"A hurricane, a star, and an insect walk into a bar..." Can we predict Hugo, Nebula, and Locus science fiction book award winners?

IST718 – Final Project
Luke Miller, Jon Kaplan, John Fields
June 11, 2020

Overview

Popular culture is strongly influenced by so-called 'speculative fiction.' From *Lord of the Rings* to *Harry Potter* we see fiction shaping the imaginations of children and adults alike.

Sometimes this speculative fiction can inspire hope and optimism. Other times, we see it challenge societal norms through the disguise of metaphors and escapism. We see life imitate art, where often technological progress is described in stories long before it becomes real. Speculative fiction can be a driving force for technological change.

A particularly influential set of fantasy and science fiction books are those that win literary awards. The three big awards for these genres are the Hugo, Nebula, and the Locus awards. When a book wins–or is even nominated for–one or more of these they receive higher recognition, sell more copies, and reach readers who wouldn't have found them before. So, these awards are important to both authors and readers and end up influencing popular culture.

We set out to explore the factors that go into an award-winning speculative fiction book. We hypothesized that we could predict award nominees from looking at reader reviews on goodreads.com, along with other features. A second goal was to attempt to differentiate between books that win awards and books that don't from within the pool of nominees.

While we achieved some good success in predicting nominees, we discovered that differentiating between award winners and losers is a much more difficult problem. Additionally, by the time books go through the nomination award process, they have been filtered down to 'small data' that may be better predicted by human experts than classification problems. However, by re-balancing the dataset, we were able to predict winners with reasonable accuracy as described in more detail below.

Specification

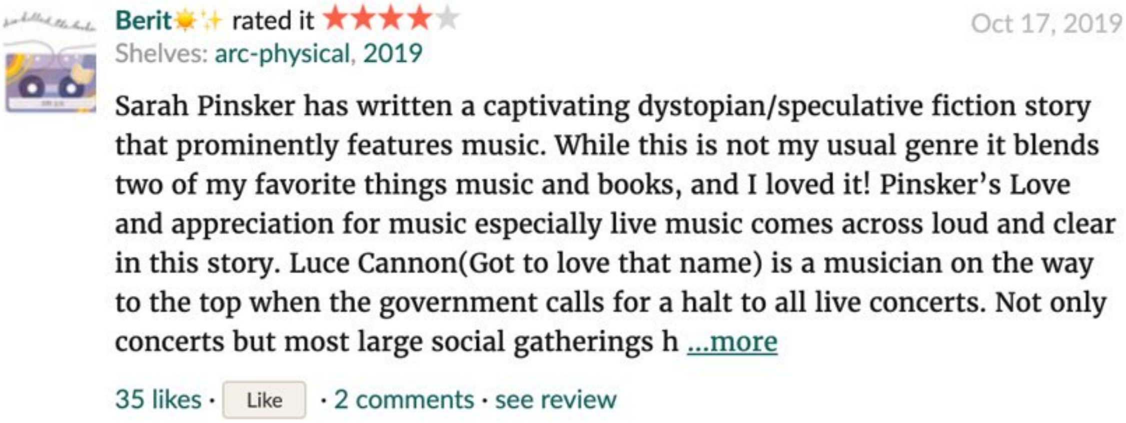
Our goal was twofold: to first classify nominees versus non-nominees and then to classify winners and losers from within the nominee pool. We would look at several data sources.

Data sources

We were able to start with a large set of Goodreads.com data scraped by the University of California-San Diego (UCSD) Book Graph project in 2017[1]. The data consisted of:

- Full Dataset - 2.4 million books and 15.7 million reviews
- Fantasy-Paranormal Subset - 259k books and 3.4 million reviews

An example review from Goodreads.com:



Berit☀✨ rated it ★★★★☆ Oct 17, 2019

Shelves: arc-physical, 2019

Sarah Pinsker has written a captivating dystopian/speculative fiction story that prominently features music. While this is not my usual genre it blends two of my favorite things music and books, and I loved it! Pinsker's Love and appreciation for music especially live music comes across loud and clear in this story. Luce Cannon(Got to love that name) is a musician on the way to the top when the government calls for a halt to all live concerts. Not only concerts but most large social gatherings h ...more

35 likes · Like · 2 comments · see review

In addition to the UCSD data we gathered additional data:

- Goodreads.com science fiction historical award data scraped in May 2020 (3590 award nominees and winners)
- Goodreads.com reviews for the Nebula award nominees scraped in May 2020 (9365 reviews)
- 6,000 science fiction authors scraped from www.worldswithoutend.com in May 2020. The author data was then merged with the Nominees Dataframe and added features such as gender, occupation, age, and nationality.

[1] https://sites.google.com/eng.ucsd.edu/ucsdbookgraph

We also attempted to use the New York Times Bestseller API in order to gain insight into how well a nominated book sold, but were ultimately unable to successfully merge with the nominations dataframe.

Strategy

Since the gathered data included structured and text data, the team employed a variety of approaches. We had some experience using Bidirectional Encoder Representations from Transformers (BERT)[2] for natural language processing, so set out to classify nominees from the text of the book reviews. We also wanted to test Apache Spark to understand the capabilities for processing large datasets.

First, we would attempt to classify nominees vs non-nominees using a transformers analysis of book reviews. Then we would attempt to classify winners vs losers using transformers as well as more traditional analyses.

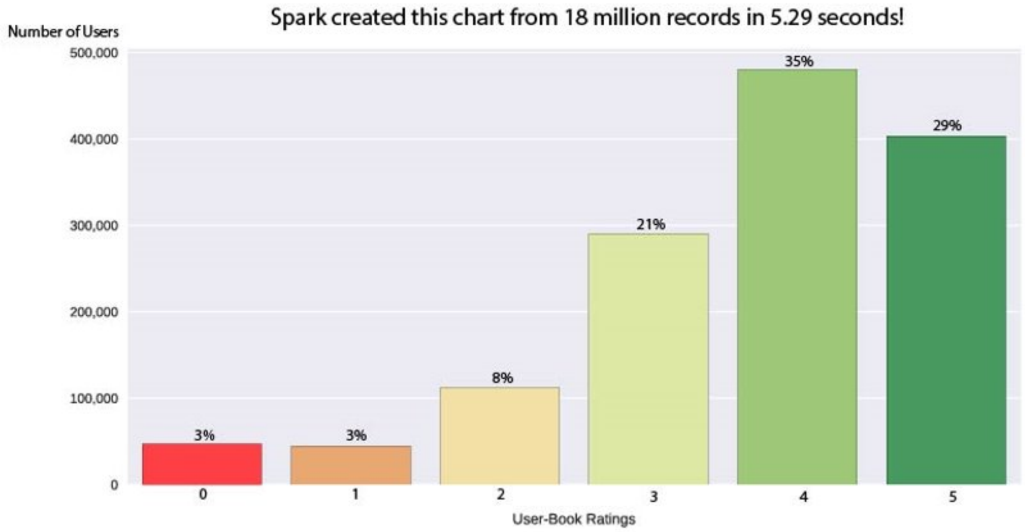
For cleaning, all the award categories were extracted and grouped together in a way that allowed for all three award entities to share the same category with the other awards. For example, Locus did not have a Best Novel category, so its Best Horror, Best Science-Fiction, and Best Fantasy categories were all grouped in the Best Novel category. For the Best New Writer category under AwardCategory, Locus called it Best First Novel while Hugo called it Best New Writer but both were grouped together anyways. Nebula did not have a best new writer category. All three award entities had a Best Novella, Best Novelette, and Best Short Story category.

The additional scraped data for demographics was also very messy, and therefore a lot of text cleaning was done to format the data better (striping white spaces, removing special characters). Once everything was cleaned and columns were converted to the right data type, the nominees dataset was ready for analysis.

[2] Devlin, J., Ming-Wei, C., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of deep bidirectional transformers for language understanding*. Ithaca: Cornell University Library, arXiv.org. Retrieved from https://search-proquest-com.libezproxy2.syr.edu/docview/2118630252?accountid=14214

Observation

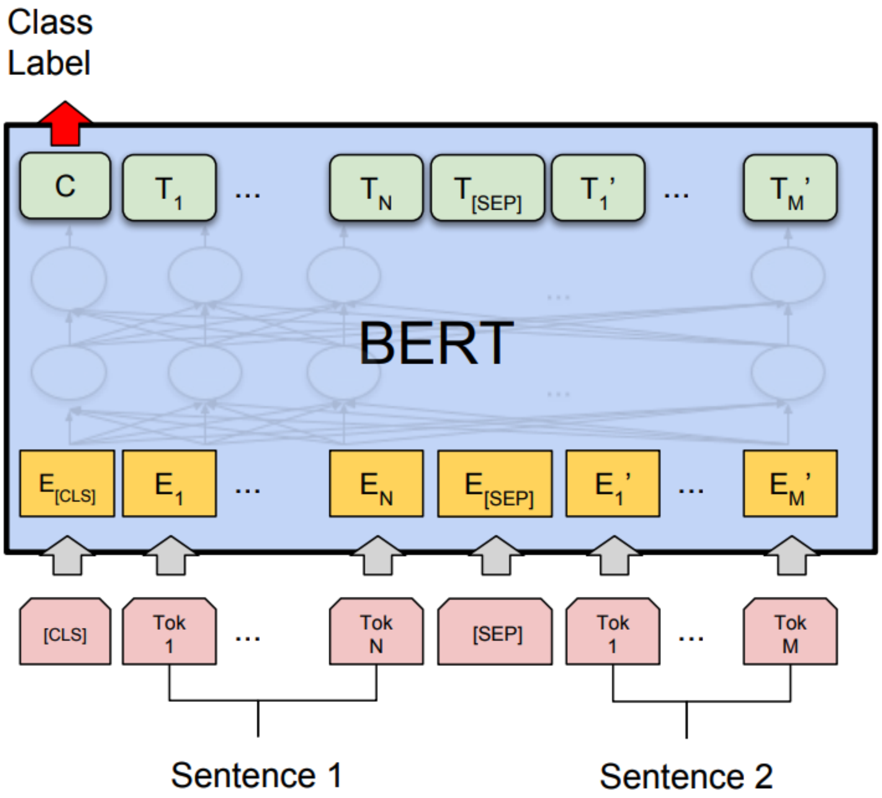
The initial tests with the 18 million records from Goodreads on Apache Spark were very promising and we were able to generate the chart below in 5.29 seconds.



Source: Goodreads | University of California-San Diego Book Graph Project

However, when we tried to merge or subset the dataset in Spark it generated errors in Google Colab and our account was disabled twice for using too many resources. After consulting with Professor Fox, we decided to utilize a subset of data from UCSD which contained the “fantasy paranormal” genre and included 35% of the reviews from nominees/winners of book awards.

Based on the NLP classification of `SimpleTransformers` we discovered that we could feed the paragraphs from the book reviews to predict nominees/winners from a subset of the 259k books and 3.4 million reviews in Fantasy-Paranormal dataset. The illustration below shows how the BERT model takes sentences from book reviews and then uses deep learning to compare the sentence structure with a pre-trained corpus from all of Wikipedia and Google Books. The sentence structure is then mapped into a 768-dimension space to make comparisons using linear algebra concepts. The result is a classification of nominee/non-nominee or award winner/non-winner.



Source: https://gluon-nlp.mxnet.io/examples/sentence_embedding/bert.html

We also utilized traditional machine learning classification tasks such as logistic regression, k-nearest neighbor, decision trees and support vector machines on other variables from Goodreads such as:

- Ratings Count
- Text Reviews Count
- Average Rating
- Author Followers
- Book Publication Year

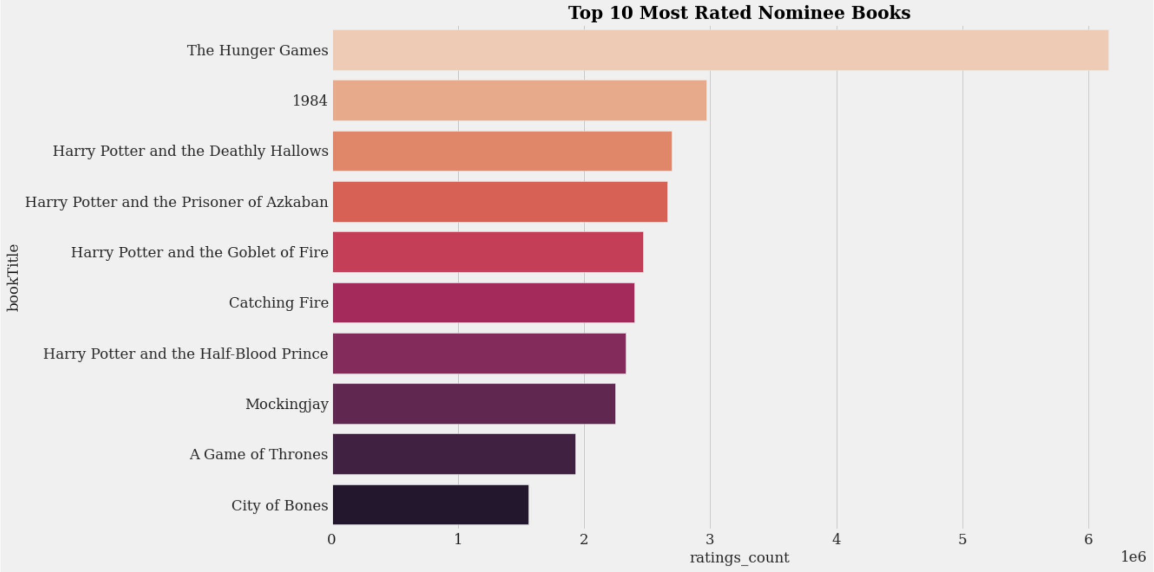
The goal for the observations of BERT/SimpleTransformers and the traditional analysis was to compare contrast these "old" and "new" methods to understand the differences in the results which will be reviewed in the analysis section.

Analysis

First, we performed some exploratory analysis to understand our data.

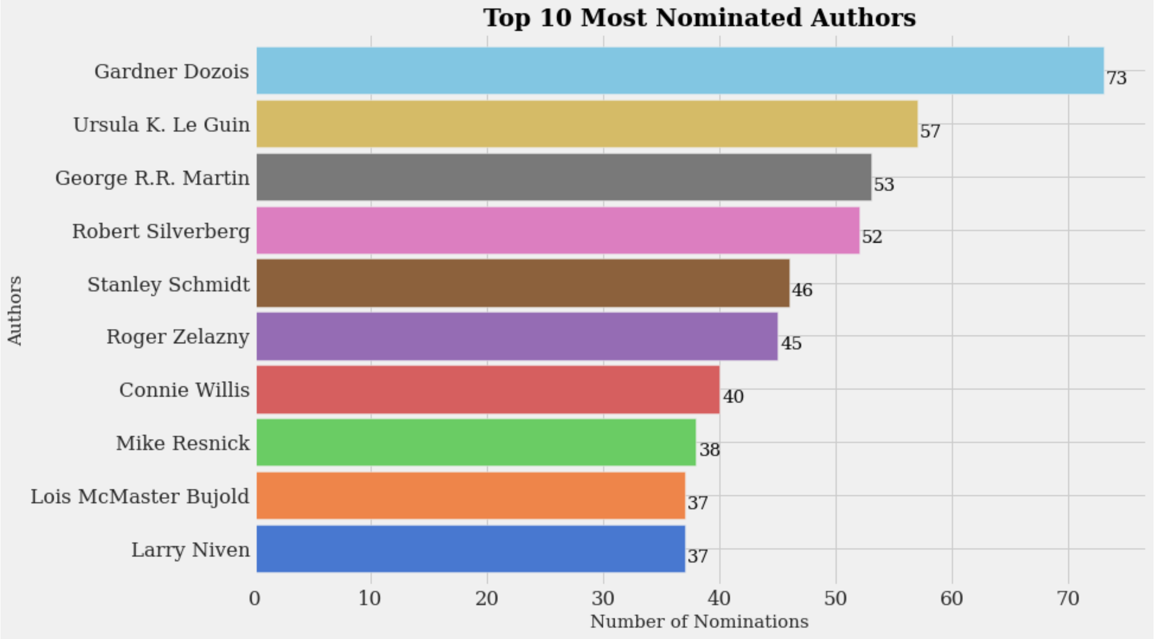
Exploratory Analysis of Nominees

We performed some general analysis to get an idea of the top books and authors across the nominees.



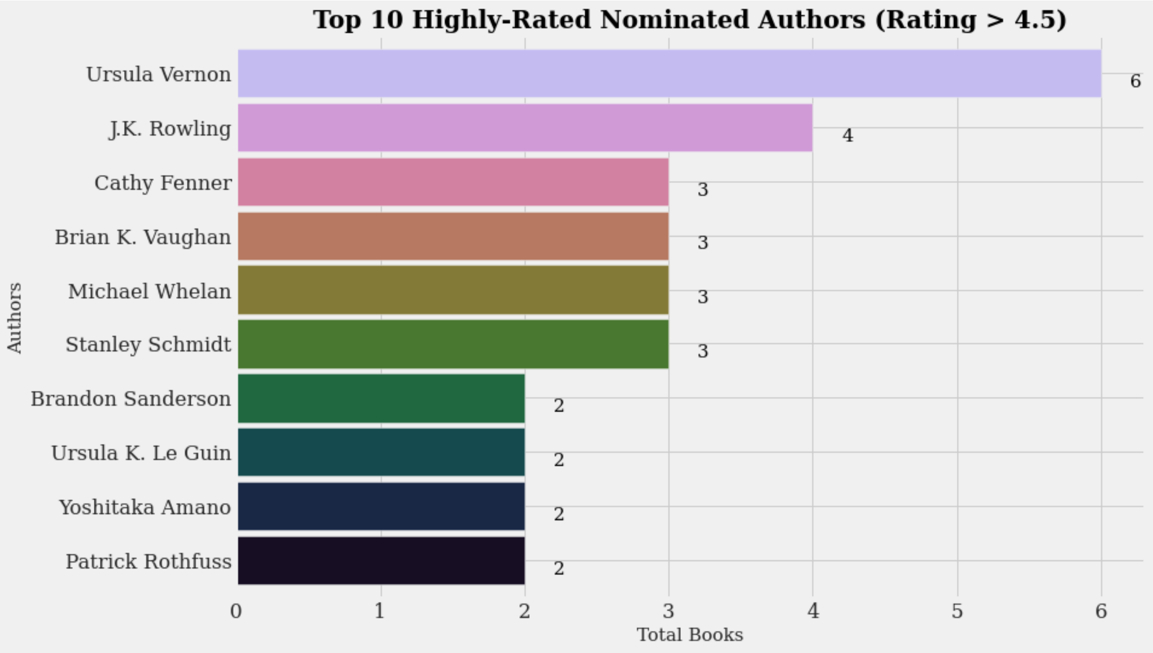
It is apparent that the books with the most ratings are all books that are also in other mediums such as television and movies. This cross-media appeal is probably the reason for the extra exposure.

When looking at the top authors nominated, it is apparent that there are a few especially quite dominant:



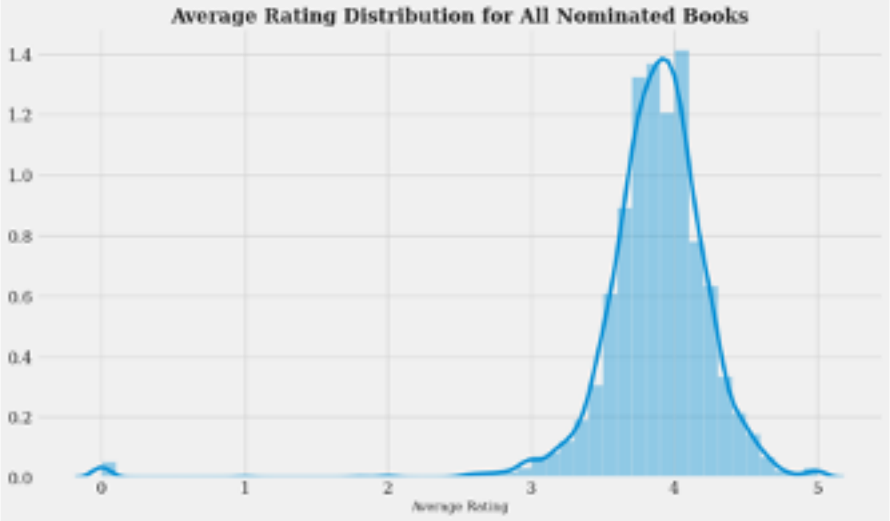
To see if any of these top nominated authors were highly reviewed as well, a count of nominated authors and books with an average rating of 4.5 was done.

Unsurprisingly, a few of the top nominated authors are also highly-rated:



Analyzing the Nominees Across Award Entities

A comparison of the different numeric categories across award entities was done to see in what ways they differed across nominees.



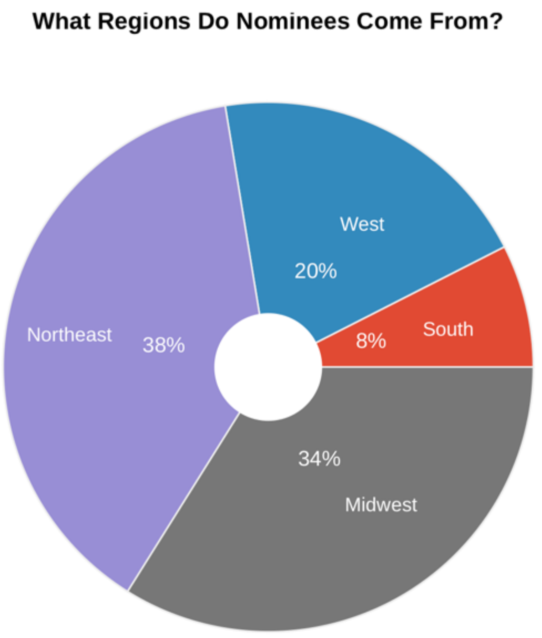
Breakdown Between Nominees

Average for Each Book:	**Nominee**	**Winner**	**Hugo Nominee**	**Nebula Nominee**	**Locus Nominee**
Average Rating	3.87	3.93	3.87	3.78	3.90
Ratings Count	29,625	36,694	24,391	20,671	36,205
Text Reviews Count	1,503	1,885	1,163	1,143	1,834
Number of Pages	311	328	273	253	354
Author Followers	12,733	15,973	8,980	4,516	18,223
Book Series	38%	36%	33%	31%	44%
GoodReads Author	35%	29%	33%	37%	37%
Age of Author	43.34	44.24	44.20	44.73	42.16
Female	28%	29%	22%	29%	30%
Male	72%	71%	78%	71%	70%
Prior Nominations for Author at Time of Nomination/Win	7.49	9.14	8.10	5.83	7.88

- The Locus nominees tend to have longer books at an average of 354 pages compared to Hugo's 273 and Nebula's 253 average pages.
- The Locus awards has the highest percentage (44%) of nominees being from a book series.
- Locus has the highest amount of authors being an official GoodReads author (37%), which may be the reason for it having the highest text reviews count (1,834), ratings count (36,205), and average rating (3.90) among all of the other awards entities.
    - One possible reason for this could be due to authors who have GoodReads profiles engage more with their fans and have more followers (Locus has the most author followers at 18,223).
- The average age of a nominated author is about 44 years old.
- Most nominated authors have prior nominations before which may indicate that it is harder to get nominated if a new writer.
- There seems to be a gender imbalance that needs to be further investigated.

Nominees Background

Most nominees are American and come from the regions of the Northeast (38%) and Midwest (34%):



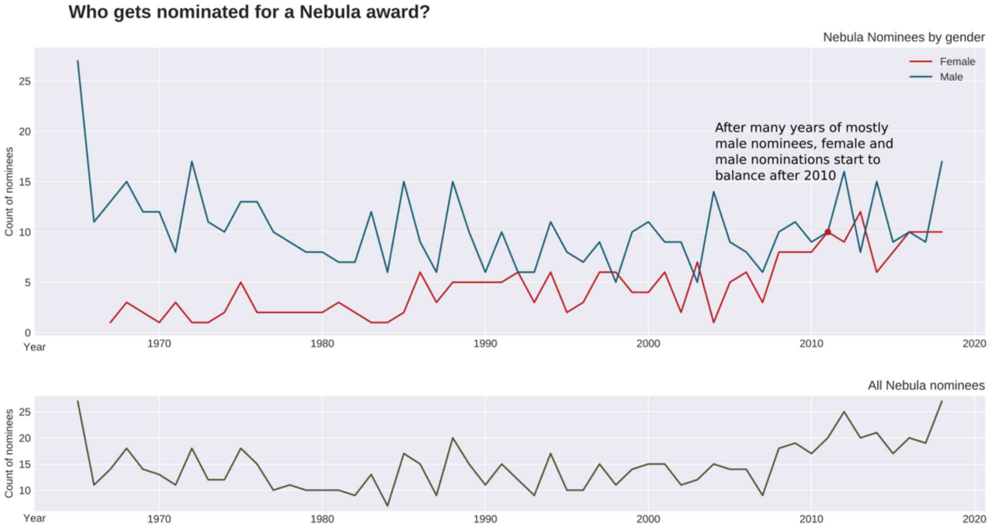
Since being a writer is known as a difficult career for financial stability, it is not surprising that many authors have other occupations outside of just being a writer:

- Librarian
- Anthropologist
- Publisher
- Psychologist
- Linguist
- Hypnotist
- Klingon language expert
- Lawyer
- Aerospace engineer
- Chemical engineer
- Futurist
- Truck Driver
- Marine biologist
- Psychic medium
- Occultist

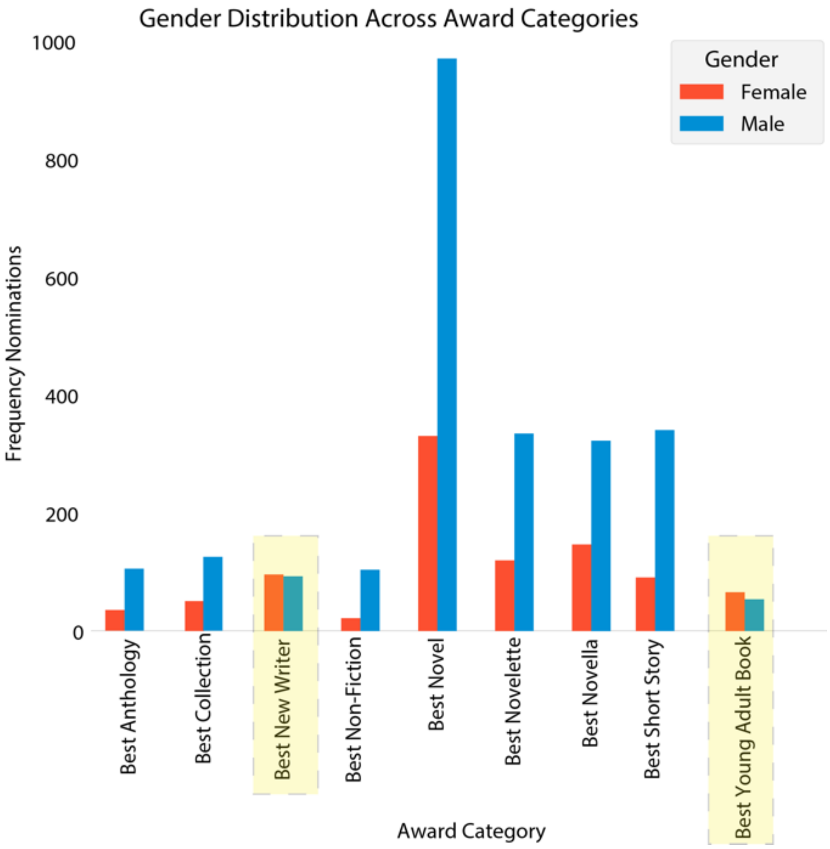
Some are expected from writer stereotypes, such as futurist, Klingon language expert, and aerospace engineer, but others are more surprising, such as truck driver and occultist. The wide range of occupations highlights how nominees come from a diverse background of fields and occupations. In the genre of science-fiction, it is most likely a great asset to have a technological background since so many stories are based around technology.

Gender

Gender is an interesting topic regarding the awards. Even though the authors nominated are mostly male, there is a new trend since 2010 of more females being nominated for awards. Since 2010, female nominees have mirrored men and the discrepancy is smaller and smaller, as shown in the Nebula trends:



When looking at the distribution between award categories, most categories are male-dominated, but two categories have more females than men, Best New Writer and Best Young Adult Book:



This may indicate a trend of more female writers breaking into a society that is increasingly becoming more open and welcoming of diverse perspectives.

Award Categories

The different award categories were examined as well to see if there are any trends between categories.

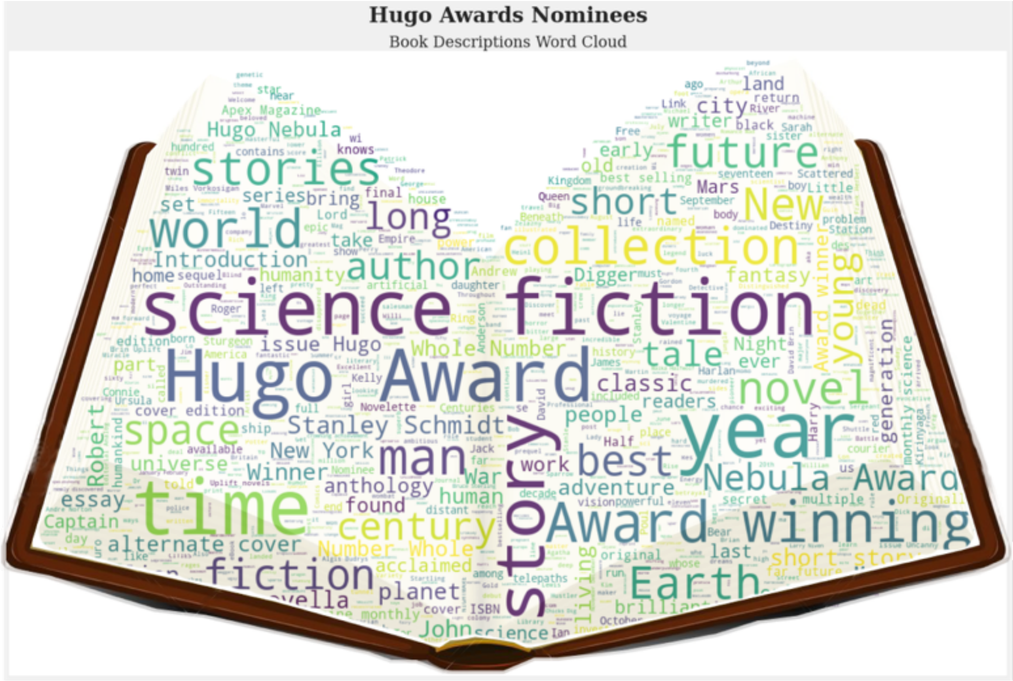
Award Categories Breakdown Between Nominees

Average for Each Book:	**Best Novel**	**Best Novella**	**Best Novelette**	**Best Short Story**
Average Rating	3.89	3.76	3.80	3.73
Ratings Count	54,966	5,247	1,226	772
Text Reviews Count	2,632	532	126	81
Number of Pages	401	170	165	169
Author Followers	20,110	6,495	6,974	4,977
Book Series	57%	25%	14%	14%
GoodReads Author	34%	33%	41%	35%
Age of Author	43.47	43.43	42.67	44.24
Female	25%	31%	26%	21%
Male	75%	69%	74%	79%
Prior Nominations for Author at Time of Nomination	7.41	8.92	8.07	6.71

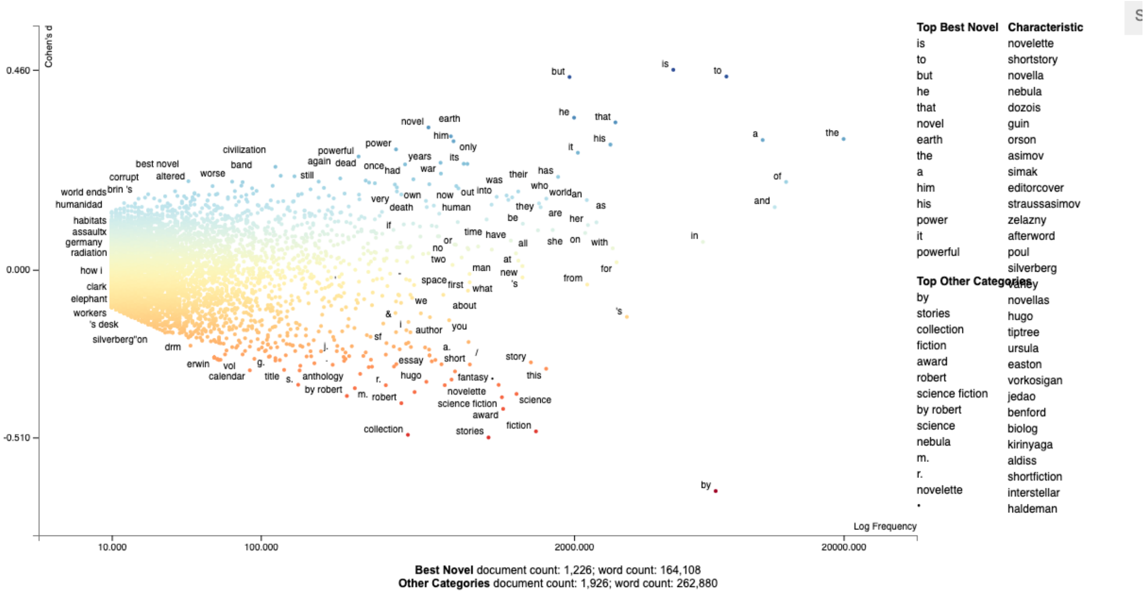
- 57% of Best Novel nominees come from a book series.
- Best Novel is clearly the most popular category with a much larger ratings count (54,966) and author followers (20,110) compared to others.

NLP of Book Descriptions

A word cloud was made to get a perspective of the different words used in the book descriptions of nominated books:



An examination of the most frequent words in the book descriptions was displayed another way using the Python library Scattertext. It enables the user to type any word they want and see what frequency that word appears across all the book descriptions of Nominees. The text below the graph also gives context for the word by showing what sentence it was used in and for what award entity. Cohen's D method was used for the scatterplot which is a measure to measure effect size. (Interact with the graph at the following link, but be aware it takes several minutes to load: https://jonkaplan18.github.io/jonkaplan18/index.html)

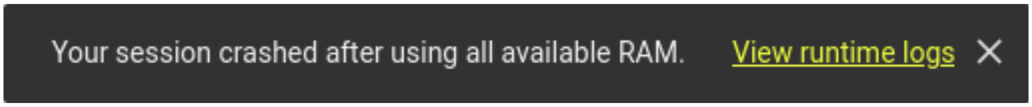


Predicting Nominees with SimpleTransformers

In order to use variations of BERT (RoBERTa, Electra, and others) we implemented the SimpleTransformers library[3]. This is built on top of the Transformers library by HuggingFace[4] and is the simplest method we found to do NLP classification with transformers. This package uses Pytorch and TensorFlow 2.0 to allow various analyses of natural language.

In order to predict whether a book is a nominee we needed to balance the classes by upsampling the nominees. Runs with no preprocessing end up achieving excellent *accuracy* by predicting 'not a nominee' in all cases. Obviously, this is not helpful. We needed to find some true positives in order to have a chance at predicting.

In Google Colab we didn't have the resources necessary to analyze the entire dataset:



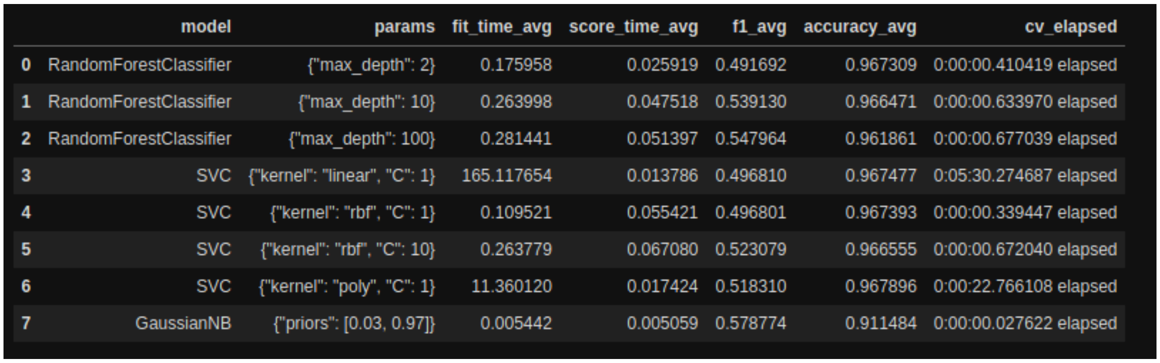
Your session crashed after using all available RAM.

We achieved 85% accuracy when doing test/train from the first 5,000 records. That number went down to 75% when we scaled up to 50,000 test/train records—probably because the very small sample set wasn't shuffled correctly. Once we had this, we were able to predict nominee status for the combination of a review and a book. However, this wasn't enough to predict a book's nominee status from ALL reviews.

Since a book has many reviews, some of them will predict nominee and others will not. We had to determine the best way to aggregate the transformers predictions in a way that would effectively predict for the book itself.

We tested several sklearn models to predict nominee status for each book that had 5 or more reviews. We used the count of predictions and the sum of the predictions that were predicted 'yes'. From here we found that we could get from 92-97% accuracy depending on the model:

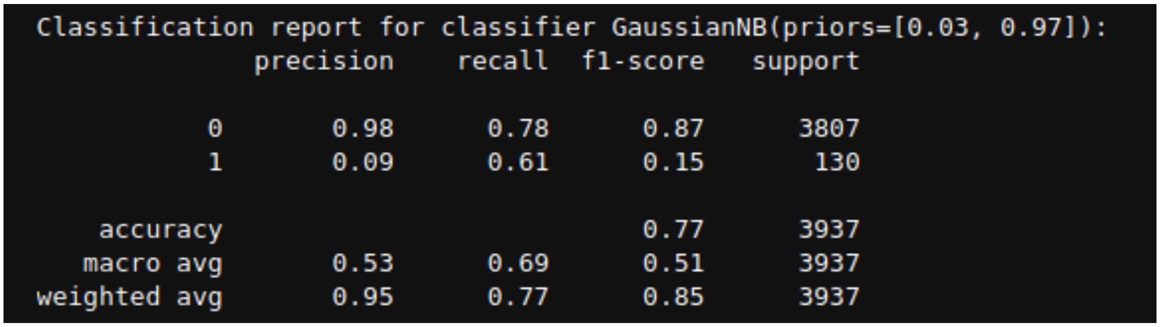
[3] https://github.com/ThilinaRajapakse/simpletransformers
[4] https://github.com/huggingface/transformers



	model	params	fit_time_avg	score_time_avg	f1_avg	accuracy_avg	cv_elapsed
0	RandomForestClassifier	{"max_depth": 2}	0.175958	0.025919	0.491692	0.967309	0:00:00.410419 elapsed
1	RandomForestClassifier	{"max_depth": 10}	0.263998	0.047518	0.539130	0.966471	0:00:00.633970 elapsed
2	RandomForestClassifier	{"max_depth": 100}	0.281441	0.051397	0.547964	0.961861	0:00:00.677039 elapsed
3	SVC	{"kernel": "linear", "C": 1}	165.117654	0.013786	0.496810	0.967477	0:05:30.274687 elapsed
4	SVC	{"kernel": "rbf", "C": 1}	0.109521	0.055421	0.496801	0.967393	0:00:00.339447 elapsed
5	SVC	{"kernel": "rbf", "C": 10}	0.263779	0.067080	0.523079	0.966555	0:00:00.672040 elapsed
6	SVC	{"kernel": "poly", "C": 1}	11.360120	0.017424	0.518310	0.967896	0:00:22.766108 elapsed
7	GaussianNB	{"priors": [0.03, 0.97]}	0.005442	0.005059	0.578774	0.911484	0:00:00.027622 elapsed

Some of these achieve higher accuracy by correctly predicting the negative case, so the F1 score is a better indicator. The GaussianNB model gives us the best F1 score (with priors calculated from the train set) of 58%. While this isn't great, it does do better than the average. So if we talk about *accuracy*, we are very successful. If we talk about predicting True Positives, we are less successful.

Here are the results trying to predict 2017 nominees:



```
Classification report for classifier GaussianNB(priors=[0.03, 0.97]):
              precision    recall  f1-score   support

           0       0.98      0.78      0.87      3807
           1       0.09      0.61      0.15       130

    accuracy                           0.77      3937
   macro avg       0.53      0.69      0.51      3937
weighted avg       0.95      0.77      0.85      3937
```

Exploratory Analysis for Winners

A comparison of the different numeric categories across award entities was done to see in what ways they differed across winners.

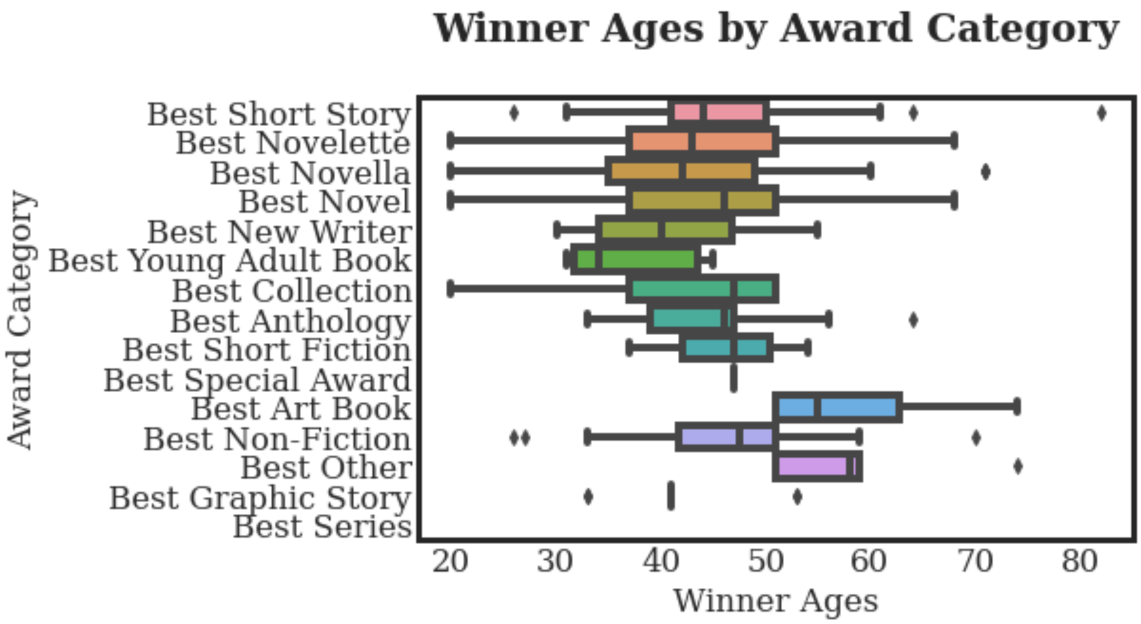
Breakdown Between Winners

Average for Each Book:	**All Winners**	**Hugo Winner**	**Nebula Winner**	**Locus Winner**
Average Rating	3.93	3.97	3.84	3.96
Ratings Count	36,694	39,207	27,775	40,073
Text Reviews Count	1,885	1,801	1,570	2,108
Number of Pages	328	288	244	395
Author Followers	15,973	12,042	5,874	23,943
Book Series	36%	34%	28%	41%

GoodReads Author	29%	27%	29%	30%
Age of Author	44.24	44.68	44.56	43.71
Female	29%	23%	38%	28%
Male	71%	77%	62%	72%
Prior Nominations for Author at Time of Win	9.14	8.05	5.86	11.61

The average winner exceeds the average nominee in every statistical category.

The age of winners was examined by award categories to see how the age distribution looks within each category:



We can see that there is a young age range for categories like Best Young Adult Book and Best New Writer which makes sense.

Best Novel Winners Breakdown

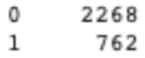
Average for Each Book:	**Best Novel Nominee**	**Best Novel Winner**	**Hugo Best Novel Winner**	**Nebula Best Novel Winner**	**Locus Best Novel Winner**
Average Rating	3.89	3.97	3.98	3.89	3.99

Ratings Count	54,966	111,026	129,199	92,449	108,611
Text Reviews Count	2,632	4,999	5,592	4,874	4,678
Number of Pages	401	439	418	372	486
Author Followers	20,110	23,226	17,290	10,426	33,535
Book Series	57%	63%	62%	57%	66%
GoodReads Author	34%	33%	31%	30%	37%
Age of Author	43.47	44.65	45.02	43.36	45.21
Female	25%	30%	24%	39%	29%
Male	75%	70%	76%	61%	71%
Prior Nominations for Author at Time of Nomination/Win	7.41	9.83	8.18	7.79	11.93

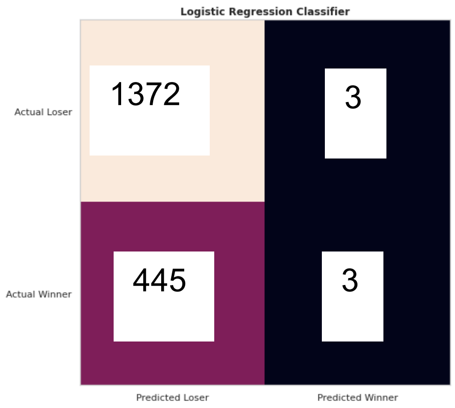
When looking at solely the Best Novel winners, it is clear that a book being a part of a book series plays a big factor (57%). This may indicate that voters possibly have a bias towards authors and books they are already familiar with.

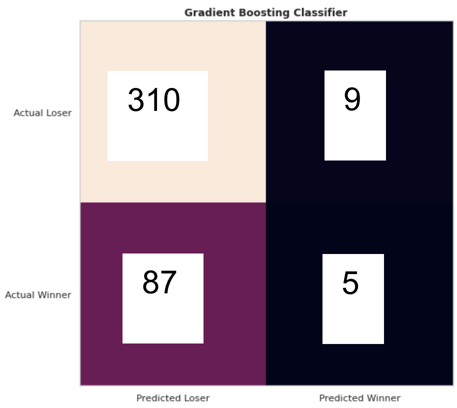
Predicting Winners

Modeling winners proved to be more challenging due to the imbalance in the data. Utilizing various sklearn algorithms resulted in the choice of "Loser" for the majority of predictions. This provided an accuracy of 75% but only 5 true positive predictions with the Gradient Boosting Classifier.



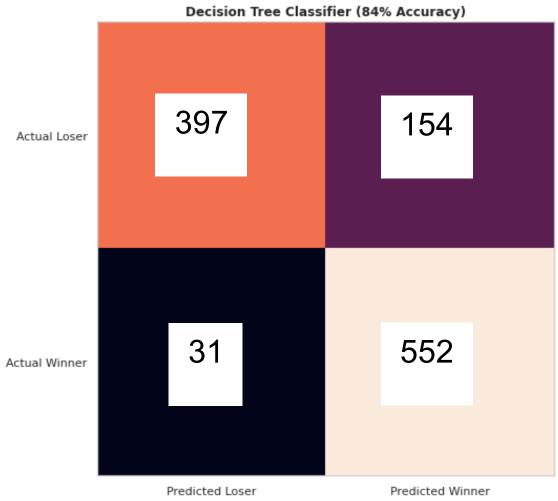
```
0      2268
1       762
```





After re-balancing the data, the ability to predict winners improved to 84% and the number of true positives increased to 552 with the Decision Tree Classifier.

```
1     2268
0     2268
```



Recommendations and Summary

Ultimately, the team was able to achieve the goal of predicting science fiction award nominees and winners using traditional machine learning algorithms in combination with new NLP techniques. Below is a summary of our results.

SciFiBook Awards	**Accuracy**
Predict Nominees (2007-2017)	75% (BERT) 92-97% (BERT + sklearn)
Predict Winners (2020)	84% (sklearn)

Stepping back, it makes sense that this is a difficult problem. This is the book that won this year's Nebula award. *A Song for a New Day* by Sarah Pinsker is about surviving a pandemic. That this would be of special interest in the current COVID19 crisis is the kind of topical knowledge that a machine learning model isn't going to pick up on in the same way a domain expert would.

We really enjoyed this puzzle, running into several challenges along the way. Spark was fast at doing the things it was good at, but then it totally failed when we wanted to do more dataframe-like operations. Colab was a convenient platform, but it regularly crashed on us and couldn't provide the resources we needed. SimpleTransformers is a very useful abstraction, but these transformers models require an expensive GPU to run successfully.

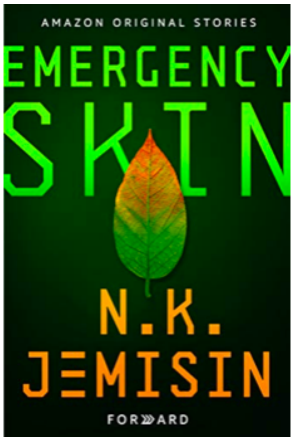
If we were to revisit this project with more compute, we might do better if we could:

1. Build our own genre filter from shelves. For this project we had to use the prebuilt ‘fantasy and paranormal’ dataset, which excluded a bunch of books that really should have been nominees.
2. Analyze text from all available reviews instead of sampling

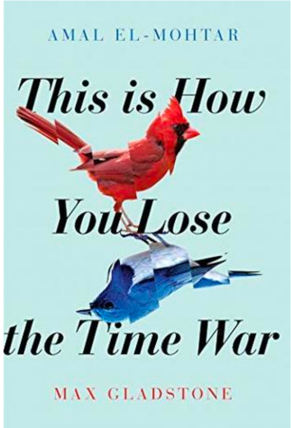
As compute resources become cheaper and faster, our ability to analyze the full text of all the reviews will become more feasible using transformer models (Roberta, Electra, Bert). Our team was only able to run our results on 50k samples and we achieved good results. As the cost and complexity of compute decreases there are many new applications where the combination of BERT + sklearn can be applied with great results.

Final thoughts and predictions

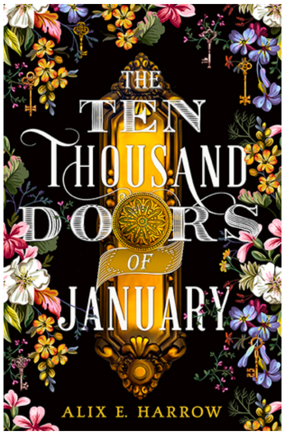
Although the awards for Hugo will not be announced until summer 2020, we wanted to provide our predictions on the winners based on our analysis. Please check the winners on August 1, 2020 to see how well we did with our predictions!



Best Novelette



Best Novella



Best Novel

Code Summary (see zip file)

- IST718_Project_BookReviews_MergingDatasets.ipynb
  - For merging BooksFantasyParanormal, ReviewsFantasyParanormal and Awards
  - Run on local computer due to RAM requirements (Colab crashes)
- IST718_Project_BookReviews_V3.ipynb
  - Spark code, visualizations, Goodreads API
  - Runs on Google Colab
- scraper_main.py and scraper_settings.py
  - Scrapes current Goodreads reviews by book ID
- Compare-gender-nominees.ipynb
  - Compares nominee/winners by gender for each of the three awards. Runs on local computer in Jupyter Notebook
- PredictWinners.ipynb
  - Uses SimpleTransformers Roberta-base to try to classify winners/losers from goodreads reviews (unsuccessful). Runs in Colab.
- Predict-nominees-desktop.ipynb
  - Uses SimpleTransformers Electra-small to classify nominee/non-nominee. Runs on local computer in Jupyter Notebook
- Predict-nominees-colab.ipynb
  - Uses SimpleTransformers Roberta-base to classify nominee/non-nominee. Runs in Colab
- PredictingHugoWinners.ipynb
  - Uses decision tree classifier to predict the winners across nominees with an 84% accuracy, and then predicts the winners in the upcoming Hugo 2020 awards. Runs in Colab.
- GoodReadsBookScraper_NYT.ipynb
  - This contains the GoodReads book info scraper that was responsible for scraping all the nominees dataframe with stats such as average rating, ratings count, text reviews count, author followers and others. It also has the code for the NYT scrape of the NYT bestsellers API that ultimately was not used due to not enough matches with the nominees. Runs in Colab.
- Nominees_DF_Visualizations.ipynb
  - This file contains the majority of visualizations code included in the powerpoint and report along with all of the nominee subdataframe transformations and added feature columns. Runs in Colab.

- WorldsScrape.ipynb
  - This file contains the scrape of the 6,000 authors science-fiction authors from the Worlds Without End database (http://worldswithoutend.com/authors_index.asp). This was the main source for all the added demographic info to the nominees dataframe such as gender, occupation, age, and USA region. Runs in Colab.

References and Data sources

Here are tools, packages, and references that informed our approach:

- https://sites.google.com/eng.ucsd.edu/ucsdbookgraph/home
- Devlin, J., Ming-Wei, C., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. Ithaca: Cornell University Library, arXiv.org. Retrieved from https://search-proquest-com.libezproxy2.syr.edu/docview/2118630252?accountid=14214
- https://gluon-nlp.mxnet.io/examples/sentence_embedding/bert.html
- https://github.com/ThilinaRajapakse/simpletransformers
- https://github.com/huggingface/transformers
- https://simpletransformers.ai/ https://medium.com/swlh/simple-transformers-multi-class-text-classification-with-bert-roberta-xlnet-xlm-and-8b585000ce3a

Nominees Dataframes for predicting winners:

nom_awards = pd.read_csv('https://www.dropbox.com/s/1mll9m9r27wy5iz/nominees_stats_awards_all.csv?dl=1') #this is the original GoodReads scraped dataset without additional demographic info.

nom_awards2 = pd.read_csv(https://www.dropbox.com/s/ysuekksts86beag/final_nominations_df.csv?dl=1'') #this is the version with the merged demographic info from Worlds Without End scrape.

Data Dictionary for Nominees Dataframe

Column	Description	Data Type
bookID	GoodReads.com book ID	int
asin	Amazon Standard Identification Number for book (usually same as ISBN).	object
isbn	International Standard Book Number for book.	object
bookTitle	Title of book	object
authorName	Author name	object
bookLink	www.goodreads.com link	object
format	Format of version of book that was scraped from GoodReads (ebook, hardcover, etc).	object
num_pages	Number of pages in book	int
languages	Language of book	object
date_published	Full date of when book published	datetime
publication_year	Year that book published	int
publisher	Publisher name of book	object
description	Book description/blurb scraped from GoodReads	object
author_bio	Author biography scraped from GoodReads	object
gender	Gender of author, scraped from www.worldswithoutend.com database of 6,000 authors. Male and female were later turned into binary columns for logistic regression analysis.	object
average_rating	Average rating of book on GoodReads.	float
ratings_count	Total ratings count of book on GoodReads	int
text_reviews_count	Total text reviews count of book on GoodReads.	int

author_followers	Total count of the amount of author's followers on GoodReads. Scraped as a possible way to gauge author internet following and popularity.	int
goodreads_author	1 or 0, 1 indicates the author has an official GoodReads profile.	int
book_series	1 or 0, 1 indicates book is part of a book series.	int
award	Original award column scraped from initial awards dataframe.	object
Nominee	1 or 0, 1 if book is Nominee. This was intended as a column to add to a larger book dataframe so could differentiate between nominees and non-nominee books.	int
Winner	1 or 0, 1 indicates a winner of award.	int
AwardCategory	This is a column that indicates the type of award category the book is nominated in (these categories were also added as binary columns but not used since did not find useful in models).	object
AwardEntity	This indicates the nomination was either from Hugo, Nebula, or Locus.	object
PastNominations	This is the amount of prior nominations to the year nominated for an author.	int
born	This is the date of when the author was born, scraped from www.worldswithoutend.com	object
occupation	This is the occupations identified by the author, scraped from www.worldswithoutend.com	object
nationality	This is the nationality of the author, scraped from www.worldswithoutend.com	object
BirthYear	This is the birth year of the author.	int
StateBorn	If applicable, this is the state in the USA the author is from.	object
AgeNominated	This is the age of the author at the time of nomination.	int

CountryBorn	This is the country where the author was born.	object
USA_Region	This is the region in the USA where the author was born.	object
Northeast	1 or 0, 1 indicates author from Northeast region in the USA.	int
West	1 or 0, 1 indicates author from West region in the USA.	int
South	1 or 0, 1 indicates author from South region in the USA.	int
Midwest	1 or 0, 1 indicates author from Midwest region in the USA.	int