Determining Genre of Classical Literature with Machine Learning

Ben Guthrie, Jordan Henstrom, Ben Horrocks, Ryan West
Department of Computer Science
Brigham Young University
CS 478, Winter 2019

Abstract

Text classification is a popular problem in machine learning. The ability for a program to understand classifications and distinctions between texts can be useful for a wide array of real-world applications. It was because of this that we decided to focus on teaching our learning model to classify entire books worth of text as either fiction or nonfiction. Using indicators such as word frequency and word count allowed us to approach the problem. We trained on a variety of different models. We initially found that due to possible imbalance in our data set, MLP tended to perform poorly, while Decision Trees and Ensembles of Random forests performed much better. We refined our model with testing a ensemble of our previous models. But found that due to the poor performance of MLP and clustering, the overall accuracy of the Ensemble we created didn't not perform as well as the Ensemble of Random Forests.

1 Introduction

Classification of literary works is a different beast than normal text classification. A big reason for this can be attributed to length, books are just much longer than most other text mediums. Additionally, the classifications of literary works are more nuanced than other sources such as a newspaper article. This has lead to a wide array of different results in previous models for literary classification. A large part of the background information that contributed to this project comes from things that we learned throughout the semester. However there were additional avenues beyond class materials that we explored in an effort to improve and refine our model.

1.1 Background information

One such avenue was the paper *Genre Identification and the Compositional Effect of Genre in Literature* authored by Joseph Worsham and Jugal Kalita. This study was of interest to us in part because they used the same data set that we used. Their study focused on specific genre classification such as romance or adventure stories. Although our learner is focused more broadly on distinguishing fiction from nonfiction, it was useful to read the work of those who approached a similar topic. When discussing features, Worsham noted that using word frequencies alone is usually inadequate for the purposes of genre classification. Word frequency is defined as how many times the a word appears in a section, in this case an entire book. This lead to us including additional features beyond a bag of words to our model.

Another model that we looked at was a model based off predicting genre only by the title of the book. THis model was created by a github user by the name Akshay Bhatia. Similarly to the paper by Worsham and Kalita, Bhatia's model focuses on classifying works into several genres like adventure and romance. However the ability to get a initial guess just off the title of a book was useful to us when we began refining our model and trying different approaches to increase our prediction accuracy.

1.2 Libraries Used

For running our models, we used the Scikit-Learn. We decided to use this model for a few reasons. First, it was an easy library to use with great documentation and examples to reference. Secondly, it was designed to work with NumPy and Pandas, two libraries that we used with in our project. Lastly, it is widely used in industry and is thought of heighly. Because of this we felt confident that it was a good choice for our models.

Additional libraries that we used were NumPy, Pandas, and SpaCy. NumPy is used widely throughout most python programming. Pandas is a common library that is often used to organize data to be fed into learning algorithms. SpaCy is a library for creating word vectors. Word vectors are lists of numbers that are used to represent a word's attributes. This allows for a numerical representation of a word that allows someone to treat them more like continuous values. This is applicable to our models because working with continuous values opens new ways we can try to learn with them. For our purposes, we didn't use SpaCy for most of our project. However we did use it when we implement improvements to our initial models.

2 Methods

As we stated earlier, we focused on the classification of literary texts as either fiction or nonfiction. In this situation

we defined nonfiction as anything that purports to be nonfictions (autobiographies, biographies, textbooks, etc.). Although works like biographies and autobiographies tend to come with a lot of bias and possible embellishments, we considered them nonfiction for our purposes. We decided on this because the author viewed the works as nonfiction and therefore likely wrote them using the conventions of the genre.

2.1 Data Source

The source for our data set was the web API of Project Gutenberg. This is a project aimed at making a selected count of literary classics available as free ebooks. This allowed us to quickly access hundreds of literary works to use in our model. Using a parser, we downloaded and parsed 963 instances. We then labeled each instance as either "fiction" or "nonfiction" based on wikipedia and goodreads.

There were a few hurdles that we had to address when it came to our data source. The first was that Project Gutenberg would have duplicates of some books. We didn't quite know why this was the case, but ultimately we decided that having a few duplicates in our data set wouldn't hurt our model's ability to work. This is because the average scenario would be that we would have duplicates of fiction works at the same rate as those of nonfiction works. This meant that our overall data would set would have the same ratio.

Another problem we had was that many books would be in different languages. We decided that since we were using english words, we should disregard these entries. As a result, we did not include any works that were written in different languages.

The last major hurdle was that the API that we pulled these books from did not have their genre listed in any obvious place. This meant that for our entire data set we had to hand label each instance as either fiction or nonfiction. This was the main reason that our data set ended up being smaller than we would have initially liked.

2.2 Data Set

When choosing how we wanted to construct our data set, we had to be selective in picking what we saw as the most important features in order to make up for only having roughly 1000 instances in our data set. Ultimately we decided that in addition to having a word count approach, we would also include average sentence length, word count, and average word length. We felt that these would allow our model to have a better classification accuracy.

We originally wanted to categorize our books into multiple genres (horror, romance, sci-fi, fantasy, mystery, and nonfiction), but when labelling our data, we found that it was difficult at times to label these books into these categories. In the end, we decided to simplify it to classification as either fiction or nonfiction. This allowed us to label data more quickly, increasing the size of our data set.

The largest part of our data set is word frequencies for the 1000 most common words in the english language. We chose this one because although each instance in the data set had 0's for most of the words, we thought that the words they did have would be a good indicator of whether or not they were fiction. For example, the word 'republican' would be an indicator that the work is either a historical fiction centering around some governmental plot, or it is nonfiction. Since the latter is more common by far, this word could help to correctly classify the work.

Another feature that we used was average word length. The average length of words in the english language is 4, however we didn't know if this average held between fiction and nonfiction. It is possible that nonfiction is more pedantic than fiction and therefore will have a larger average. We didn't know and decided it would be an interesting metric to try to see if it was indicative in any way.

We chose to include average sentence length for the same reason as average word length. There is no solid evidence of a correlation in sentence length and genre. However this could have been a feature that when paired with another, gives us an idea on it's genre.

Our last feature was how many words appeared in the book. This metric allowed for multiple of the same word, so it was mere a measurement of length. We concluded that this would be a useful metric if either fiction or nonfiction works were generally longer than the other. In that case, this feature would likely have a strong correlation to the genre of the book.

2.3 Selected Models

We decided on four models initially: Multi-Layer Perceptron, a clustering algorithm utilizing nearest centroid clustering, a Decision Tree, and an ensemble of Random Forest classifiers. Although other algorithms like K Nearest Neighbor and Naive Bayes were considered for our initial runs, we thought that sticking to a small number for our initial results would be better.

We decided that using a MLP learner would be a good idea because of how many features we had. Using an MLP would allow our model to virtually filter out features that were not important for prediction. Additionally this was the algorithm that we all had the most experience because of the lab we did in class being the longest.

We decided to use a clustering algorithm because they tend to lend themselves well when things of the same classification are similar in the content. That is to say that an assumption that we made is that nonfiction books are similar to other nonfiction books, and the same is true for fiction books. This is a pretty big assumption. However by including the learner we were able to test this idea.

The third model that we used was decision tree. This model would work well if there were certain words that had a large effect on what genre the book was. In order to use this algorithm had to be able to divide the feature values into nominal buckets, but luckily the python library we used, Scikit-Learn, was able to do this quite well.

The last model that we initially tried was a Ensemble of Random Forests. We used this one largely because we wanted to include an ensemble as one of our models, and random forests are essentially collections of Decision Trees. This allowed us to explore something that we didn't cover intensively in class.

3 Initial Results

4 Data and Feature Improvements

5 Final Results

6 Conclusions

7 Future Work

7.1 Feature Refinement

7.2 Other Models