Predicting the Genres of Over 70,000 Books on GoodReads

Haley Johnson & Sonali Pai

Project Motivation

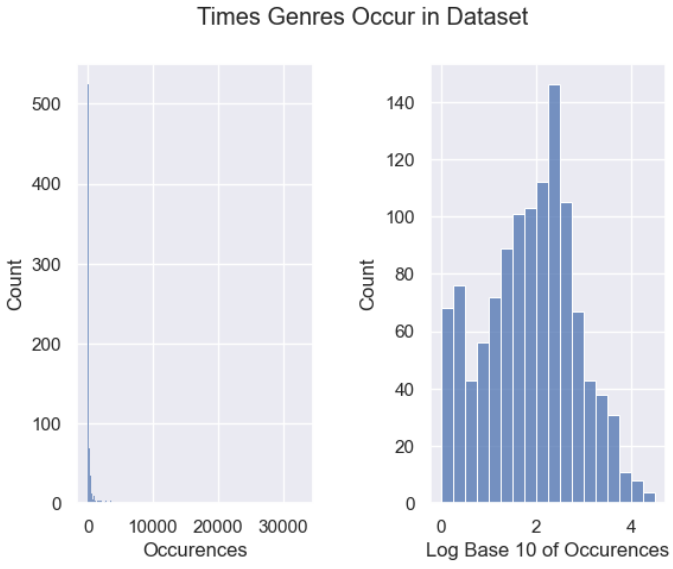
What distinguishes one genre from another? How meaningful are these distinctions and do they hold up for works that blend different literary styles?

GoodRead is a popular website for readers to rate and review books. This project uses a dataset of information scrapped from GoodReads last year to classify books by their primary genre. We also leverage dimensionality reduction techniques to analyze which genres are most similar. This work contributes to the growing fields of cultural analytics and digital humanities, which use computational methods to examine and model cultural phenomena.

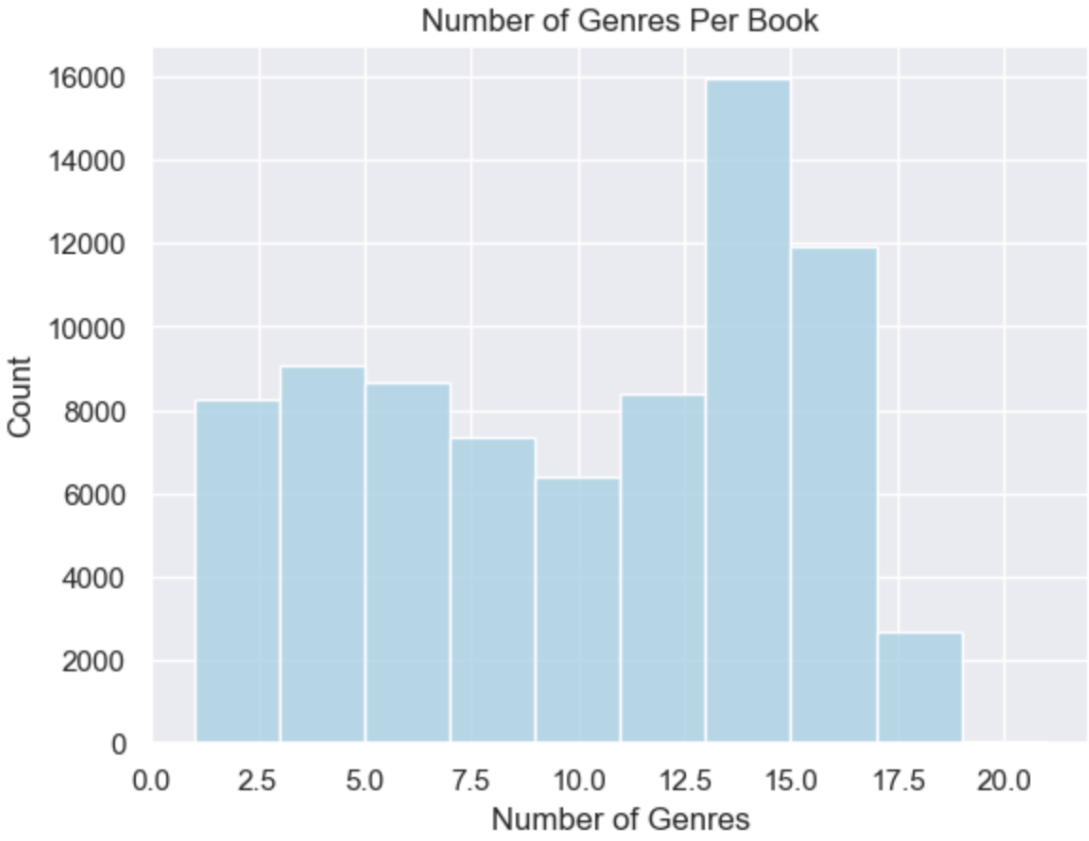
Analysis

Preliminary Analysis

There are 1,173 unique genres in the dataset. The median number of times a genre appears is 90, but the average number of times is 641.67, indicating that the distribution of genre occurrences has a strong rightward skew. A graphical analysis confirms that a handful occur extremely often in the dataset:



In this dataset, books were assigned to multiple genres. In fact, just 3.25% of entries were associated with only one genre.



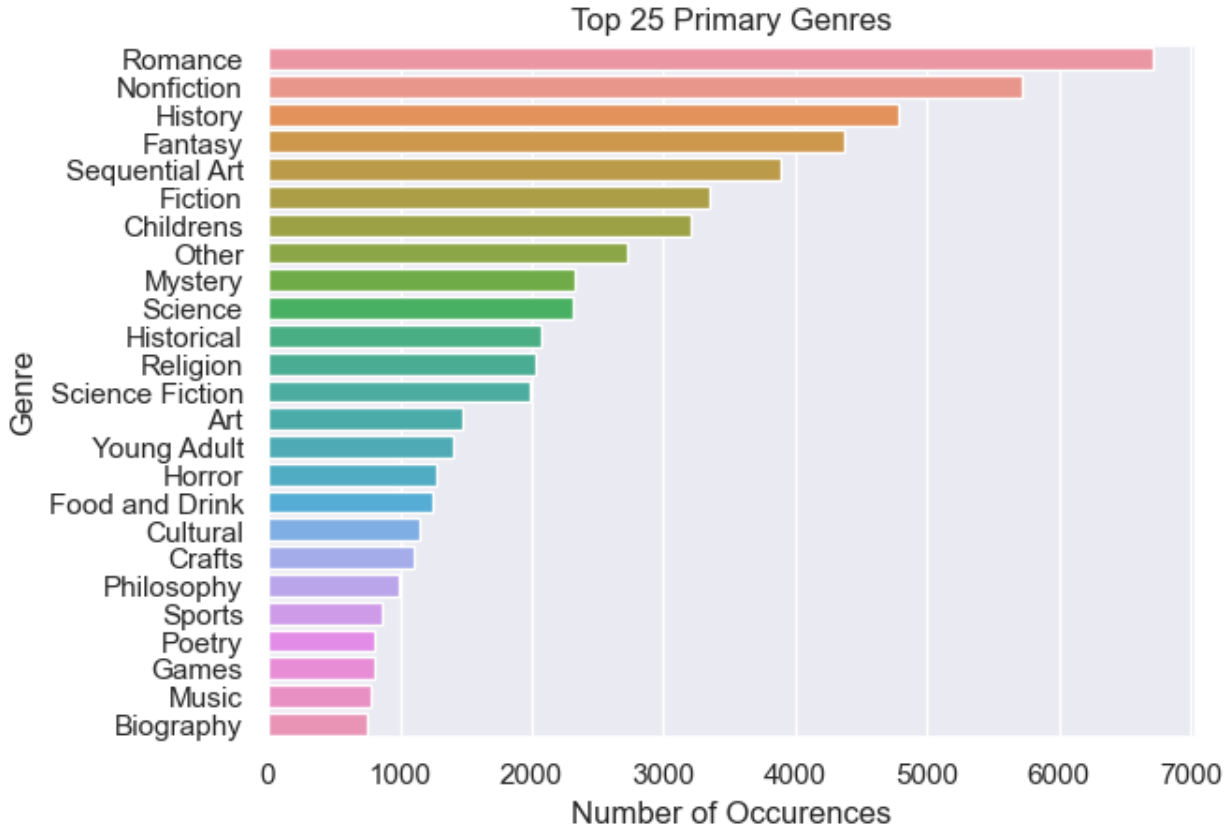
This project focuses on predicting the book's primary genre. Since genres are listed in order of their relevance, we define the primary genre as the first one given.

```
df.head()
```

	author	bookformat	desc	genre	img	
0	Laurence M. Hauptman	Hardcover	Reveals that several hundred thousand Indians ...	History,Military History,Civil War,American Hi...	https://i.gr-assets.com/images/S/compressed.ph...	(
1	Charlotte Fiell,Emmanuelle Dirix	Paperback	Fashion Sourcebook - 1920s is the first book i...	Couture,Fashion,Historical,Art,Nonfiction	https://i.gr-assets.com/images/S/compressed.ph...	1

For example, in row 0 'History' is the primary genre because it is listed first. Likewise, in row 1, 'Couture' is the primary genre. We would consider 'Fashion' and 'Historical' to be other genres that are associated with row 1.

Romance was by far the most common primary genre in the dataset, followed by nonfiction, history and fantasy. 2730 entries had their primary genre classified as 'other.'



Data Cleaning

There were several non-English entries in the dataset. We used a Python implementation of langdetect, a package originally developed in Java by Google, to remove non-English entries from the dataset. Books where the genres or descriptions were missing were also dropped, since these were the columns we would be using to build our classifier. This left us with a final dataset of 78,722 books.

```
# keep results consistent across runs
DetectorFactory.seed = 42

def get_lang(s: str):
    try:
        return detect(s)
    except:
        return 'unable to detect language'

df['desc_language'] = df.desc.apply(get_lang)
df = df[df.desc_language == 'en']
```

Next, we normalized the tests and removed any punctuation, non-ASCII characters, and white space. The text was broken into tokens and lemmatized using NLTK so that variants of the same word (e.g. running vs ran) were not treated as different words.

Stopwords were removed by the TF-IDF vectorizer.

```
# normalize & remove whitespace
df['desc'] = df.desc.str.lower().str.strip()

# remove quotes at start of description
df['desc'] = df.desc.str.replace("^'", "", regex = True)
df['desc'] = df.desc.str.replace('^"', "", regex = True)

# remove quotets at end of description
df['desc'] = df.desc.str.replace("'$", "", regex = True)
df['desc'] = df.desc.str.replace('"$', "", regex = True)

# fix apostrophe's
df['desc'] = df.desc.str.replace("\'", "'", regex = True)

# remove non-ascii characters (takes a while to run)
punc = """ '",{}[].`;:  !? —"""
valid_chars = df.desc.apply(lambda s: [w for w in s if ((w in string.ascii_letters) or (w in string.digits) or (w in punc))])
df['desc'] = valid_chars.apply(lambda s: "".join(s))

# lemmatize words
lemmatizer = WordNetLemmatizer()
df['tokens'] = df.desc.apply(word_tokenize)
df['lemmas'] = df.tokens.apply(lambda s: [lemmatizer.lemmatize(w) for w in s])
df['desc'] = df.lemmas.apply(lambda s: " ".join(s))
```

Clustering Genres Based on TF-IDF Vectors

Text data is, by definition, a high dimensionality space. While analyzing text data can provide valuable insights about what distinguishes one genre from another, it can also be difficult to work with due to the sheer size of the data. Our discussions of dimensionality reduction and clustering in class focused on numeric data — this project attempts to use some of these same principles to identify what genres are similar. This analysis was inspired by this online textbook and this tutorial.

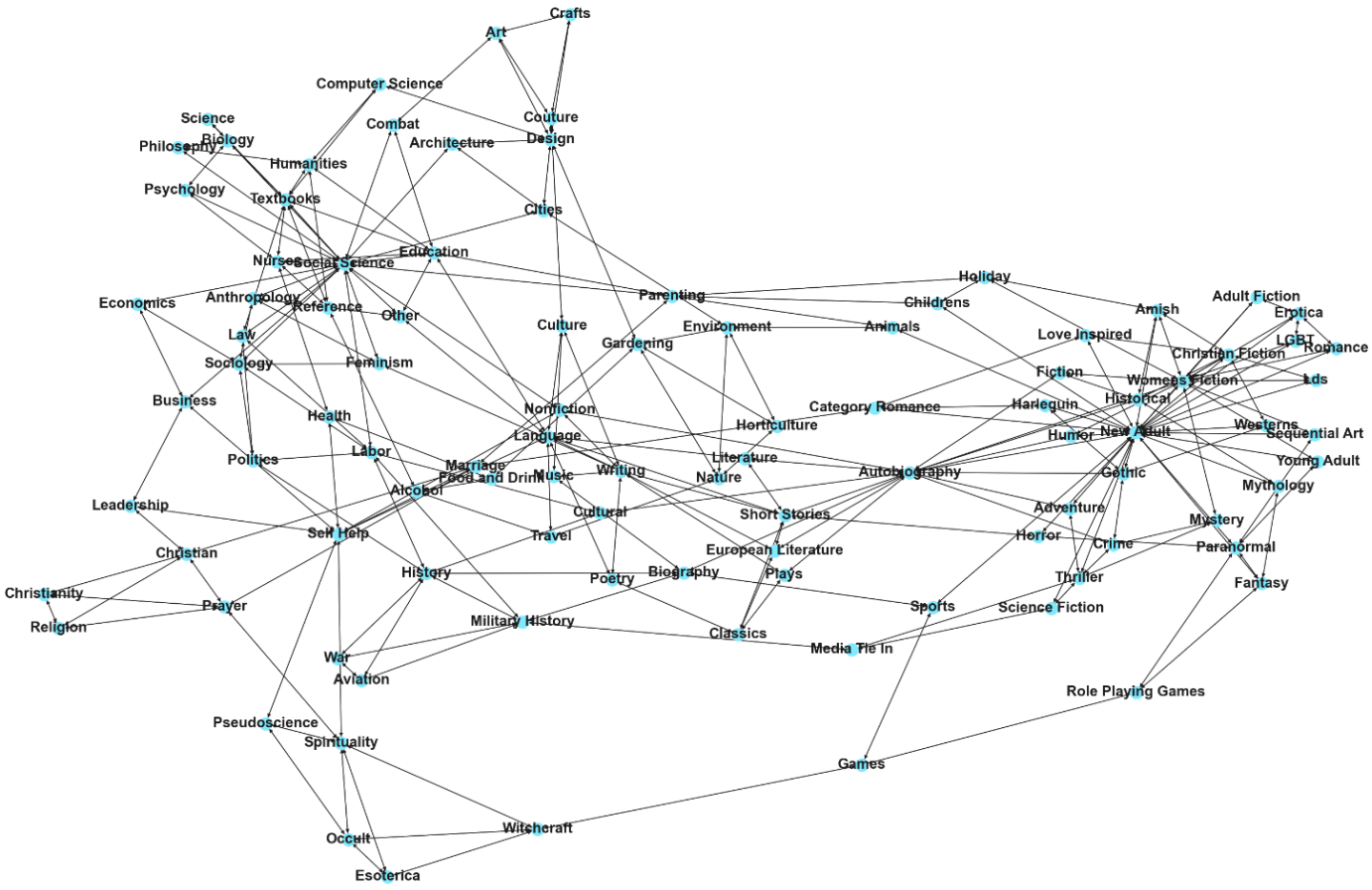
```
tfidv = TfidfVectorizer(smooth_idf = True, lowercase = True, analyzer = 'word', stop_words = 'english', max_features = 10000)

X_train_tfidv = tfidv.fit_transform(X_train.desc)
X_test_tfidv = tfidv.transform(X_test.desc)
```

Term frequency-inverse document frequency (TF-IDF) is "a method that tries to identify the most distinctively frequent or significant words in a document." We utilized TF-IDF vectors to find which words were most indicative of each genre. The TF-IDF vector was restricted to 10,000 features to speed up computing time.

There are many entries for each genre. We computed a TF-IDF vector for each book description, then took the average value of each word for every genre in the dataset. Genres were considered to be 'similar' when their average TF-IDF vectors had high cosine similarities.

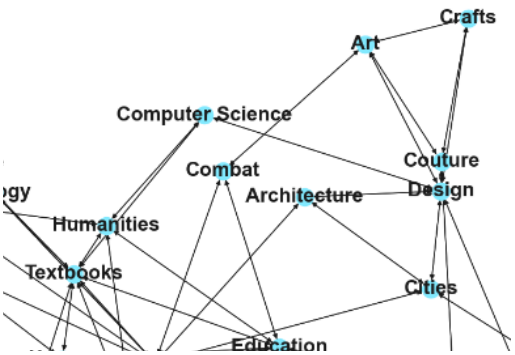
```
similar_dict = {}
for genre in genres:
    cosine_similarities = linear_kernel(np.array(all_terms.loc[genre]).reshape(1, -1), all_terms).flatten()
    similar_idx = cosine_similarities.argsort()[:-5:-1]
    similar_genres = [keys[idx] for idx in similar_idx if keys[idx] != genre]
    similar_dict[genre] = similar_genres
```

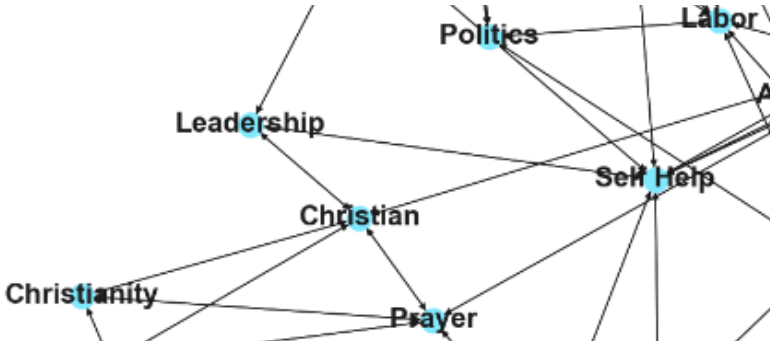
Using the cosine similarities, we mapped the top 3-4 most similar genres for each genre in the dataset:



Note that this network is directed. Genre A may have a high cosine similarity to Genre B, but Genre B may not be similar to Genre A.

This network reveals some interesting findings. For example, the average TF-IDF vector for books about computer science was similar to design books. Books about leadership were also similar to books about Christians and self-help books.






```
similar_dict['Computer Science'], similar_dict['Leadership']
```

```
(['Design', 'Textbooks', 'Humanities'], ['Business', 'Christian', 'Self Help'])
```

Classification

Logistic Regression

We fit a simple logistic regression model using TF-IDF vectors as inputs. This model is meant to serve as a baseline for comparison against more sophisticated methods. Logistic regression achieved 54.81% accuracy on the test data — in other words, the logistic model is doing much better than just randomly guessing genres.

```
lr = LogisticRegression(max_iter = 200, random_state = 42, solver = 'sag')
lr = lr.fit(X_train_tfidv, y_train)
```

```
y_pred = lr.predict(X_test_tfidv)
accuracy_score(y_test, y_pred)
```

```
0.5480920684924546
```

Moreover, since we are interested in understanding if there are meaningful differences between genres, we also implemented a custom accuracy function. This function looks at cases where the primary genre was misclassified and determines if that predicted genre appears anywhere in the full list of genres associated with the book. Even if the primary genre was labeled incorrectly, predicting another one of the book's genres is more accurate than predicting an unrelated one.

For 27.35% of entries, the classifier predicted a genre that was associated with the book but wasn't its primary genre.

Extra Trees Classifier

The next method we tried was an Extra Trees Classifier. This model was chosen because it is relatively simple and interpretable.

```
extra_trees = ExtraTreesClassifier(n_estimators = 250, random_state = 42)
extra_trees.fit(X_train_tfidv, y_train)
```

```
ExtraTreesClassifier(n_estimators=250, random_state=42)
```

```
extra_trees.score(X_test_tfidv, y_test)
```

```
0.47045373710685434
```

Unfortunately, this classifier performed worse than the logistic regression model. It classified 47.05% of books correctly. In 28.97% of cases, the classifier predicted a genre that was associated with the book but was not its primary genre.

The most important features were extracted from the Extra Trees Model. Many of these features were fairly generic words that could appear in any book description (i.e. 'novel,' and 'author'). In other cases, words like 'history' and 'poem' directly indicated what the genre was. This suggests that authors are directly signaling what genre a book is in the descriptions, perhaps because readers want to know this information when they're deciding whether or not to pick it up.

```
features_importance = pd.DataFrame(data = {'features': tfidv.get_feature_names_out(),
                                           'importance': extra_trees.feature_importances_}
                                   )
```

```
features_importance.sort_values(by = 'importance', ascending = False)[:25].values
```

```
array([['book', 0.00298722928157785],
       ['history', 0.0026984541322315 17],
       ['ha', 0.0026374188477736723],
       ['life', 0.0025762632294643816],
       ['novel', 0.002491813030417834],
       ['recipe', 0.002440761193606558],
       ['story', 0.0024301672885690133],
       ['collecting', 0.0023950536730319863],
       ['world', 0.0022514067079497 04],
       ['new', 0.002215785189102138],
       ['man', 0.002191817121730177],
       ['love', 0.002190870710023597],
       ['wa', 0.0021793449800811453],
       ['war', 0.0018932459833823291],
       ['time', 0.0018735398023163928],
       ['poem', 0.0018071708443603506],
       ['year', 0.0016917516726854722],
       ['work', 0.0016737036155492725],
       ['woman', 0.0016280128560480292],
       ['art', 0.0016133012439975426],
       ['way', 0.0015868272379583564],
       ['author', 0.0015852749689841648],
       ['make', 0.0015465586391883032],
       ['heart', 0.0014818383932714 11],
       ['child', 0.0014587246436357302]], dtype=object)
```

Deep Learning Classifier

Due to the poor performance of the Extra Trees Classifier, we turned our attention to more sophisticated machine learning methods, with the hope that they'd be able to detect more subtle differences between genres.

The TF-IDF vectors were fed into a deep-learning neural network, which was created using TensorFlow. Two features were added to the model to prevent overfitting: dropout layers, which randomly remove a percentage of the nodes from a neural network, and early stopping, which ends the mode's training early if it doesn't make sufficient

improvement between epochs.

```
neural_network = Sequential([
    tf.keras.Input(shape = (NUM_INPUT_FEATURES)),
    layers.Dense(512, activation = 'relu'),
    layers.Dense(256, activation = 'relu'),
    layers.Dropout(0.25),
    layers.Dense(256, activation = 'relu'),
    layers.Dropout(0.1),
    layers.Dense(NUM_CLASSES, activation = 'softmax')
])
```

```
opt = tf.keras.optimizers.Adam(learning_rate = 1e-5)
neural_network.compile(optimizer = opt, loss = 'categorical_crossentropy', metrics = ['accuracy'])
```

```
early_stopping = tf.keras.callbacks.EarlyStopping(monitor = "accuracy", min_delta = 0.005, patience = 1)
```

The model trained for 33 epochs before the early stopping callback ended training and achieved an accuracy of 48.71% on the testing dataset.

```
neural_network.evaluate(X_test_nn, y_test_nn)
```

```
616/616 [==============================] - 9s 13ms/step - loss: 2.0165 - accuracy: 0.4871
[2.016467332839966, 0.48711955547332764]
```

Due to the way softmax outputs predictions, we could not calculate how often the model predicted a book's secondary genre instead of its primary one.

Unfortunately, the deep learning model didn't perform significantly better than the Extra Trees Model and required much more computational power to train and make predictions.

Summary

Model	Accuracy	Secondary Accuracy*
Logistic Regression	54.81%	27.35% overall / 60.52% of errors

Extra Trees	47.05%	28.97% / 54.71% of errors
Deep Learning	48.71%	–

* Secondary Accuracy: If the primary genre label is incorrect, does it appear anywhere in the book's list of genres?

Key Findings

The results from our dimensionality reduction analysis suggest there is a high degree of overlap in how book's from different genres are described. Even genres that initially seem like they'd be quite distinct — such as computer science and the humanities — had high similarities. The lackluster results from the three classifiers our project fitted further suggest that it's difficult to distinguish between genres. Likewise, in the majority of cases where the logistic regression and extra trees models made errors, it predicted one of the book's secondary genres. This is more accurate than a complete misclassification and indicates that the model is learning something about these examples.

Limitations

In this dataset, books have a median of 10 genres assigned to them. Our project initially wanted to develop a multi-label classification model to predict up to 5 genres for each book. Due to technical difficulties, we revised the scope of our project and developed a model that only predicted the primary genre for each book. Our results show that there are similarities between genres; a multi-label classification model is more appropriate for this task and would likely achieve better results.

Furthermore, we only considered the 100 most common primary genres in the dataset. There were 271 unique primary genres and the 100 most common accounted for 96.5% of entries in the dataset. If a book has an uncommon primary genre, it was reclassified as 'other'. Therefore, this classifier will likely achieve lower accuracy on datasets composed of books from uncommon genres.

Finally, it's not clear what (if any) sampling procedure was used to collect the GoodReads dataset. There are significant class imbalances in this dataset — these may reflect the actual distribution of genres in the population of books or biases in the way the dataset was collected. If the dataset does not represent a random sample of titles from GoodReads, then our insights may have low generalizability.

Future Work

Our dataset contained a ‘title’ column. We considered trying to predict a book’s genre based on their titles alone but believed their descriptions would contain more useful information. Future work on this dataset should examine if a high level of accuracy can be achieved just by using the title, or if combining information from the title and the book’s description improves the classifier's performance.