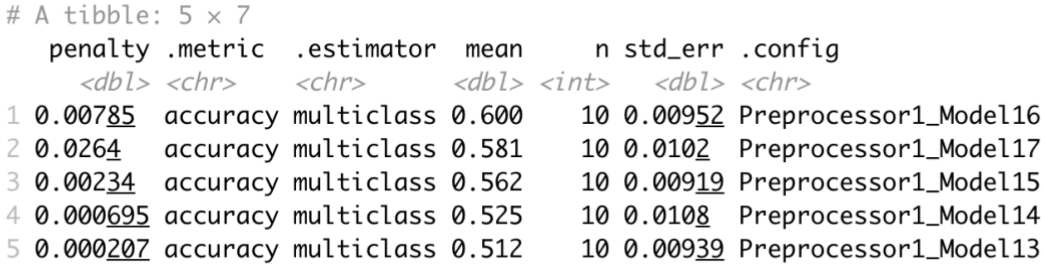
I "cheated" when I transformed the multi-label problem to a multi-class problem, knowing that the labels are not mutually excluding. I think that is partly why the models has a low accuracy. The other part I think has to do with the number of classes. With a binary classification problem there are 1 right and 1 wrong, but in the first attempt there where 1 right and 7 wrong answers.

I removed speculative fiction, novel, and young adult. I chose these because they have high Type I (False positives) and Type II errors (False negative). Shown in the confusion matrix above, where the diagonals are right predicted values. Where fiction has 11 false negative – where the model predicted novel, and novel has 14 false negative, where the model predicted fiction. Similar results are shown between speculative fiction and mystery, and children's and young adult.

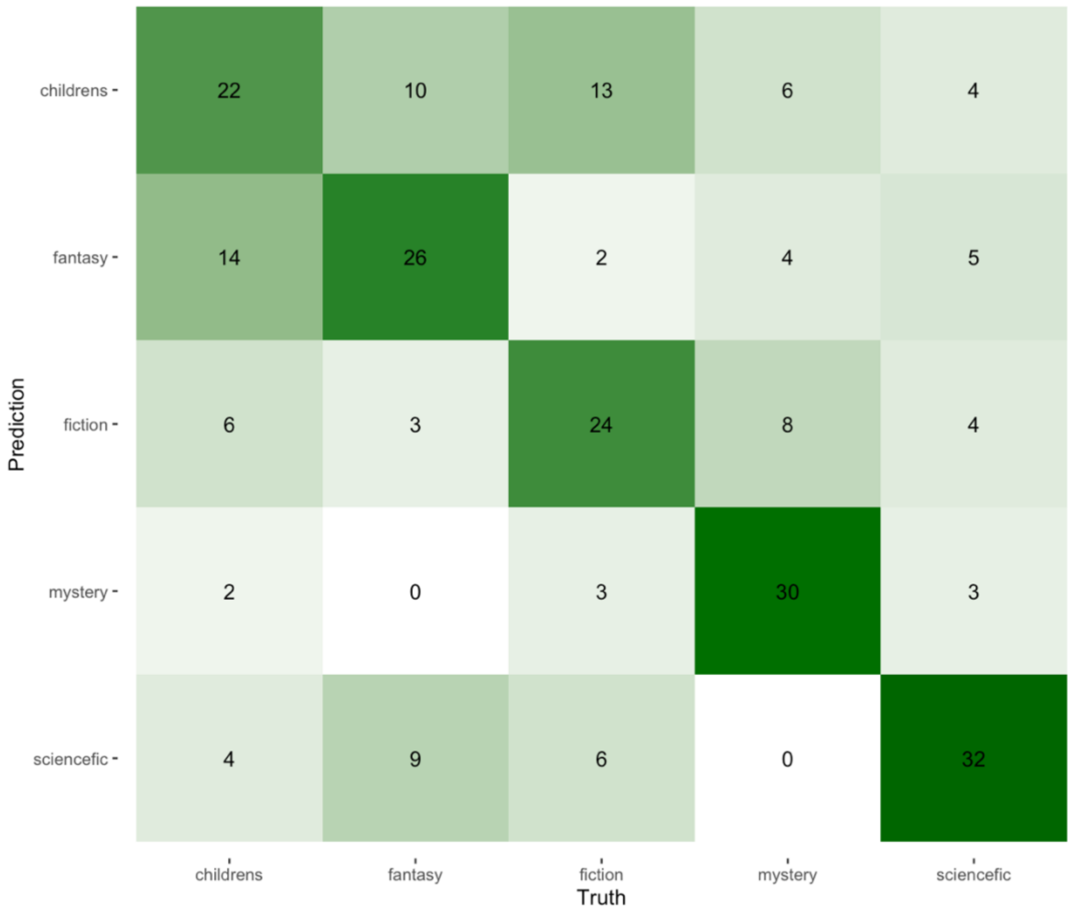


Plot 13 shows the new genres with plots to predict in the second attempt.

Second Results



```
# A tibble: 5 × 7
   penalty .metric  .estimator  mean     n std_err .config
     <dbl> <chr>    <chr>      <dbl> <int>   <dbl> <chr>
1 0.00785  accuracy multiclass 0.600    10 0.00952 Preprocessor1_Model16
2 0.0264   accuracy multiclass 0.581    10 0.0102  Preprocessor1_Model17
3 0.00234  accuracy multiclass 0.562    10 0.00919 Preprocessor1_Model15
4 0.000695 accuracy multiclass 0.525    10 0.0108  Preprocessor1_Model14
5 0.000207 accuracy multiclass 0.512    10 0.00939 Preprocessor1_Model13
```



Prediction \ Truth	childrens	fantasy	fiction	mystery	sciencefic
childrens	22	10	13	6	4
fantasy	14	26	2	4	5
fiction	6	3	24	8	4
mystery	2	0	3	30	3
sciencefic	4	9	6	0	32

In the second attempt with the significantly reduced data set, the model has a 60% average accuracy, so now the right predictions are at least better than the wrong ones. But the method of getting these results is questionable.

Other methods

I've included two of the other methods I tried in the bottom of the R script. But they had very low accuracy.

Project conclusion

To make the dataset into a multi-class I did some merging and removing of the genres that probably affected the results. Also, the text type "book summaries" might not be easy to predict as a pretend multiclass problem. Even though my results might not be too reliable, I got to try out a variety of methods in R when it comes to text analysis and classification.

References

Chuang, https://cynthiachuang.github.io/Difference-between-Multiclass-Multilabel-and-Multitask-Problem/

Couch, A, (10/19/2020) "*Tidy Tuesday Multiclass Classification*" https://github.com/andrew-couch/Tidy-Tuesday/blob/master/Season%201/Scripts/TidyTuesdayMulticlassClassification.Rmd

Devopedia. 2019. "Confusion Matrix." Version 6, August 20. Accessed 2021-09-09. https://devopedia.org/confusion-matrix

Hvitfeldt, E. and Silge, J. (2021). *Supervised Machine Learning for Text Analysis in R* https://smltar.com/mlclassification.html

Silge, J & Robinson, D. (2021). *Text Mining with R: A Tidy Approach*
(built by the bookdown R package and last built on 2021-09-02).
https://www.tidytextmining.com/index.html