Unsupervised Book Genre Classification
Evan Hintz

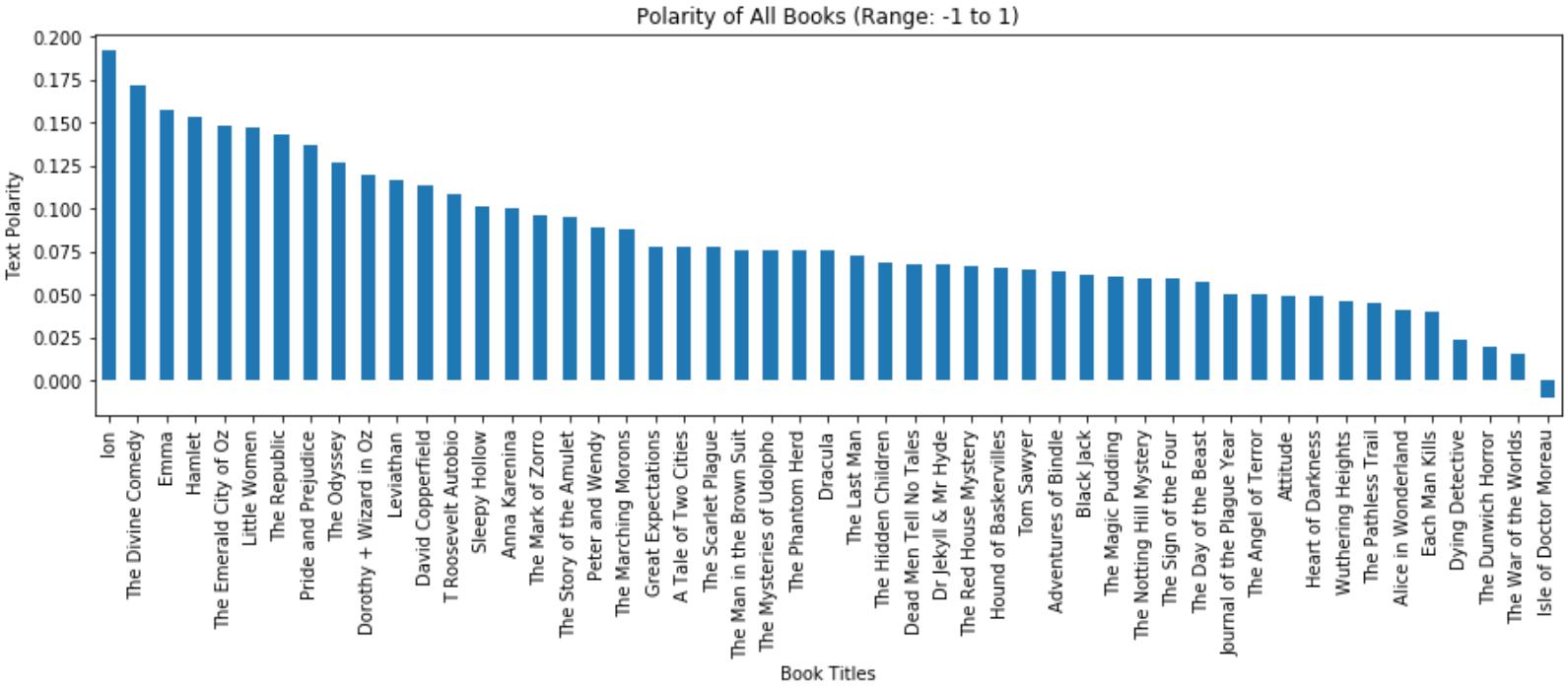
This project will focus on unsupervised machine learning models to classify various book texts into unlabeled genres. Since book genres already exist I'm very curious to see the patterns that a machine learning model will pull out of texts as opposed to attempting to classify them into pre-existing themes. The end goal is to be able to offer companies a new and potentially improved scheme for labeling books which will aid in consumer recommendation systems. Theoretically, an unsupervised model will be able to see patterns that may not be included in the traditional book genres helping suggest books to consumers that may not occur to an expert.

The data used will be collected from Project Gutenberg, an online library of open-source texts (expired copyrights, or granted rights). Generally, this means that the texts will be significantly older. A selection of various genres will be used although this is restricted to the books available.

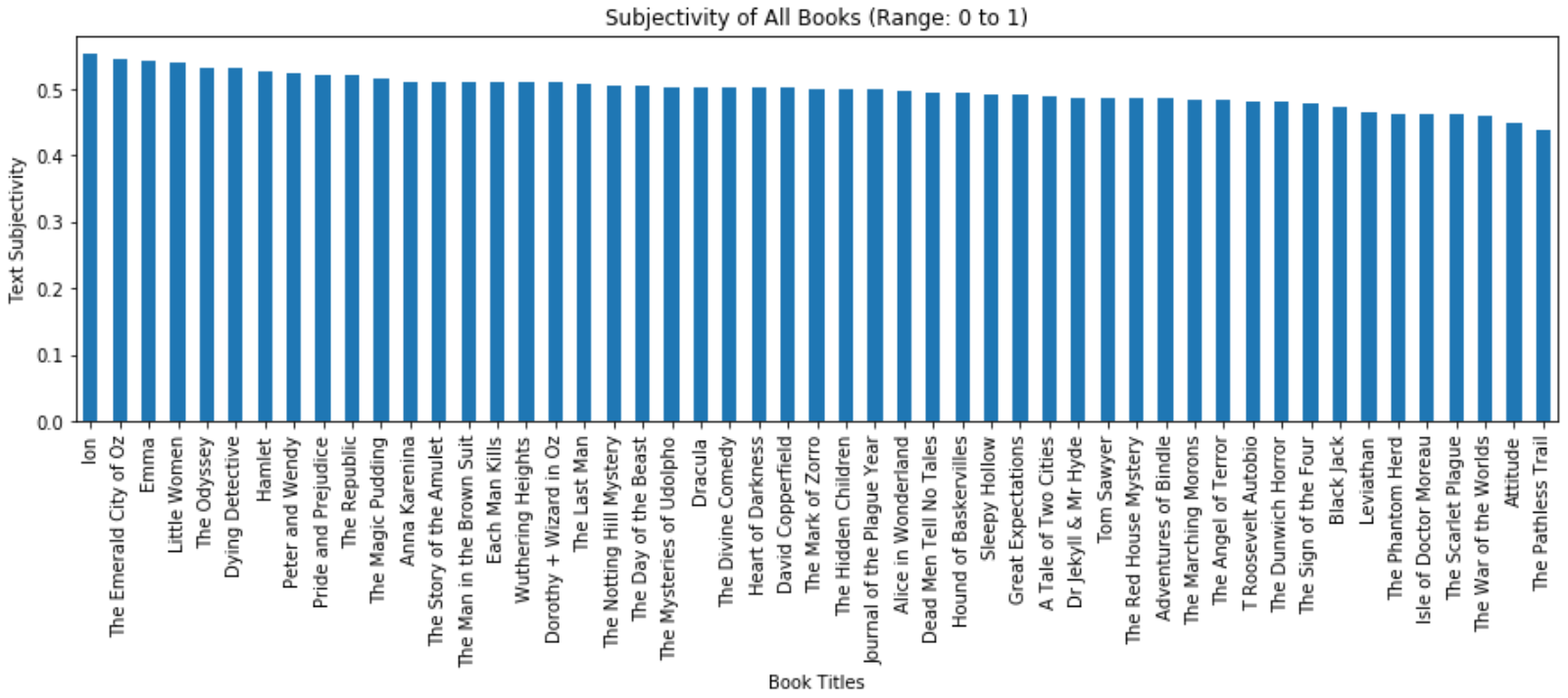
First the data must be gathered from Project Gutenberg. In this current iteration 101 hand-picked books are being used, mostly chosen from the website's "Top 100" books. 78 will be used for training and the remaining 23 for testing.

The exploratory dive into the data will only include 50 of the books. The initial view is using sentiment analysis which includes looking at the raw polarity and subjectivity of each book.

Text polarity attempts classify the emotions expressed within the text as generally positive, neutral, or negative. It returns a number between -1 and 1. The following graph displays all book's polarity. It is interesting to note that even the lowest value (The Island of Doctor Moreau) is hardly below 0 and the highest (Ion) is still quite far from 1.



Text subjectivity analysis tries to give a numerical value to describe how objective or subjective the text is. In other words, did the text consist of factual statements (objective) or more emotional and opinionated (subjective). This scale is between 0 and 1, 1 being the most subjective or emotional. As you can see in the below graph there isn't much range in the subjectivity of the books; almost all books are just about at the midway point - although non-fiction texts will generally receive lower scores.



Word clouds are a great way to see the main focus on any texts. This first cloud includes all words from all books. Not surprisingly the most used words are all very common and generic. These are the type of stop words that will be removed later to help the machine learning models focus on more book-defining words.

The following clouds are of a few of the books with the more extreme statistics from the other graphs shown. These have much more specific words, many of which are relevant to their genres and will be helpful during the actual classification process.

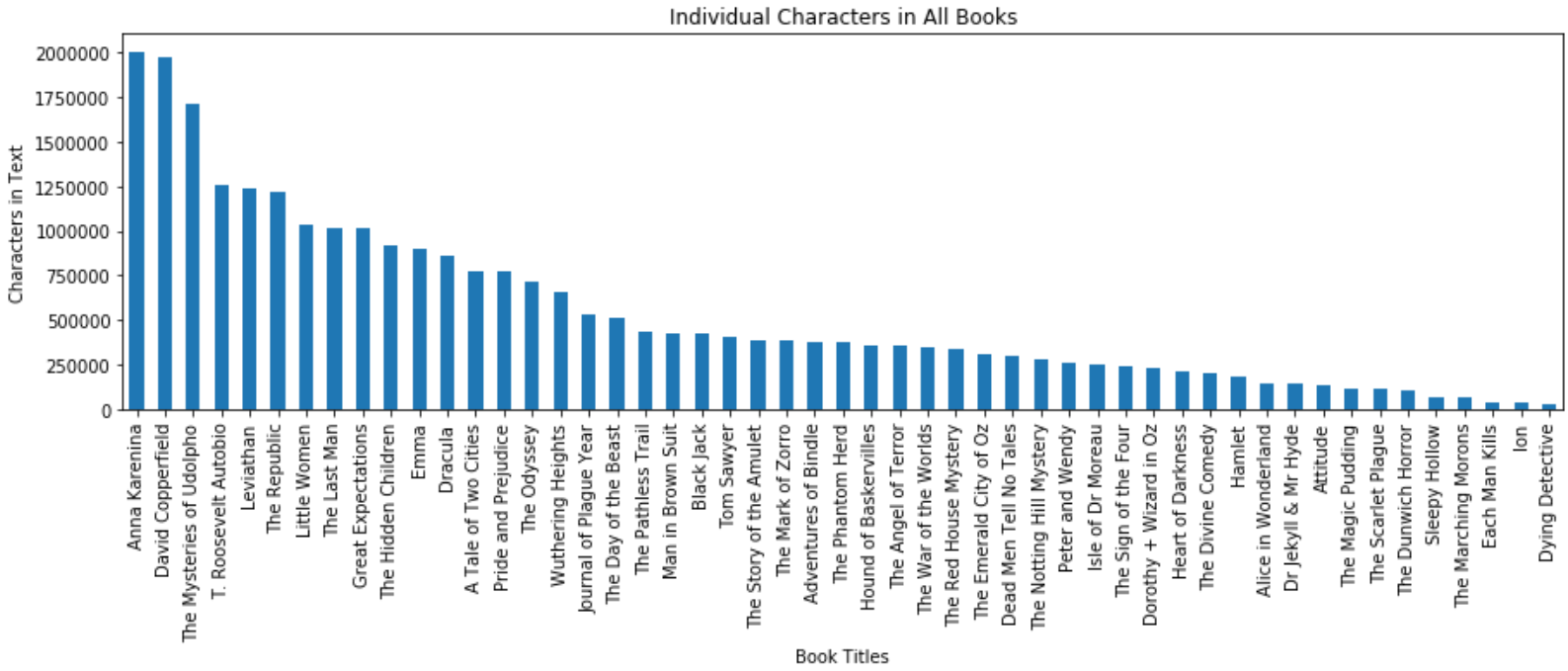
Isle of Dr Moreau by Wells, H. G. (Herbert George)

Ion by Plato

Anna Karenina by Tolstoy, Leo, graf

Dying Detective by Doyle, Arthur Conan

Understanding the differences in books is important. Using the length of characters in each it is easy to compare the length of books and the range of sizes. The following graph shows the huge variance from a short story - The Adventure of the Dying Detective to the expanded novel of Anna Karenina by Leo Tolstoy.



The next step is to clean and prepare the texts for the machine learning models. In natural language processing there are options for a lot of different types of wrangling. Depending how the text is being used they can contain a lot of noise and unwanted variety. This can include uppercase letters, punctuation, as well as different forms of words like plural,or past-tense. Texts also have a lot of words that don't necessarily add any substance to classifying models; these are known as 'stop-words'. Stop-words include 'the', 'and', 'a', and usually changes for different uses. These are usually just removed from texts to take out the noise and increase processing speeds.

The case of letters can create issues as Python will read "Snakes" as a different word from "snakes" however this can also change the meaning of words; for example with acronyms like "US" as a country versus "us". 'Immigrate to the US with us!' The simplest solution is to bring all the letters to lowercase and accept the hopefully few inaccuracies that will occur.

Punctuation will also have a large effect on the meaning of a sentence and this is another topic that machine learning models will struggle with and will often have negative outcomes on predictability. Following general practice, punctuation will be completely removed.

After the quick removals, the books are tokenized. This process separates the text as a single string into individual words that will be easier to process and understand for the algorithms. Another technique is the use of n-grams which are strings of n words that often improve analysis.

Example: "I'm not surprised!" tokenized as ('im'), ('not'), ('surprised')
or a 2-gram: ('im'), ('not surprised')

Each of these can have very different effects when used as input to a machine learning algorithm. n-grams may be implemented later to compare.

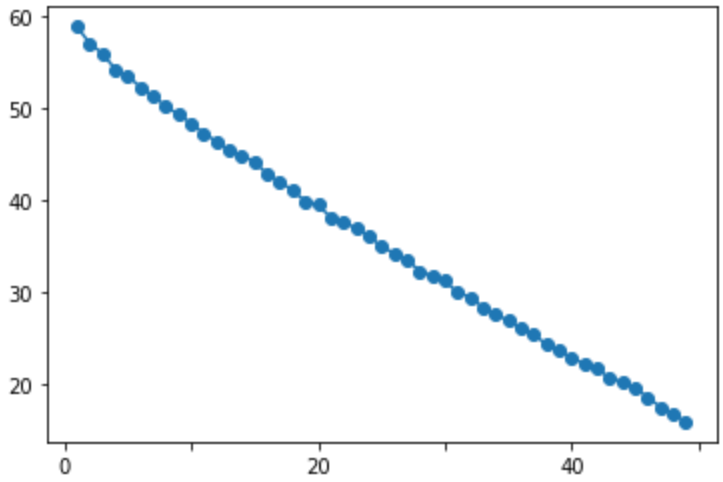
Since all the words have been individually wrapped into their own strings it is now easier to remove any stop-words that are present to reduce noise and increase processing speeds.

Next, words are tagged with their part of speech. This step is necessary in order for the following lemmatization to occur properly. This will help the model better understand the words and their context within the book. Each word within the texts will be tagged with the correlating part of speech: verb = 'VB', noun = 'NN', etc.

Lemmatizing is a form of standardizing or normalizing the text so that it becomes easier for the models to compare between books. Since this project is focused on unsupervised machine learning to classify genres, language and context is important between texts so lemmatization was chosen over stemming. Stemming is generally faster but will oftentimes convert a word to a stem that is not an actual word. Whereas lemmatizing brings the word to its root while keeping it a full word and it does this using a corpus (a collection of words) which makes it significantly slower. While speed is not currently a concern at this point, the use of stemming may be explored later on if this is applied to a larger database. Lemmatizing also provides more readability if necessary while working through this project.

Before the data can be plugged into any machine learning models it first needs to be converted into a format that can be better understood by the algorithm. One way to do this is by vectorizing the data. This was done using the term frequency inverse document frequency (tf idf) vectorizor. It evaluates how relevant the words are to the books based on how often the word shows up and the inverse document frequency of the word across all books. If the word appears a lot in one book it gains importance, but if it shows up in all the other books as well, it loses value in ability to define a single book. Many of these words were already removed as stop words but the remaining are assigned numerical values as vectors. Following vectorizing, the work can transition to machine learning models to start classifying each book using KMeans and Agglomerative Hierarchical Clustering with a quick look at Non-Negative Matrix Factorization before comparing accuracies.

KMeans is one of the simplest, most common, and often most accurate clustering models. This chooses k (number of clusters) random positions and assigns a cluster center for each. The data is split between these by assigning the data point (vectors in this case) to the nearest cluster center or centroid. Decision boundaries are set for each centroid. The centroid is then updated to be in the middle of the data within the decision boundary. This is repeated until clusters are fully centered around each centroid (or as close as possible). The following graphs help decide on the optimal number of clusters. A general rule is around the square root of the number of data points, which in this case is ~10.

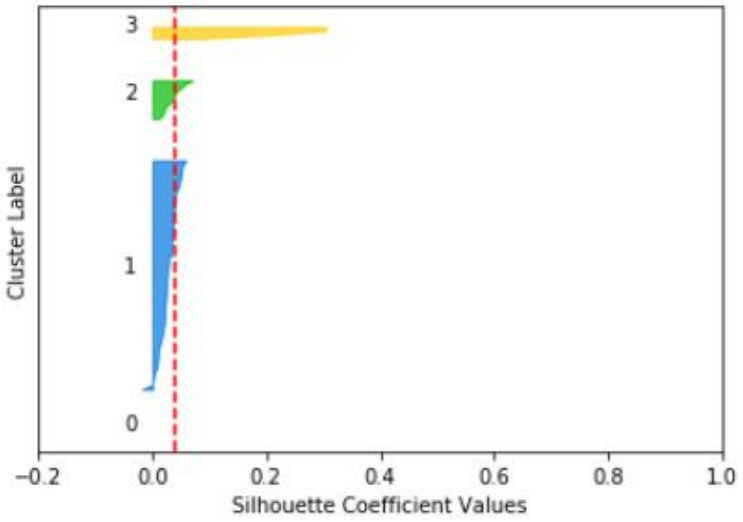


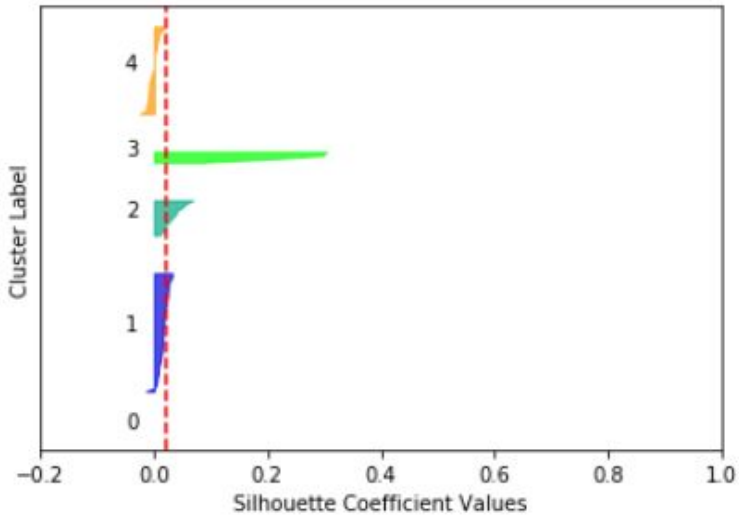
This graph shows the variance of the data explained within the model at each number of clusters from 1 to 50. The goal is to find a turning point where the benefit of a marginal cluster is much lower. In this graph there is no real ‘elbow’ so it is not very useful. One of the biggest dangers during this process is overfitting the model to where it works well for the training data but won’t be able to be successfully applied to any other sets of data.

Another potentially helpful way to choose the number of clusters is by calculating the silhouette scores. This value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The silhouette ranges from −1 to +1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters. If most objects have a high value, then the clustering configuration is appropriate. If many

points have a low or negative value, then the clustering configuration may have too many or too few clusters. Below are the scores for a range of clusters and two silhouette plots for the two highest scores. They are indicating the optimal number of clusters will be around 4 or 5.

```
Silhouette Score for n = 4: 0.03788939626377465
Silhouette Score for n = 5: 0.022442601533582063
Silhouette Score for n = 6: 0.006359777498229408
Silhouette Score for n = 7: 0.0034744797494738336
Silhouette Score for n = 8: 0.015660054903741776
Silhouette Score for n = 9: 0.0071181863665264085
```





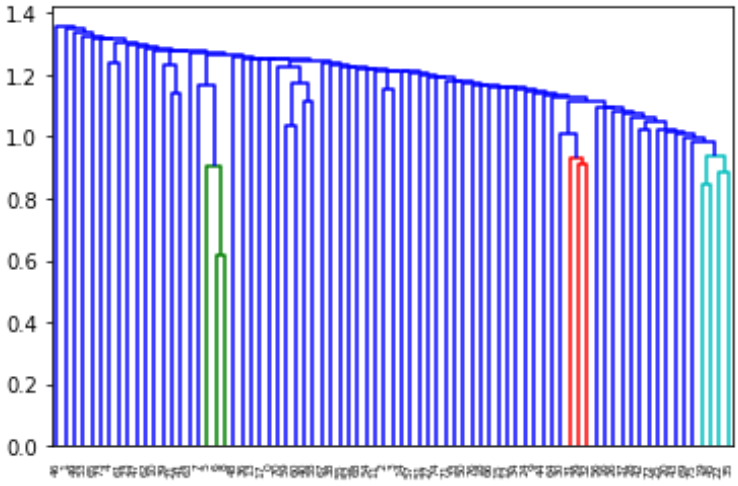
Cluster 0:	Cluster 1:	Cluster 2:	Cluster 3:	Cluster 4:
peter	one	dorothy	señor	say
wendy	say	scarecrow	zorro	go
mcgregor	make	oz	diego	one
mr	would	say	señorita	mr
say	come	wizard	gonzales	come

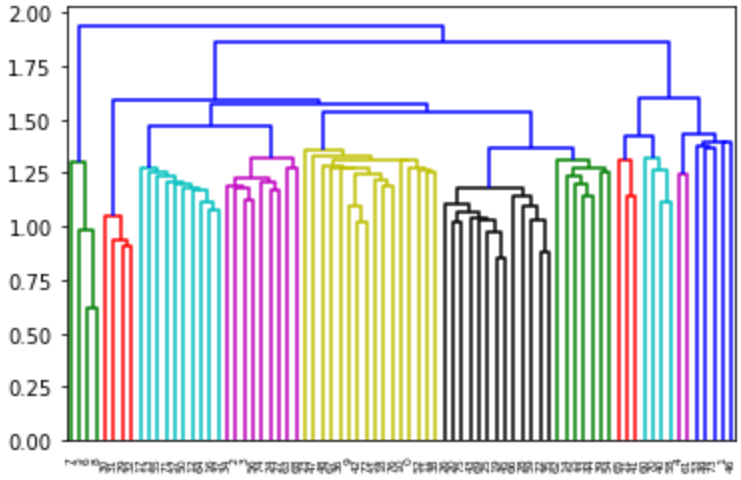
Here we can see the top words defining each cluster. Looking at this it might be worth it to increase the vocabulary of the stop words list; such as "come" and "would". It also looks like there were enough L. Frank Baum books to have their own Oz land cluster (5) along with cluster 4 for Sherlock and Watson's mysteries.Since this is unsupervised learning there is no metric to measure by except our own knowledge and understanding. To the right is a selection of test books along with their predicted cluster - we can compare these predictions to the words and labels from the fitted model on the training set of books. We can see Ozma of Oz fits right where we'd expect it to. Cluster 4 seems to dominate even in this slice of the test books, this is most likely due to the generic nature of the words that make up the cluster.

```
Prediction
4 Sense and Sensibility
2 Ozma of Oz
4 Cabin Fever
4 Jane Eyre: An Autobiography
4 The Mysterious Affair at Styles
4 The Nigger of The "Narcissus"
1 A Journal of the Plague Year
4 David Copperfield
4 The Adventures of Sherlock Holmes
```

Next we'll take a look at Agglomerative clustering which is a bottom-up approach. Each book is put into its own cluster and the next closest clusters are then grouped together. This goes on until all books are in a single cluster or when it reaches the number of desired clusters.

The two graphs below show average linkage and ward linkage. Average uses the average of the distances of each observation of the two sets. Ward minimizes the variance of the clusters being merged.





Ward linkage looks like it does a much better job dividing the books into various genres. It looks like it initially splits into 7 different hierarchies so we'll set up the model with 7 clusters, fit the model to the training books and make some predictions on the testing books.

```
Training Labels:
6 Little Women
2 The Piebald Hippogriff
0 Pride and Prejudice
0 Emma
2 Peter Pan
3 Dorothy and the Wizard in Oz
3 The Wonderful Wizard of Oz
3 The Marvelous Land of Oz
3 The Emerald City of Oz
6 The Phantom Herd
6 Black Jack
6 Wuthering Heights
0 The Secret Garden
0 Alice's Adventures in Wonderland
5 Don Quixote
```

```
Predicted Labels:
1 Sense and Sensibility
5 Ozma of Oz
0 Cabin Fever
1 Jane Eyre: An Autobiography
1 The Mysterious Affair at Styles
4 The Nigger of The "Narcissus"
4 A Journal of the Plague Year
1 David Copperfield
1 The Adventures of Sherlock Holmes
4 Each Man Kills
0 The Mysterious Rider
3 The Odyssey
4 The Legend of Sleepy Hollow
4 The Jungle Book
0 The Call of the Wild
```

Looking at a slice of the training labels and predicted test labels, there are a few surprises. The model ended with 6 clusters rather than 7 (not all are shown). While it trained all the Oz books into cluster 3, it put Ozma of Oz into cluster 5. It also put an adventure book (Call of the Wild) in with Pride and Prejudice & Emma where we would have expected Sense and Sensibility to be. From a glance it doesn't appear as if this model did a very good job.

Non-negative Matrix Factorization; NMF is not a traditional cluster algorithm but a topic clusterer and dimension reducer. This model won't be used to predict labels but is an interesting method of analyzing text data.. After fitting to the training set it creates different topics and expresses each book as a combination of these topics - like variable coefficients. Following are a few different models with the top words for each created topic.

```
Out of Box Model:
Topic #0: one go come say could would get make time see
Topic #1: one hath thou make may say socrates thy would man
Topic #2: dorothy oz scarecrow say wizard ozma woodman zeb nome emerald
Topic #3: mr say would go could come look one know emma
Topic #4: holmes watson say upon man mcmurdo one would sir well
Topic #5: say go dont get see one come well know look
```

The Frobenius norm can be considered as a vector norm, defined as the square root of the sum of the absolute squares of its elements.

```
Frobenius Norm Model:
Topic #0: say one go come would make could see time get
Topic #1: hamlet horatio polonius laertes rosencrantz ophelia guildenstern lord king marcellus
Topic #2: dorothy oz scarecrow wizard ozma woodman zeb nome toto emerald
Topic #3: bindle mr wot earty hearty stiffson millie ai gupperduck macfie
Topic #4: holmes watson upon baskerville culverton man mcmurdo sir stapleton barrymore
Topic #5: puddin bunyip bill possum bluegum say thieves sam wombat owners
```

The Generalized Kullback-Leibler divergence, handled by explicit normalization of one of the matrices.

```
Kullback-Leibler Divergence Model:
Topic #0: young youth way window wife wink whole whose whether ask
Topic #1: dissolve austerity fourths supplant prudence wood frequent capability besmear favorable
Topic #2: cloth bravely chasm cheer branch butcher chariot blew bread prisoner
Topic #3: bill harm chose uncertainty hitherto farthing agree unfolded within assume
Topic #4: beards deep brow dearest civilize bishareen come wire borderer axe
Topic #5: babylon carrot olympus world broaden fillip comin aster woodland worm
```

While these models are not being used to predict on the test set it is very interesting to see the top words for each model that it creates out of the training set of books and the variety of words between each version. The out of box model seems to match up more with the KMeans model and specific book elements are easily recognized; this is much less true for the two defined models of Frobenius Norm and Kullback-Leibler.

When comparing unsupervised learning models it can be difficult. There are no specific accolades that you can test for and evaluate. You either have to be knowledgeable about the topic or have some other expert who can say whether the model is performing well or not. While I am familiar with much of this data, I'm certainly no expert. I decided to create my own system to assess in order to have real numerical values that can represent some level of ability for each model. Using the existing subject lists for each book I can compare the predicted book clusters to the trained ones.

In some ways this undermines the purpose of having used unsupervised models instead of supervised. Some degree of labels do already exist but it is most likely there will be a substantial amount of overlap between patterns found by the unsupervised learners and pre-defined subjects. This estimated accuracy is by no means a comprehensive calculation of the model's performance but a quick comparison of similarity between books placed in the same clusters.

This estimated accuracy is determined using Project Gutenberg's subject labels; an example using Alice's Adventures in Wonderland is below. "PR" and "PZ" are classifications given by the Library of Congress where PR is 'language and literatures: English lit', and PZ is language and literatures: Juvenile Belles Lettres'.

```
({'Alice (Fictitious character from Carroll) -- Juvenile fiction',
  "Children's stories",
  'Fantasy fiction',
  'Imaginary places -- Juvenile fiction',
  'PR',
  'PZ'})
```

The created estimated accuracy metric takes one test book at a time and compares the words in its subject with *all* of the words in each subject of the training book set for the same cluster of the selected test book. It then calculates the percentage of words found in the training set that were also in the test book's subject.

For example, if the test book Ozma of Oz is placed into cluster 3, the function pulls all the training books that were also in cluster 3 and takes each word in each subject to compare to the subject of Ozma of Oz. The resulting percentage is the amount of words the test subject shares with the training subjects.

This is a decent general estimated accuracy calculator for each prediction but there are a few potential major pitfalls to this system that must be noted. The bigger the training set of books is the more likely the 'accuracy' of the predicted books will increase; this is due to the larger list of words the test subject will be compared to. For the same reason, when a model is using fewer clusters the accuracy may also artificially increase. The subjects for each book are retrieved from Project Gutenberg, some of these are overly specific (which may help or hurt scores) and while often include the Library of Congress book labels they are not always what you may expect for a certain book. A few of these subjects are only a few words making it very easy for them to hit perfect scores when compared to trained subjects.

Now, let's compare some actual scores! Below are the printouts of estimated accuracies for a range of number of clusters for both the KMeans and Hierarchical clustering models.

KMeans Model

3 clusters
Average Accuracy: 82.07%

4 clusters
Average Accuracy: 81.86%

5 clusters
Average Accuracy: 81.65%

6 clusters
Average Accuracy: 81.45%

7 clusters
Average Accuracy: 81.45%

8 clusters
Average Accuracy: 70.66%

Hierarchical Model

3 clusters
Average Accuracy: 15.39%

4 clusters
Average Accuracy: 16.21%

5 clusters
Average Accuracy: 26.13%

6 clusters
Average Accuracy: 27.17%

7 clusters
Average Accuracy: 24.13%

8 clusters
Average Accuracy: 26.30%

For a simple model KMeans is really doing a pretty good job while the Hierarchical model is almost impressively incompetent. Next are the estimated accuracies for a selection of the test books individually using the two models with KMeans at 5 clusters and Hierarchical 6. 5 was selected for KMeans despite its lower percentage in an attempt to return more accurate estimations (while lower) and there should be at least 5 fairly distinct types of books in this dataset.

KMeans:

Austen, Jane -- Sense and Sensibility
Accuracy: 85.71%

Baum, L. Frank (Lyman Frank) -- Ozma of Oz
Accuracy: 90.00%

Bower, B. M. -- Cabin Fever
Accuracy: 100.00%

Doyle, Arthur Conan -- The Adventures of Sherlock Holmes
Accuracy: 92.31%

Twain, Mark -- The Adventures of Tom Sawyer
Accuracy: 77.27%

Plato -- Ion

Hierarchical:

Accuracy: 44.44%
Austen, Jane -- Sense and Sensibility
Accuracy: 19.05%

Baum, L. Frank (Lyman Frank) -- Ozma of Oz
Accuracy: 30.00%

Bower, B. M. -- Cabin Fever
Accuracy: 100.00%

Doyle, Arthur Conan -- The Adventures of Sherlock Holmes
Accuracy: 23.08%

Twain, Mark -- The Adventures of Tom Sawyer
Accuracy: 18.18%

Plato -- Ion Accuracy: 11.11%

It was already clear that KMeans is the superior model, but this comparison shows a good example of the issues previously stated with the estimated accuracy calculator. Even the *awful* hierarchical prediction for Cabin Fever is 100% accurate. This is because the subject retrieved from Project Gutenberg is only a couple words that were easily found in the training set. We can also see that there are some predictions significantly under the average although these problems may be corrected by simply having a larger dataset. A full list of the test book accuracies can be found at the end of this report.

While this KMeans model clearly struggles on some genres like philosophy and some adventure novels - it does very well with most children's/juvenile and mystery books. This can most likely be contributed to the small and somewhat narrow selection of books. The methods used in this project (not including Hierarchical clustering) show great potential in the ability of unsupervised learning that may soon be successfully applied to a book recommendation system.

Of course there is plenty of room for improvement and exploration into other concepts such as gram modeling (using word phrases rather than individual words), latent dirichlet allocation, further dimensionality reduction, and improving subject descriptions for a more precise accuracy estimator. First and foremost the dataset will be expanded which is often the best way to advance a machine learning model.

KMeans Book Classification Estimated Accuracy:

Brontë, Charlotte -- Jane Eyre: An Autobiography
Accuracy: 89.47%

Christie, Agatha -- The Mysterious Affair at Styles
Accuracy: 76.92%

Conrad, Joseph -- The Nigger of The "Narcissus"
Accuracy: 68.75%

Defoe, Daniel -- A Journal of the Plague Year
Accuracy: 81.82%

Dickens, Charles -- David Copperfield
Accuracy: 83.33%

Glad, Victoria -- Each Man Kills
Accuracy: 100.00%

Grey, Zane -- The Mysterious Rider
Accuracy: 100.00%

Homer -- The Odyssey
Accuracy: 75.00%

Irving, Washington -- The Legend of Sleepy Hollow
Accuracy: 100.00%

Kipling, Rudyard -- The Jungle Book
Accuracy: 72.22%

London, Jack -- The Call of the Wild
Accuracy: 37.50%

Lovecraft, H. P. (Howard Phillips) -- The Dunwich Horror
Accuracy: 100.00%

Roosevelt, Theodore -- Theodore Roosevelt: An Autobiography
Accuracy: 57.14%

Shelley, Mary Wollstonecraft -- Frankenstein; Or, The Modern Prometheus
Accuracy: 64.29%

Stevenson, Robert Louis -- Treasure Island
Accuracy: 81.82%

Wallace, Edgar -- The Clue of the Twisted Candle
Accuracy: 100.00%

Wells, H. G. (Herbert George) -- The Island of Doctor Moreau
Accuracy: 100.00%

Hierarchical Book Classification Estimated Accuracy:

Brontë, Charlotte -- Jane Eyre: An Autobiography
Accuracy: 10.53%

Christie, Agatha -- The Mysterious Affair at Styles
Accuracy: 15.38%

Conrad, Joseph -- The Nigger of The "Narcissus"
Accuracy: 25.00%

Defoe, Daniel -- A Journal of the Plague Year
Accuracy: 27.27%

Dickens, Charles -- David Copperfield
Accuracy: 8.33%

Glad, Victoria -- Each Man Kills
Accuracy: 28.57%

Grey, Zane -- The Mysterious Rider
Accuracy: 100.00%

Homer -- The Odyssey
Accuracy: 16.67%

Irving, Washington -- The Legend of Sleepy Hollow
Accuracy: 14.29%

Kipling, Rudyard -- The Jungle Book
Accuracy: 27.78%

London, Jack -- The Call of the Wild
Accuracy: 31.25%

Lovecraft, H. P. (Howard Phillips) -- The Dunwich Horror
Accuracy: 12.50%

Roosevelt, Theodore -- Theodore Roosevelt: An Autobiography
Accuracy: 0.00%

Shelley, Mary Wollstonecraft -- Frankenstein; Or, The Modern Prometheus
Accuracy: 28.57%

Stevenson, Robert Louis -- Treasure Island
Accuracy: 27.27%

Wallace, Edgar -- The Clue of the Twisted Candle
Accuracy: 25.00%

Wells, H. G. (Herbert George) -- The Island of Doctor Moreau
Accuracy: 25.00%