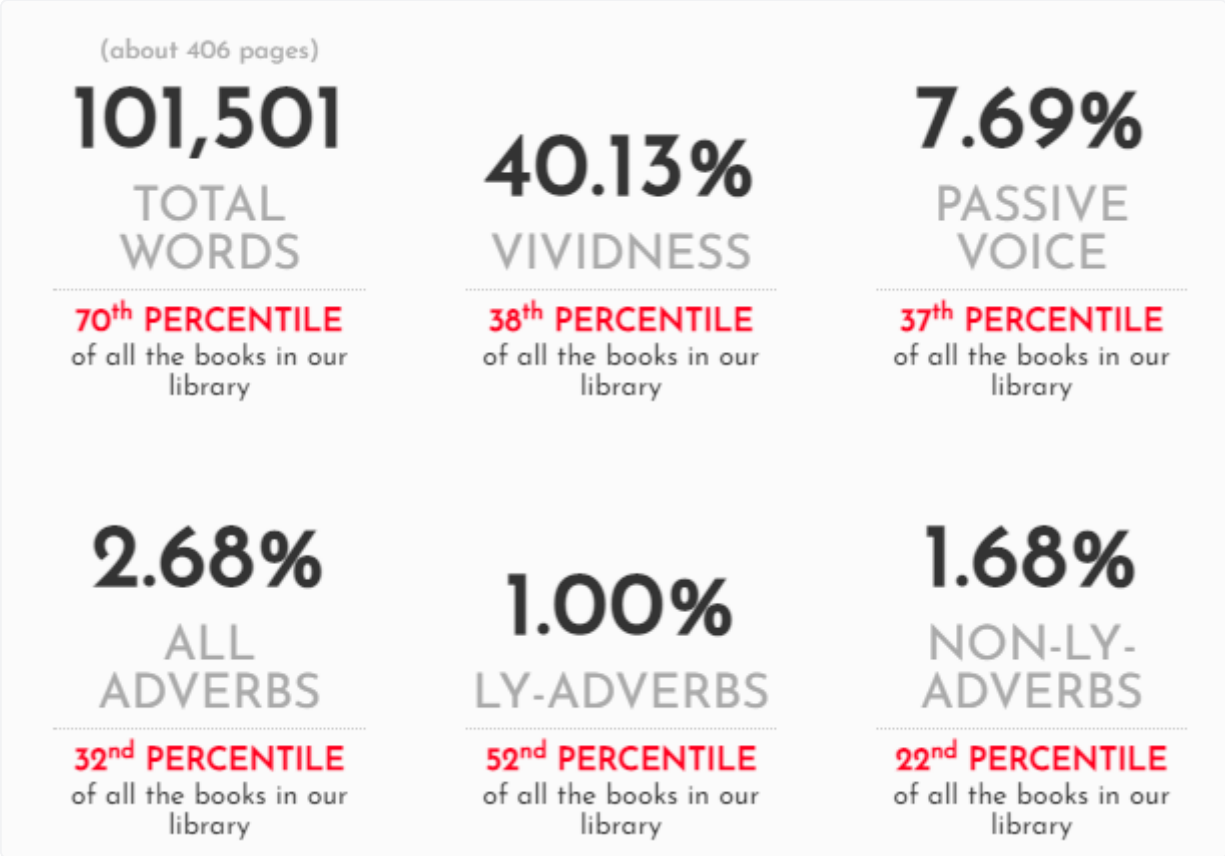
Genre Prediction from Linguistic Data

Problem Statement

In 2017, software engineer Benji Smith created a project called Prosecraft which has since been taken offline (see: Notes on Data Sources). The website, formerly hosted on prosecraft.io, contained linguistic analysis data on about 25,000 books on various features, including percentiles in comparison to the other books in the database.



I proposed to find out whether there was an association between these features and the genres of the books, and whether it could be used to predict the genre of a book.

This would be a multilabel classification problem, as each book may have multiple genres, and I wanted to predict as many as possible.

Data Wrangling

Initial data collection

The first goal was to get data from Prosecraft. In order to achieve this, I first found a list of every book in the library from the wordcount percentiles page, using the BeautifulSoup library. I saved this list as a .json.

Importing Data

Next, I created a function that, given a title and an author's name from the list, would turn clean and combine them into a URL, from which the linguistic data could be gathered onto a DataFrame.

Cleaning Prosecraft Data

For a bit more of a challenge, it was time to collect the genres. I obtained the genres from Amazon's Goodreads website, a collection of information and reviews on millions of books. I ran a search for each book and extracted the URL, when possible, from the cover image.

Get genres

I put every step together, extracted the list of genres from each book, and created a sample dataframe.

Test data collection

In the final dataframe, I included the publication year from Goodreads, with the hypothesis that would impact the classification of at least some of the genres. The search function would search Goodreads for the book title and scan the first page of results for an author match. If that failed, it would search the author's name and scan for a title match.

Create DataFrame

Filling the gaps

Before going on with my analysis, I searched for missing data points. First, I searched for missing Prosecraft features, and found 410 books. It turned out that any author or book with a special character in their name had not made it to the correct URL. This took a bit of trial and error, but I was eventually able to find every single book in the Prosecraft library.

Fix missing prosecraft

The missing data on Goodreads were a bit more of a challenge to deal with. I discovered that of the approximately 25,000 books in the library, 9.5% of the books had not been found on Goodreads. Those 2,400 or so, I would try to find. Another 8.7% had been found (indicated by the presence of a year on the dataframe) but had no genres listed. Since genre is the target feature, those 2,200 or so books had to be dropped.

As a first try, I searched past the first page of results for both title and author, but I later improved upon this strategy by searching for both the title and author at once, which worked both faster and with much greater success. In this updated version, I also tried removing middle initials, after discovering by observation that they often interfered with the search.

Fix missing Goodreads 1

After finding many more missing books, I dropped another 388 for which Goodreads had no genre data--17% of the newly found books. This is twice the missing genre rate, but it seems plausible that there would be a correlation between the ease of finding a book on Goodreads and the amount of data the site has.

An important breakthrough

I discovered that many of the missing values were based on simple typos on Prosecraft--small errors like Julie changing to Julia, or Georgia to Georgina. I contacted Prosecraft's owner, Benji Smith, and asked if he would like a list of the errors I found. It turned out that he was both grateful for my list and excited about my project. Excited enough that he shared an extra dataset--a list of the counts of each word in each book in the library! This would later come to change the course of my project.

In order to find the typos, I used the Levenshtein library to search for author's names that were off by a Levenshtein distance of 2 or less from what I had in my dataset. For instance, changing Julie to Julia changes one letter--a distance of 1--and adding an N to Georgia also represents a distance of 1.

With this, I was able to fix 202 of the 523 missing books, or 39%.

Fix missing Goodreads 2

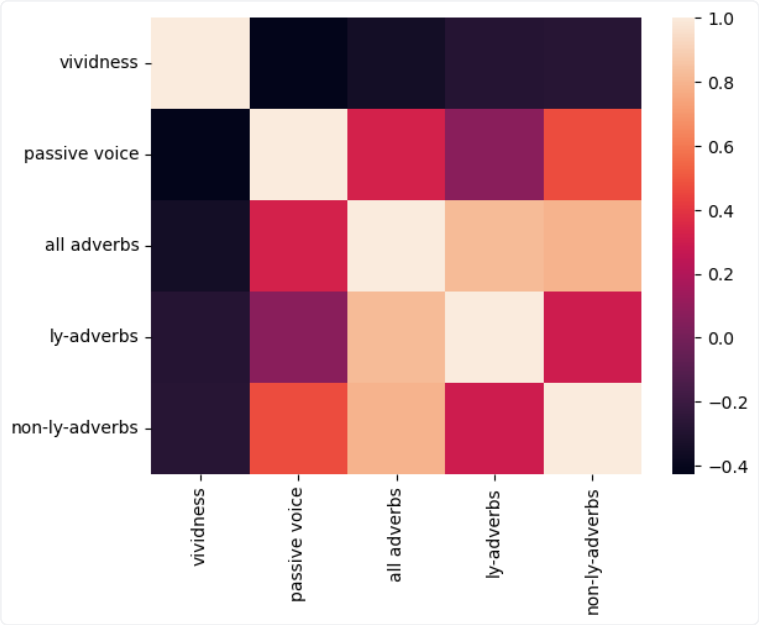
In one more notebook, I squeezed the last data I could out of Goodreads and compiled everything I'd found before beginning EDA.

A few more books

Exploratory Data Analysis

Preliminary EDA

The first thing I checked in my initial EDA was how correlated the linguistic data might be.



From this, it appeared that vividness was moderately negatively correlated with every other feature.

Unsurprisingly, "all adverbs" was highly correlated with both "ly adverbs" and "non-ly adverbs," which makes sense since it's just the sum of the other two. As a linear combination of two other features, it should likely be dropped.

Interestingly, it appears that passive voice is moderately correlated with non-ly adverbs.

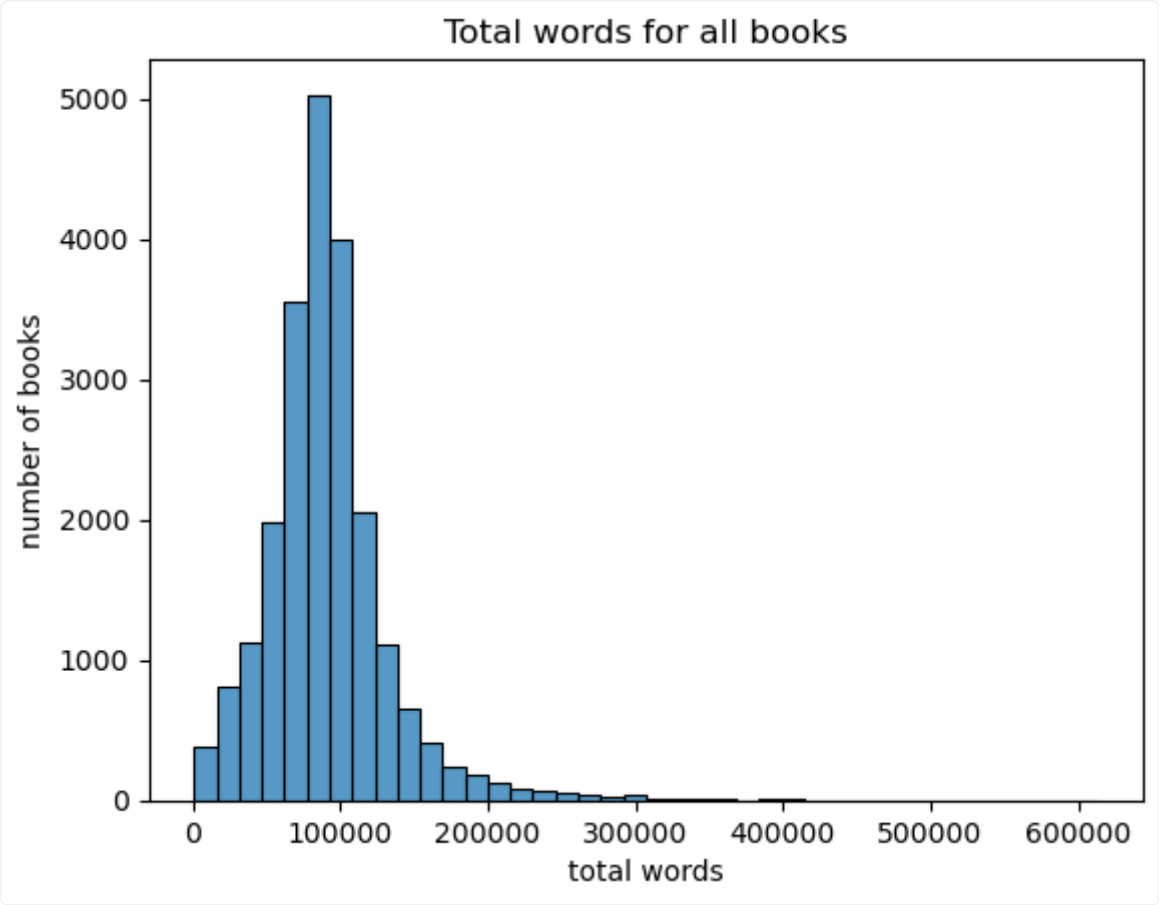
A heat map including the other two quantitative variables (total words and year) discovered no particular correlation with any other feature.

Although Github does not display the CSS for the numerical correlations with the background gradient, I've included a picture here.

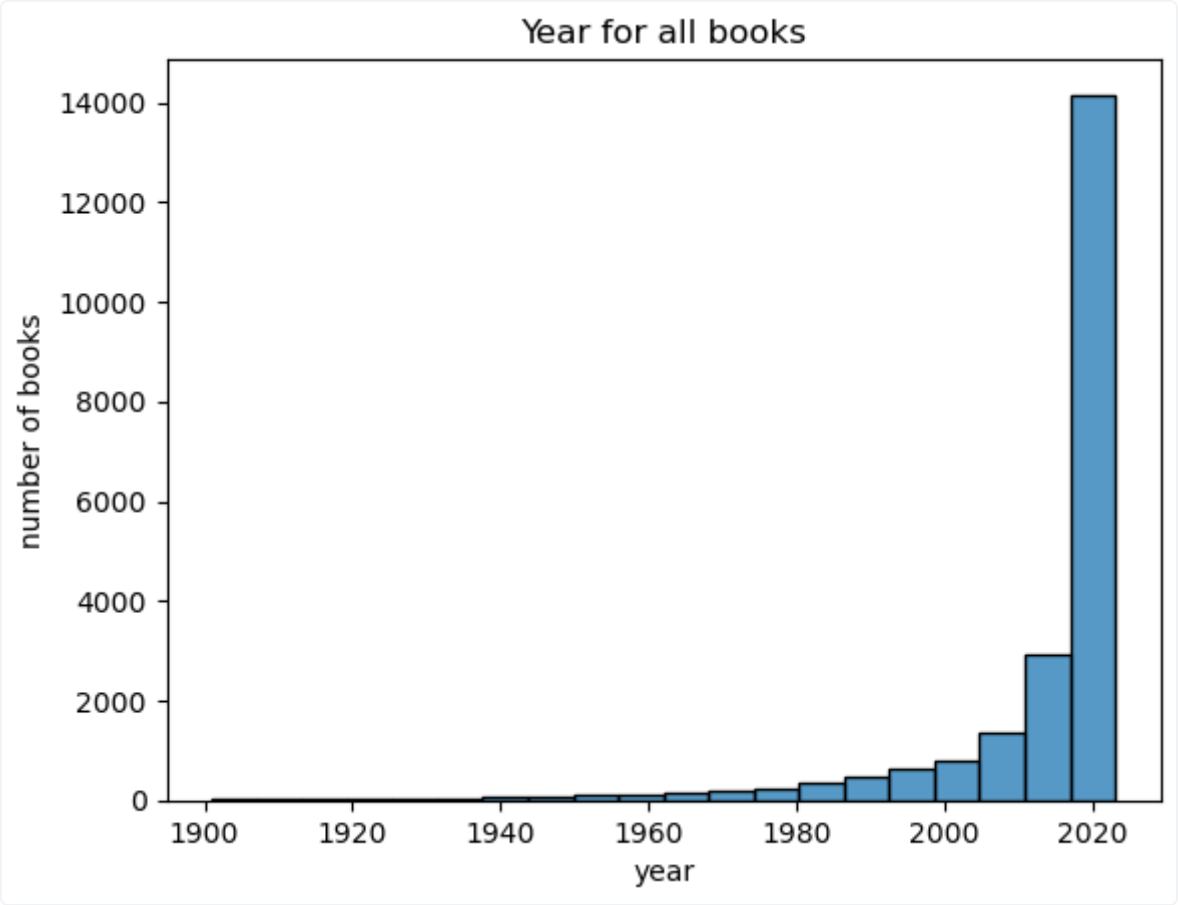
	total words	vividness	passive voice	all adverbs	ly-adverbs	non-ly-adverbs	year
total words	1.000000	-0.001394	-0.018643	0.004145	0.023499	-0.017973	0.011426
vividness	-0.001394	1.000000	-0.425165	-0.348603	-0.282485	-0.279284	0.058717
passive voice	-0.018643	-0.425165	1.000000	0.323722	0.064872	0.468020	0.056973
all adverbs	0.004145	-0.348603	0.323722	1.000000	0.817468	0.793055	-0.099414
ly-adverbs	0.023499	-0.282485	0.064872	0.817468	1.000000	0.297553	-0.036549
non-ly-adverbs	-0.017973	-0.279284	0.468020	0.793055	0.297553	1.000000	-0.126100
year	0.011426	0.058717	0.056973	-0.099414	-0.036549	-0.126100	1.000000

Most of the linguistic data was fairly symmetrical and approximately Normal, with few if any outliers.

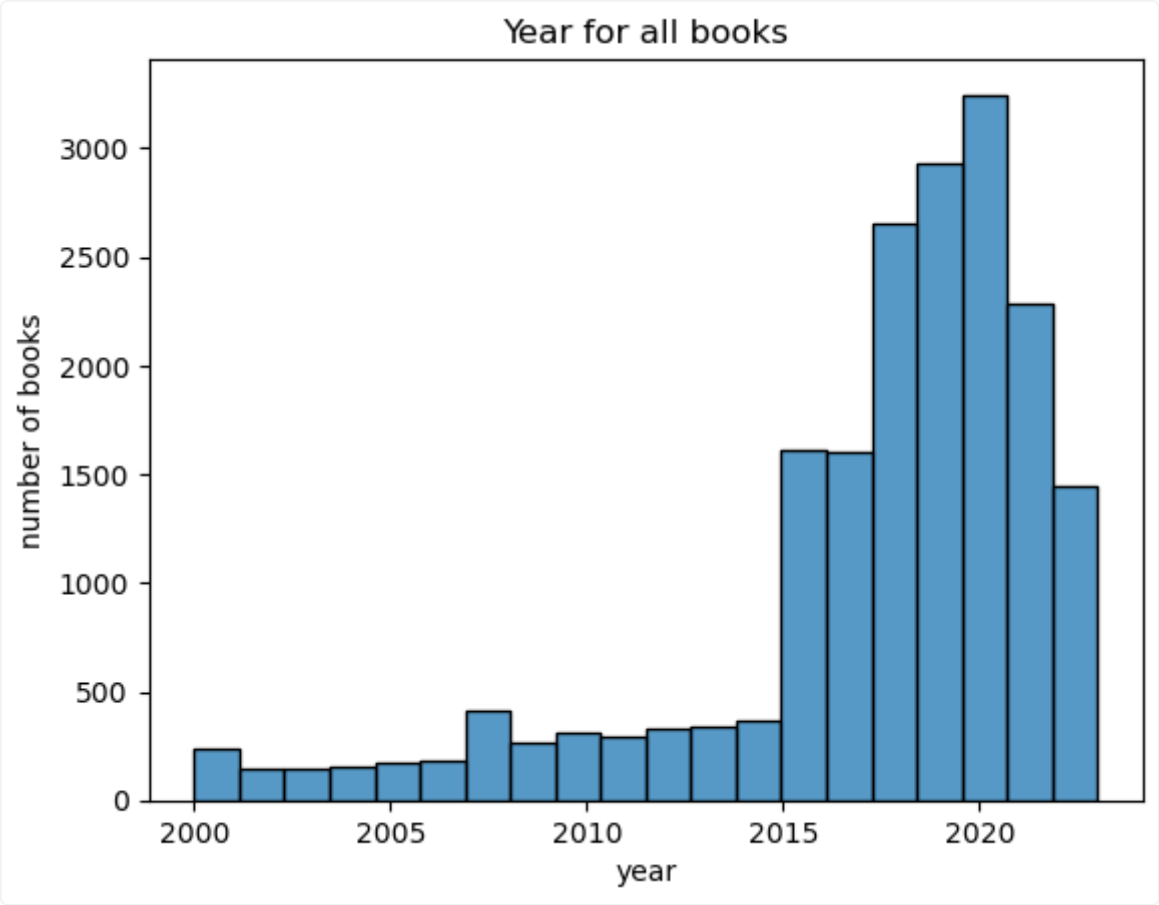
The wordcount, however, had a strong right skew.



The year had an even stronger left skew, with almost nothing but the 2000s visible until the outliers were removed. Only 1.2% of the data was before the year 1900, but outliers existed as far back as 180 AD, making the graph nearly impossible to read. Removing those, the years represented in the dataset still had a very heavy skew.



It can be seen here that most of the data come from close to the year 2020. Restricting the graph to books published in the 21st century, we can get a feel for just how recent most of these books are.



In fact, about 72% of the books in the dataset are from 2015 or later.

Finally, I checked the representation of different authors. It turns out that the mean author has about 1.8 books in this library, but the top 20 authors have more than 30 books each. The most-represented author is Franklin W. Dixon, with an incredible 146 books. As it turns out, he was the attributed writer of the Hardy Boys mysteries, although a bit of research shows that the books were ghostwritten, as one can also surmise by looking at the publication dates (nearly a century). Similarly, second most-represented author, Carolyn Keene, is the attributed writer of Nancy Drew.

Together, these two series of Children's Mystery Fiction account for a full 1% of the dataset! This does present some concern about bias, but 1% is likely not enough to have a serious impact.

Preliminary EDA

Analysis of target feature

Next, it was time to analyze the target feature! Unfortunately, Pandas DataFrames are not necessarily intended to hold lists in one column, so the actual datatype of the 'genre' feature is a string. Some research led me to an article by Max Hillsdorf with a suggestion for how to deal with this using the apply() and eval() functions

As it turns out, each book had an average of 6.0 genres.

Trouble with Goodreads genres

This was where things got a bit complicated: Since Goodreads "genres" are based on users' shelves, many of the tags are not what a reader or publisher would likely consider a genre. Elements like "Australia" or "Star Trek" show up occasionally, and "Audiobook"--which is a format, rather than a genre--is in the top 5 most frequent.

A quick check shows that in more than 22,000 books, we have 688 supposed "genres." However, 117 of the genre labels appear only once. These are categories such as "M M sports Romance," "Cthulhu Mythos," and "Green."

As it turns out, about half of the so-called genres appear in fewer than 10 books. Genres with exactly 10 books include "Birds," "Ukraine" and "Read for school," which are also not exactly genres.

Coming at it from the opposite direction, we find 27 genres with more than 1000 books. This is where we'll look for our target feature.

"Fiction" includes more than half of our dataset, which may or may not be a useful classification. "Mystery Thriller" is simply a combination of "Mystery" and "Thriller," which turned out to have more than 80% overlap with the other two genres, and a similar situation occurs with "Science Fiction Fantasy," so each of those should be split into their component parts.

Before doing so, I created a heatmap of the correlations between genres to see which seemed redundant.

index	Fiction	Mystery	Thriller	Fantasy	Audiobook	Mystery Thriller	Crime	Science Fiction	Contemporary	Romance	Nonfiction	Suspense	Historical Fiction	Adult	Young Adult	Historical	Horror	Adventure	Paranormal	History	Literary Fiction	Science Fiction Fantasy	Magic	Novels	Biography	Classics	LGBT
Fiction	1.000000	0.451045	0.358278	0.280345	0.286807	0.245797	0.217182	0.186919	0.197732	0.150722	0.000000	0.180919	0.174458	0.156458	0.137206	0.112547	0.089273	0.075427	0.042065	0.000000	0.078658	0.073976	0.049581	0.072130	0.000000	0.061779	0.048856
Mystery	0.872577	1.000000	0.628316	0.097577	0.290689	0.481378	0.433036	0.055867	0.139796	0.082908	0.004082	0.344770	0.135969	0.102041	0.081122	0.091709	0.086990	0.046939	0.048214	0.002551	0.020536	0.010332	0.014413	0.023087	0.001148	0.036480	0.019005
Thriller	0.895665	0.811933	1.000000	0.061645	0.320092	0.550849	0.453766	0.080930	0.128400	0.040547	0.002143	0.434811	0.064282	0.107796	0.047140	0.029998	0.116038	0.054557	0.027031	0.001648	0.013186	0.009890	0.001319	0.024065	0.000659	0.008241	0.017307
Fantasy	0.749912	0.134921	0.065961	1.000000	0.171781	0.012169	0.011111	0.337566	0.047443	0.231217	0.000000	0.010229	0.117969	0.149030	0.289947	0.072840	0.133510	0.118871	0.189065	0.000000	0.018695	0.156614	0.195414	0.027337	0.000000	0.047090	0.085714
Audiobook	0.794230	0.416104	0.354574	0.177835	1.000000	0.269673	0.219281	0.137667	0.204674	0.130729	0.171444	0.187877	0.114661	0.188424	0.067555	0.084170	0.051488	0.046376	0.024283	0.054044	0.072668	0.055505	0.035969	0.029578	0.080701	0.015337	0.025561
Mystery Thriller	0.947636	0.959329	0.849517	0.017539	0.375445	1.000000	0.524911	0.027453	0.136756	0.038638	0.000254	0.516014	0.063803	0.121505	0.051601	0.033045	0.064311	0.019319	0.014489	0.000254	0.012201	0.001017	0.000508	0.014743	0.000000	0.023386	0.014743
Crime	0.874900	0.901726	0.731208	0.016733	0.318991	0.548473	1.000000	0.016733	0.074635	0.028685	0.026029	0.367596	0.087649	0.052058	0.018592	0.060823	0.040372	0.020983	0.006906	0.015936	0.008234	0.003187	0.002125	0.020186	0.011155	0.030013	0.012483
Science Fiction	0.813953	0.125764	0.140970	0.549526	0.216480	0.031008	0.018088	1.000000	0.036750	0.078868	0.000000	0.032156	0.048808	0.111111	0.167959	0.018375	0.130060	0.109101	0.031582	0.000000	0.016078	0.241459	0.034166	0.058857	0.000000	0.059432	0.068045
Contemporary	0.901142	0.329327	0.234075	0.080829	0.336839	0.161659	0.084435	0.038462	1.000000	0.389724	0.009014	0.127704	0.093450	0.279748	0.134916	0.032452	0.029447	0.007512	0.029447	0.000601	0.245793	0.001803	0.005108	0.145433	0.003906	0.015024	0.075120
Romance	0.706865	0.200989	0.076067	0.405380	0.221398	0.047001	0.033395	0.084725	0.401051	1.000000	0.007112	0.094620	0.176252	0.220470	0.270563	0.142239	0.010513	0.025356	0.180581	0.000309	0.027211	0.011132	0.128942	0.022573	0.004020	0.018553	0.080396
Nonfiction	0.000000	0.010303	0.004185	0.000000	0.302318	0.000322	0.031552	0.000000	0.009659	0.007405	1.000000	0.000644	0.000000	0.027688	0.000000	0.036703	0.004829	0.024791	0.003220	0.383773	0.001932	0.000000	0.003542	0.000000	0.344173	0.025113	0.027688
Suspense	0.934287	0.920327	0.898195	0.019748	0.350358	0.691181	0.471229	0.038134	0.144705	0.104188	0.000661	1.000000	0.032346	0.110317	0.015322	0.010895	0.073885	0.040858	0.021110	0.000661	0.005107	0.004066	0.001362	0.012598	0.000340	0.002724	0.004767
Historical Fiction	0.907096	0.365444	0.133699	0.229345	0.215290	0.086047	0.113130	0.058279	0.106516	0.195406	0.000000	0.032568	1.000000	0.141584	0.104902	0.622215	0.046523	0.053137	0.030854	0.000000	0.137127	0.009256	0.019198	0.082962	0.000000	0.068564	0.047309
Adult	0.885778	0.298619	0.244121	0.315416	0.385218	0.178425	0.073162	0.144457	0.347518	0.266144	0.032102	0.120941	0.154152	1.000000	0.036954	0.119448	0.085106	0.028369	0.046286	0.008959	0.113475	0.003467	0.009724	0.035834	0.013438	0.004106	0.092572
Young Adult	0.818325	0.250098	0.112466	0.646481	0.145497	0.079827	0.027527	0.230043	0.176563	0.344082	0.000000	0.017696	0.120330	0.038930	1.000000	0.074322	0.084153	0.177350	0.104601	0.000000	0.009438	0.053480	0.169878	0.009044	0.000000	0.099882	0.116398
Historical	0.854782	0.360040	0.091137	0.206810	0.230846	0.065098	0.114672	0.032048	0.054081	0.230346	0.057086	0.016024	0.908963	0.160240	0.094642	1.000000	0.032549	0.048072	0.029044	0.056585	0.105158	0.005508	0.018027	0.043565	0.027541	0.038057	0.046570
Horror	0.848903	0.427596	0.441378	0.474608	0.176803	0.158621	0.095298	0.284013	0.061442	0.021317	0.009404	0.136050	0.085266	0.142947	0.134169	0.040752	1.000000	0.034483	0.203135	0.006270	0.027586	0.033229	0.015047	0.029467	0.003135	0.048903	0.048276
Adventure	0.851824	0.274013	0.246463	0.501862	0.189129	0.056590	0.058824	0.282949	0.018615	0.061057	0.057334	0.089352	0.115413	0.056590	0.335815	0.071482	0.040953	1.000000	0.014147	0.034996	0.004468	0.111690	0.098287	0.030529	0.029784	0.162323	0.009680
Paranormal	0.490350	0.290546	0.126057	0.823982	0.102229	0.043812	0.019985	0.084550	0.075327	0.448885	0.007686	0.047656	0.069178	0.095311	0.204458	0.044581	0.249039	0.014604	1.000000	0.003075	0.002306	0.014604	0.269792	0.003075	0.000769	0.004612	0.030746
History	0.000000	0.016502	0.008251	0.000000	0.244224	0.000825	0.049505	0.000000	0.001650	0.000825	0.983498	0.001650	0.000000	0.019802	0.000000	0.093234	0.008251	0.038779	0.003300	1.000000	0.001650	0.000000	0.002475	0.000000	0.339934	0.039604	0.010726
Literary Fiction	0.994167	0.134167	0.066667	0.088333	0.331667	0.040000	0.025833	0.046667	0.681667	0.073333	0.005000	0.012500	0.333333	0.253333	0.020000	0.175000	0.036667	0.005000	0.002500	0.001667	1.000000	0.003333	0.000000	0.349167	0.000833	0.072500	0.076667
Science Fiction Fantasy	0.944444	0.068182	0.050505	0.747475	0.255892	0.003367	0.010101	0.707912	0.005051	0.030303	0.000000	0.010101	0.022727	0.143098	0.114478	0.009259	0.044613	0.126263	0.015993	0.000000	0.003367	1.000000	0.147306	0.067340	0.000000	0.079125	0.039562
Magic	0.664898	0.099912	0.007073	0.979664	0.174182	0.001768	0.007073	0.105217	0.015031	0.368700	0.009726	0.003537	0.049514	0.141468	0.381963	0.031830	0.021220	0.116711	0.310345	0.002653	0.000000	0.154730	1.000000	0.003537	0.000000	0.003537	0.041556
Novels	0.990942	0.163949	0.132246	0.140399	0.146739	0.052536	0.068841	0.185688	0.438406	0.066123	0.000000	0.033514	0.219203	0.086957	0.020833	0.078804	0.042572	0.037138	0.003623	0.000000	0.379529	0.072464	0.003623	1.000000	0.000000	0.216486	0.022645
Biography	0.000000	0.008357	0.003714	0.000000	0.410399	0.000000	0.038997	0.000000	0.012071	0.012071	0.992572	0.000929	0.000000	0.033426	0.000000	0.051068	0.004643	0.037140	0.000929	0.382544	0.000929	0.000000	0.000000	0.000000	1.000000	0.028784	0.040854
Classics	0.885633	0.270321	0.047259	0.262363	0.079395	0.086957	0.106805	0.196552	0.047259	0.056711	0.073724	0.007561	0.189036	0.010397	0.240076	0.071834	0.073724	0.206049	0.005671	0.045369	0.082231	0.008847	0.003781	0.225898	0.029301	1.000000	0.011342
LGBT	0.734390	0.147671	0.104063	0.461665	0.138751	0.057483	0.046581	0.234886	0.247770	0.257681	0.085233	0.013875	0.136769	0.245788	0.293360	0.092170	0.076313	0.012884	0.039643	0.012884	0.091179	0.046581	0.046581	0.024777	0.043608	0.011893	1.000000

Each cell in this dataframe represents the percentage of the row genre that also belongs to the column genre. So the dark blue in the Fiction column means that each genre except Nonfiction, History, and Biography is made up of a significant amount of Fiction. The three exceptions have no overlap with fiction at all, which is promising for the validity of the data.

98% of Magic books fall under Fantasy, so Magic can likely be dropped.

When it came time to start dropping features, the first thing I did was to filter out every genre with less than 1000 books. This dropped the average number of genres per book from 6.0 to 4.0, and the median decreases from 7 to 4. 211 books have lost all of their genres, and must be dropped from the dataset.

Next, Mystery Thriller and Science Fiction Fantasy were broken up into their component genres.

Finally, I used my own domain knowledge and consulted with a group of avid readers to determine which of the remaining genres were not "genres." I removed 7 "genres," including formats like Audiobook, Novels, Fiction, Nonfiction, and Short Stories, and two that were deemed topics rather than genres--Magic and LGBT.

This disqualified a final 1532 books with no genres left, leaving 20490 books in our dataset.

Finally, the moment of truth: I checked the statistics for each genre to see if there was noticeable variation. The results were somewhat disappointing. For instance, mean vividness between genres ranged from about 43 to 51, but the standard deviations tended to be around 10, so the differences generally were not much more than could be explained by random chance.

Analyzing the target feature

The Pandas Profiling report from YData Profiling can be viewed here:
https://msem11.github.io/

Pre-processing

Adding wordcount data

With the current data seeming unlikely to be sufficiently predictive, I decided to go in a slightly new direction, adding in the wordcount data that Smith had sent me, with the condition that I not publish the raw data.

Now was also the time that my CPU became insufficiently powerful to handle the dataframes. After breaking the list of books into 20 chunks of 1000 and the remaining 490, each smaller dataframe had between 322,000 and 404,000 columns--many of which didn't match. Rare words like "zucar" and "émigré" were too numerous. After an attempt to switch from Pandas to Dask didn't help, I tried alternative methods.

Wordcount data

To reduce the size of the dataframes, I removed some of the wordcount features that were unlikely to be predictive. It seemed plausible that a word that only appeared one time in a book, or a word that appeared in only one book, would not help a model figure out the genre.

As it turned out, dropping words that only appear once in each book reduced the number of features in the first chunk from nearly 400 thousand to only about 163 thousand, better than halving the size. Cutting that down to words that appear in at least 5 books in each chunk of 1000 books reduced the number to only 45 thousand columns, decreasing the size of the dataframe by about a factor of 10 from the original.

However, trying to save these files as a CSV still resulted in a memory error.

Pandas without singletons

Size reduction

Finally, I moved from Jupyter to Google Colab, where I was able to take advantage of cloud computing, which was capable of holding my dataframe in memory and to turn it into a parquet. The combined dataframe turned out to have just over 66 thousand columns, many of which were null in for all but the minimum of 5 books, and used over 10.8 GB of memory.

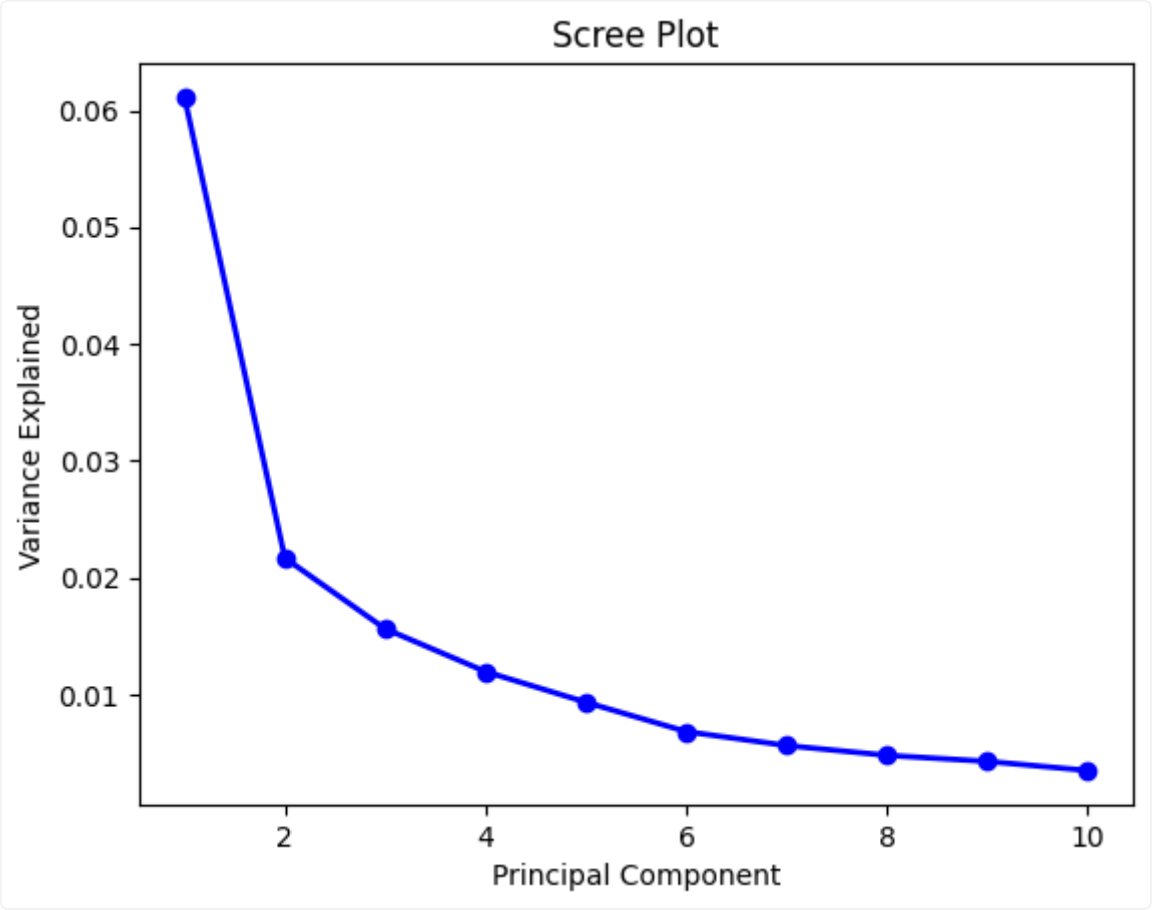
While I was here, I created a dataframe of each book's genres, with a 1 for genres that applied to the book and a 0 elsewhere.

Smaller Dataframes to Parquet

I scaled the different types of data separately. The linguistic data and meta features, such as passive voice, total words, and year, I scaled with a Standard Scaler. For the wordcount data, which was nearly all 0 or 2 (since I'd taken out words that only appeared once), with larger values for more common words, I used a minmax scaler. Unfortunately, when I tried to use Cross-Validation to fit a K-nearest-neighbors model for each label, ever fit failed, and the ValueError indicated that my model may be misconfigured.

I had planned to try PCA regardless, but now I was relying on it. I decided to use PCA only on the wordcount data, to keep the linguistic data separate and interpretable.

On my first attempt, with 10 components, the Scree plot was not encouraging.



No single feature explained more than 6% of the variance, and all 10 together explained only 14%.

Trying 100 components instead only brought the sum up to about 23%.

I noted that the number of features was about 4 times the number of samples in my training dataset, and that 94% of the cells were 0. I decided to try SparsePCA.

Unfortunately, SparsePCA does not have a native explained_variance_ratio_ attribute, so I was unable to create a scree plot, but when I tried to fit a model to the data, the results indicated a moderate success.

Scaling and PCA

Modeling and Evaluation

First model

At last, I was ready to create a model. I decided to use a binary relevance (one-vs-rest) classification, since the genres had no particular ordering, and could come in nearly any combination.

For my first model, where I was just testing the viability of my transformations, I tried an untuned Random Forest Classifier. I created a separate classifier for each genre, fit them to the data, and observed the results.

My first results were somewhat encouraging, with a Micro F1 of 0.63 and a macro F1 of 0.53. Unfortunately, upon examining the data, I realized I'd forgotten to remove "num genres" as a predictive feature. The number of genres was a feature derived from the target feature, and as such should not have been able to be used for prediction. I quickly removed it and reran the model, with a slight decrease in predictive ability.

```
Hamming Loss: 0.0953
F1 Score (Micro): 0.5814
F1 Score (Macro): 0.4745
```

The results from genre to genre varied, with precision varying from 0.69 for Crime to 0.93 for Adventure (a micro average of 0.78 across all genres), and recall from only 0.04 for Horror to 0.74 for Mystery (an average of 0.46). This indicates that, for instance, only 4% of the actual Horror books were classified correctly, and 93% of books classified as Adventure actually were.

Model Creation

Alternative models

Now, it was time to create more models, and see how well a classifier could do.

First, I tuned each of my Random Forest classifiers to their respective genres. I used a RandomSearchCV with a sample space of 360 possible hyperparameter combinations, of which I tried 10. On average, the best model for each genre tended to have no bootstrapping, a max depth of 30, a minimum of 2 samples per leaf, and 150 estimators. Each of these optimal hyperparameters had been in the middle of the range I'd tried, so I was reasonably satisfied that I chose my sample space well. Here are the full results from the tuned Random Forest models:

	precision	recall	f1-score	support
Mystery	0.77	0.74	0.76	1585
Thriller	0.74	0.69	0.72	1323
Fantasy	0.82	0.66	0.73	1181
Science Fiction	0.74	0.42	0.54	726
Crime	0.68	0.42	0.52	729
Contemporary	0.72	0.35	0.47	702
Romance	0.77	0.38	0.51	642
Suspense	0.76	0.23	0.35	610
Young Adult	0.80	0.30	0.43	502
Historical	0.79	0.38	0.51	599
Horror	0.84	0.07	0.13	298
Adventure	0.90	0.20	0.33	255
Paranormal	0.84	0.08	0.14	273
History	0.92	0.74	0.82	250
Literary Fiction	0.78	0.07	0.12	267
Biography	0.74	0.45	0.56	210
Classics	0.83	0.68	0.75	176
Memoir	0.74	0.44	0.55	192
micro avg	0.77	0.48	0.59	10520
macro avg	0.79	0.41	0.50	10520
weighted avg	0.77	0.48	0.56	10520
samples avg	0.65	0.51	0.54	10520

This tuning did not significantly improve upon the evaluation metrics, with the Hamming Loss decreasing by only 0.001 and each of the F1 scores increasing by about 0.01. The worst recall score increased from 0.04 to 0.08 (Horror and Paranormal), and the best recall score did not change. The worst Precision score very slightly decreased, from 0.69 to 0.68 (Crime), as did the best, from 0.93 to 0.89 (Adventure), with History becoming the new highest at 0.91. All of these changes are likely within the realm of random chance.

Just to see if it would help, I tried another Random Search that checked 20 possibilities from the sample space instead of 10. This resulted in no change at all.

Finally, I tried an XG Boost Classifier. I created a sample space of over a million hyperparameter combinations and tried a RandomSearchCV on 50 of them.

	precision	recall	f1-score	support
Mystery	0.70	0.88	0.78	1585
Thriller	0.66	0.85	0.74	1323
Fantasy	0.72	0.76	0.74	1181
Science Fiction	0.56	0.68	0.61	726
Crime	0.52	0.73	0.61	729
Contemporary	0.56	0.65	0.60	702
Romance	0.59	0.66	0.62	642
Suspense	0.48	0.66	0.56	610
Young Adult	0.61	0.55	0.58	502
Historical	0.56	0.61	0.58	599
Horror	0.36	0.32	0.34	298
Adventure	0.46	0.36	0.40	255
Paranormal	0.45	0.27	0.34	273
History	0.84	0.81	0.83	250
Literary Fiction	0.48	0.40	0.43	267
Biography	0.60	0.68	0.64	210
Classics	0.77	0.80	0.78	176
Memoir	0.71	0.68	0.70	192
micro avg	0.61	0.70	0.65	10520
macro avg	0.59	0.63	0.60	10520
weighted avg	0.61	0.70	0.65	10520
samples avg	0.62	0.71	0.63	10520

The XGB model created was a decided improvement. Although the Hamming Loss increased by about 0.01, the F1 scores increased by 0.07 for Micro and a full 0.13 for Macro.

Looking at a complete classification report, one can see a marked improvement in recall, with the worst genre (Paranormal) now having 0.27 instead of 0.08. The best performer (Mystery) has also increased from 0.74 to 0.88. There is a corresponding loss of precision, with Horror in particular going from 0.84 to 0.36 (now the lowest), but its F1-score has nearly tripled (0.13 to 0.34), so it seems like a more than fair tradeoff.

Feature importance

Satisfied with the model, I decided to check the importance of each feature for each model.

According to the Random Forest model, a PCA feature was the most important in every genre except Adventure and Classics, where 'publication year' lead the way. In fact, for Classics, publication year had an importance of 0.48, which makes sense intuitively.

Since PCA features are difficult to interpret practically, I decided to check the most significant non-PCA features--the ones that explained more than 4% of the variance in each model. The two models generally chose similar features, with some small differences. The output can be seen in the notebook linked below.

Finally, to satisfy my own curiosity, I checked the z-score for the mean of each genre vs the mean of the full collection, for each of the features XG Boost deemed significant. A z-score indicates how many standard deviations above the population mean each genre's mean falls. For instance, a z-score of 1.0 indicates that the mean of a given genre is one standard deviation above the mean for all books, indicating that books from that genre, on average, tend to be higher than about 84% of books from other genres, using Normal distribution calculations.

There were a few surprises, such as use of passive voice being deemed significant for Mystery and Thrillers, despite a z-score of only 0.03, as well as publication year for Historical (z = -0.03), and total words for Memoir (z = -0.04).

There were also some that made quite a bit of sense, like History's use of passive voice having a z-score of -1.56--in historical nonfiction, it makes sense that people do things rather than things being done by people. The publication year for Classics (the only significant non-PCA feature) having a z-score of -1.38 is also to be expected, as being old is a large part of what makes them classics.

Some that I didn't necessarily expect was the total words for Historical having a z-score of 1.57, and Adventure 1.62--each genre has, on average, nearly twice the mean words. Romance's total words having z = 1.11 compared to Fantasy's z = 0.96 was definitely not something I'd have predicted, as I tend to picture Fantasy novels as sweeping sagas and Romance novels as lighter, faster reads.

Model comparison

Opportunities for improvement

If I had to repeat the process, there are quite a few things I'd do differently. Because I completed this project over the course of a Bootcamp curriculum, there were strategies I learned later that would have helped me earlier, such as using a separate .py file rather than copy/pasting functions across notebooks.

One error I discovered after everything was over was that after deciding to remove "all adverbs" as it was a linear combination of "ly adverbs" and "non-ly adverbs," I seem to have missed the step of actually removing it. It did, in fact, end up up in my model. This multicollinearity is unlikely to have directly impacted the predictions, and I was not looking for specific coefficients for each feature, so I elected not to restart the model creation as I did when I discovered that I'd forgotten to drop "num genres."

Some potential source of bias come from the selection of books in the Prosecraft library, which was a distinctly non-random selection of books that favored newer books. In addition, having to drop the books without a genre list on Goodreads likely biased the remaining dataset toward more popular books. This isn't necessarily a bad thing, if we consider that the population for which we can generalize our result is made up mostly of popular books.

The high performance of the Mystery genre for both precision and recall could potentially be attributed to the previously mentioned 1% of the dataset being Hardy Boys or Nancy Drew novels, which likely have some similarities despite not being written by the same author.

A note on the data sources

Unfortunately, near the end of my completion of this project, the main source of data, Prosecraft, came under heavy criticism from authors. Allegations included potentially unethically-sourced data, and a lack of permission from the authors for the inclusion of their own books. This criticism caused Smith to immediately shut down the website, so Prosecraft.io no longer exists.

I hope that the use of the data for a one-time personal project which I do not intend to publish or profit from is considered a fair use. I am committed to data ethics, and am using this dataset for educational purposes only. Because of the concerns about the data collection, I will be unable to share the dataset. Instead, after the completion of this project, I will be purging it from my own computer as Smith has already done.

One source of potential bias in the model is that Goodreads shelves are user-generated and often contain inaccuracies. For instance, some books are shelved as both "Young Adult" and "Adult," and some users' opinions of a genre or category do not agree with the publisher's or author's intention. It has been noted in particular that adult fantasy novels written by women are disproportionately misclassified as Young Adult.

Disparities in user-generated shelves was noted in the previous section on The Trouble with Goodreads Genres, but in the absence of a larger or more reliable listing of books and their genres, to a great extent I had to rely on them anyway, placing some trust in the so-called "Wisdom of the crowd." In other words, I hoped that with enough diverse opinions, the inevitable errors would average out to "close enough."

Acknowledgements

I'd like to thank my mentor at Springboard, Chris Esposo, for his help with both the general ideas and specific implementation for this project.

Thanks also to Benji Smith for the original dataset, his encouragement and enthusiasm, and the gift of the extra data which so improved my model.