Semester Project for Advanced Topics in Machine Learning

Calida Pereira
Otto Von Guericke University
Magdeburg, Germany
calida.pereira@st.ovgu.de

Chandan Radhakrishna
Otto Von Guericke University
Magdeburg, Germany
chandan.radhakrishna@st.ovgu.de

Thi Linh Nham Dao
Otto Von Guericke University
Magdeburg, Germany
thi1.dao@st.ovgu.de

Priyanka Bhargava
Otto Von Guericke University
Magdeburg, Germany
priyanka.bhargava@st.ovgu.de

Rashmi Korpade
Otto Von Guericke University
Magdeburg, Germany
rashmi.koparde@st.ovgu.de

***Abstract*—Genre Classification is one of the most challenging tasks in the area of Natural Language processing. Fiction novels tend to belong to more than one genres, therefore classifying a book with only a single label can become a bit overwhelming because this would mean learning core features of different genres, that actually contributes in classifying them. We intend to approach this problem by extracting various textual features and word embedding from the Gutenberg corpus of 996 fiction novels and training classifiers like Random Forests, SVM and Decision Trees, thus ultimately choose the best model.**

I. Motivation and Problem Statement

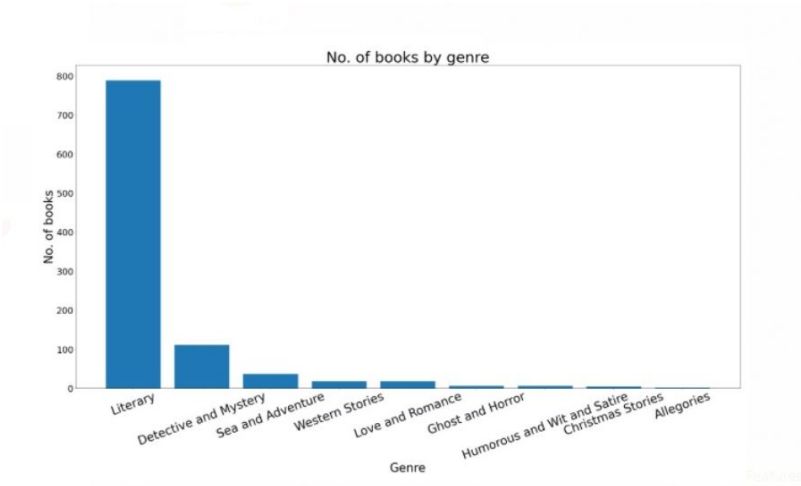
That's being said "There's no friend as loyal as a book". From centuries, books have been an inexhaustible source of information and a pleasure to it's readers.

In the early days any written document was considered a book and when mankind evolved, books were made from many different materials like beech bark, bamboo, clay, papyrus plant etc. From using parchments to papers, books saw a great deal of change. It was not until the Gutenberg printing press, that books really caught on. The main intention of Johannes Gutenberg was to make the books available for all. With this, people started to began the journey of true armchair travellers.

It's been decades and we now exist in the era of digitization, where there's plethora of knowledge related to everything. There are Fiction and Non-fiction books each of which have different genres under them. Genres are nothing but types that help differentiate amongst books, it's like an overall idea about a book. Novels are an example of fiction books, and they comprise mainly of characters, plots, story and genres. It's a normal human tendency to first look up for the genre of the book, before one actually starts reading it. Now we could do the genre classification task manually as well, but it would take loads of time and energy, also in the era of Machine learning why would one do it manually? Therefore, the main motivation behind this project, is to build a machine learning model, which helps us achieve this task of Genre classification. Every book is different from another, in terms of it's writing style, plot complexity, lexical richness and many other features which we'll be describing ahead. But there does lie similarities between different Genres. We intend to extract those similar features in such a way, that helps our model learn the core differences between Genres and ultimately helps us identify a Genre of a book that we don't know.

II. Data set

The dataset consists of a corpus of 996 books, derived from the 'Gutenberg Project' which is a voluntary organisation aimed at encouraging the creation and distribution of e-Books. Each book has a Book name, Book id, genre and Author name associated with it.



III. Concept

A. Notion of genres

The notion of genre is a very abstract concept. A Genre can be characterized by a particular style, form, content, rules and conventions that were developed overtime. If a book is deemed to be part of a genre, it doesn't mean it should not be part of any other genres. For example, 'Tarzan of the Apes' is classified as an Adventure novel but looking closely we see that the book is not just restricted to the adventure genre, it also consists of romance and mystery. Although classifying

the books into one single genre is a difficult task but we would restrict ourselves for identifying one genre for each book. Therefore multi-label classification is not the scope of our project.

B. BOW Model

The baseline or the ground truth for our model is the simple Bag of Words model. The ground truth model or the BOW model is compared with our Genre Classifier. The main purpose is to determine the enhanced classification results with respect to the newly extracted features.

C. Genre Classifier

Our proposed model called the Genre Classifier, classifies the books as per their genres using the features in Table 1.

TABLE I

Feature Type	***Feature Name***
Writing Style	Female Pronoun, Male Pronoun, Personal Pronoun, Possessive Pronoun, Preposition, Colon, Semi-colon, Hyphen, Interjection.
Sentence Complexity	Co-ordinating Conjunction, Comma, Period, Punctuation and sub-ordinating Conjunction, Sentence Length.
Sentiment	Negative, Positive, Neutral, Compound
Ease of readability	Flesch Reading Score
Plot complexity	Number of characters
Lexical richness	Type Token Ratio
Book encodings	Word2Vec encodings

D. Why these features?

1) Writing Style and Sentence Complexity: Every author has his/her own style of writing. It has been observed that most authors begin their careers by experimenting with different genres, and once they get a positive reception for a particular genre of book, they tend to write more books on the same or similar genres. This phenomenon can be seen with authors such as Arthur Conan Doyle, who wrote most of the books in Mystery, Adventure and Charles Dickens, who wrote most of the books in literary and Christmas stories.

We assume books written by an author belong to the same or similar genres and hence include Writing Style as a feature. The features are made up of Female Pronoun, Male Pronoun, Personal Pronoun, Possessive Pronoun, Preposition, Colon, Semi-colon, Hyphen, Interjection. Sentence Complexity can also be attributed to the style of the author but not restricted to it. Some genres predominantly have books that have long sentences coupled with more number of conjunctions and punctuation, this can be mainly seen in Literary fiction. Other novels with smaller sentence lengths with fewer conjunctions, and this can be mainly seen in Fantasy fiction.

2) Sentiment and Ease of readability: The sentiment of a book is how it makes readers feel about the content. A book may comprise of a mixture of many sentiments or there might be one dominant sentiment that rules throughout. In Jane Austen's work, Emma and Northanger Abbey have the most similar plots, with their tales of naive women who come to understand their own folly and it's how they learn to become mature and significant women that the books are all about. Mansfield Park, Persuasion and Jane Eyre from Charlotte Bronte seem similar too, all of these are more serious, darker stories with main characters being pitied upon at stages and the readers tend to feel sad as the story progresses. Persuasion also appears uniquely in starting out with neutral sentiment and then moving to more dramatic shifts in sentiments.

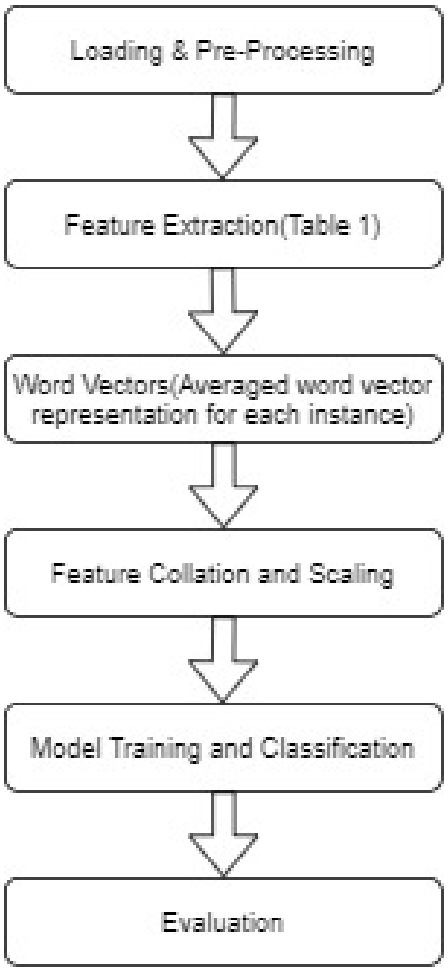
Ease of readability refers to the difficulty level of a passage. The difficulty level not only depends on the genres but also author's way of construction of sentences. Generally observed, children's novels tend to have a high score of readability which simply means the prose is easy to understand, whereas novels like Jane Eyre, Wuthering heights have a negative score making them a complex prose. Thus requiring high level of proficiency in order to be able to understand the novel. We intend to experiment if using ease of readability helps us in achieving our main objective that is classification of genres.

3) Plot Complexity: We've made a simple assumption that the more characters in the book, the more complex the plot is. Books belonging to mystery or adventure genres generally deals with more number of sub-plots and sometimes, sub-plot inside another sub-plot and this makes the book complex plot-wise. Other genres such as Christmas stories and fantasy genres are not so complex, since the audience in this case are the children. So we believe plot complexity could help in distinguishing genres.

4) Lexical Richness: Lexical Richness talks about the number of unique words used in the book. Books of History or literary genres are expected to contain more complex and distinct words due to their settings. Books of romance and horror are expected to generally have lower number of distinct words.

5) Book Encodings: Word embeddings are a vector representation of individual words, and Word2Vec is one way to generate these individual vectors for a word. Word2Vec aims to capture the semantic and syntactic similarity of words rather than the lexical similarity of the Bag Of Words model. We believe books with similar word encodings to belong to same or similar genres.

IV. Implementation



A. Books retrieval and pre-processing

Books are retrieved from the corpus and pre-processed. The Pre-processing pipeline involves retrieving the text from pre-converted HTML pages, tokenising the text and doing stop-word removal.

B. Features Extraction

1) Writing Style: Writing style has different sub-features like Female pronoun, Male Pronoun, Personal Pronoun, Possessive Pronoun, Preposition, Colon, Semi-colon, Hyphen, Interjection. The nltk RegexpTagger was used, which allows one to define our own word patterns for determining the part of speech tag. For example, the count of Female pronoun in a book was extracted by counting the no of occurrences of She/she/her/Her. Similarly it was done for other sub-features.

2) Sentence Complexity: Sentence Complexity also includes several sub-features such as Co-ordination Conjunction (words like: and, but, for, nor, or, so, yet), Comma, Period, Punctuation and Sub-ordinating Conjunction (words like: after, although, as, as if...), Sentence Length. These sub-features were created by counting the times that these specific words/characters appear in the whole book (except for Sentence Length sub-features). For Sentence Length, the value was calculated by taking the average sentence length of the book.

3) Sentiment and Ease of readability: Vader's sentiment analyzer was used for this purpose. We have taken the distinct words in the book and have run the sentiment analyzer on those words. The obtained sentiment scores were stored as separate columns in the Dataframe.

Flesch score was used for calculating ease of readability. The textstat package was used to determine readability score for each book. The range of score is not fixed and can take negative to more than 100 value as well. For a very complex prose a score of -100 is also valid whereas a simple text can range between 50 to 70. We ran the flesch score function on each raw content of the book. The scores obtained were stored as separate column in the dataframe.

4) Plot Complexity: Spacy's Named Entity Recognition tagger was used for this purpose. The original content of the book is passed through BeautifulSoup parser to remove HTML tags and this output is then passed to the NER tagger. The NER tagger tags every word in a sentence based on a named entity, such that "Steve" is tagged as "PERSON" and "Paris" is tagged as "LOCATION". This mechanism was leveraged to only retrieve words which were tagged as "PERSON". Once the parsing was done, the next step is to remove duplicate and improperly tagged words. We did that by assuming a character appears at least twice in the book and thereby removing all words that appeared only once in the book. Also we have created a character set to only accept distinct characters into the set and reject all duplicate characters. The number of distinct words present in this character set is then counted and added as a column in the Dataframe.

5) Lexical Richness: Lexical Richness is one of the most important features for any book as it characterises the diversity of vocabularies. This helps in knowing the complexity of a book as well. It's generally observed that the complexity of book increases with surge in it's lexical richness. We have obtained the Type Token Ratio for each of the book in order to represent it's Lexical richness. Type Token ratio was calculated by considering a ratio of unique words in a book to all the words in the same book. This is then fed to a column of our Dataframe where each row represents features for a book.

6) Book Encodings: Book Encodings were done using word vectors. So our main idea was to extract word vectors for each of the books. As we intended on having one vector representation for each book, we took an average of all the word vectors belonging to one book with respect to the unique number of words in that book. So the final outcome of Book encodings was a Dataframe having columns as dimensions that we'd fed during word2vec modelling, and each row belonged to one book. The word2vec model from the gensim package was used for this purpose.

As mentioned earlier we had extracted other features as well. So before feeding in the feature set to train our model, we had to collate all the textual features extracted separately with word vectors together in one dataframe. After bringing all the features under one shell it was observed that scaling

was required on the feature set as there were some features having their variance much larger than other features. If orders of magnitude of some features are larger than others, it might dominate the objective function and make the classifier unable to learn from other features correctly as expected. Therefore we sought help from the sklearn library which has a pre-processing scaling module.

V. Evaluation

The metric we're using for evaluating our models is the F1 score. As the data is highly biased towards one Genre meaning the maximum books in the corpus are of genre literary, accuracy will not be a good measure.

We have implemented BOW as our base model which we've used for comparison against the model we propose. The pipeline for our proposed model is explained in the flow chart. Bag of Words Model tends to overfit when applied on the raw content of the books, all for Random Forest, SVM and decision trees. The final labels that we observed were predicted to be literary for all the books across all the models.

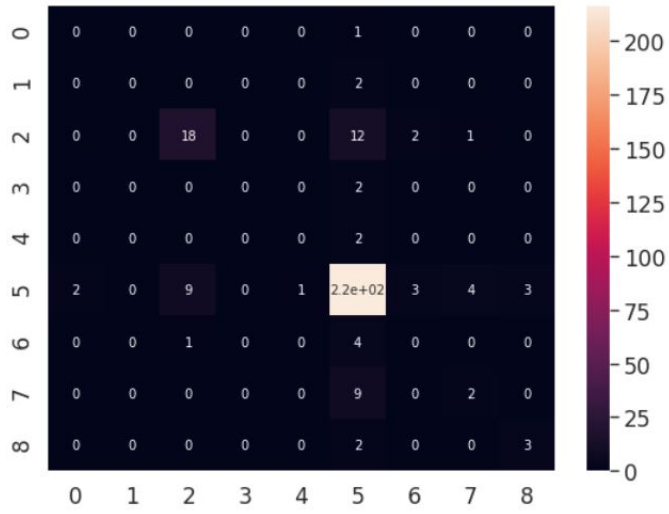
Looking closely at the extracted features, we observe that some of the features had "NAN" in them, so we had to remove those rows. Also since we had applied sentiment analysis on the entire book, the results were similar for almost all the books, so we had to remove them from the feature set as well.

In order to deal with the issue of class imbalance, we initially incorporated stratified sampling.The data was partitioned into training and testing sets in the ratio of 7:3. The f1 score generated was good but most of the predicted classes were "Literary" which is the dominant class. To mitigate this problem, we have tried doing over-sampling and a combination of over and under sampling. Even though the f1 score for over-sampling is lower than our baseline model, looking at confusion matrix we deduce that our model has performed better prediction, predicting classes other than just the dominant "Literary" class. The same thing is applicable for the combination of over and under-sampling model as well.

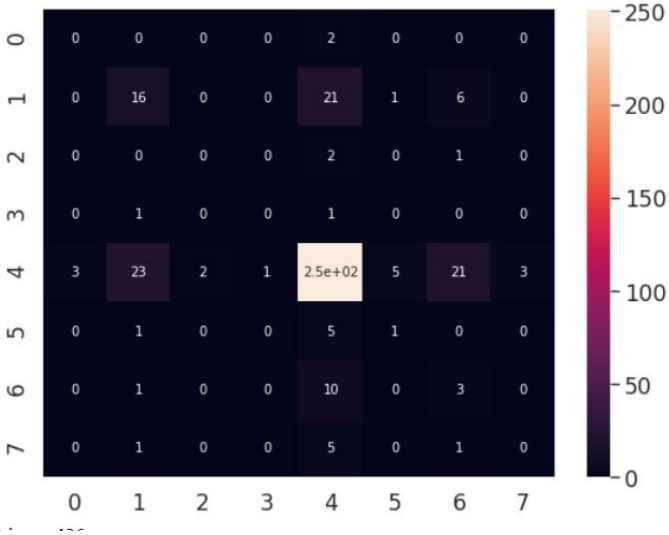
TABLE II

Models	**Random Forest**	**SVM**	**Decision Trees**
Bag of Words	0.75	0.73	0.79
Feature Extraction without Sampling	0.73	0.71	0.67
Feature Extraction with Over Sampling	0.74	0.28	0.72
Feature Extraction with Over and Under Sampling	0.78	0.19	0.61

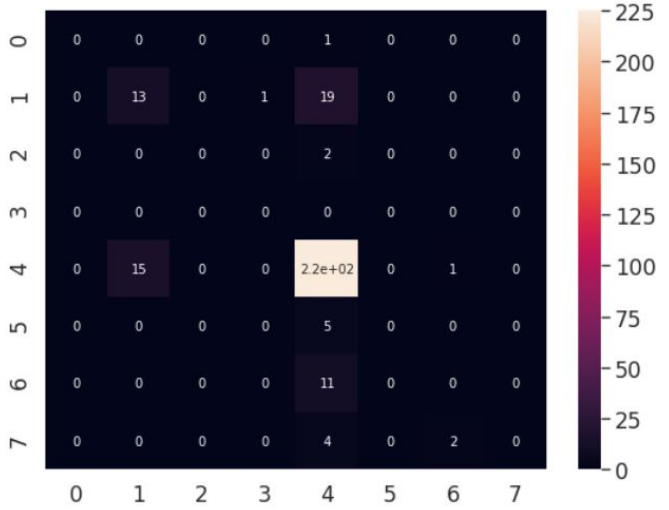
Bag of Words with Decision Trees



Over Sampling with Decision Trees



Over and Under Sampling with Random Forest



VI. Conclusion

Looking at the evaluation results, our extracted features along with oversampling performs the best with Decision Tree classifier, among the tried models. This was always bound to be a huge challenge since more than 3/4 of the dataset were ”Literary” books and its not always the case where books are classified into only one single genre. Even though our initial model failed to predict beyond the ”Literary” class, using oversampling helped prediction by predicting classes other than just the ”Literary” class. Dataset with better spread of classes could have resulted in better prediction results. We have extracted the features for an entire book in our approach, instead of this we could have divided the books into chunks and then performed the feature extraction on those chunks. This could have resulted in better feature extraction and may have led to better results. Also extracting topics through topic modelling could have greatly enhanced our predictive ability.

References

[1] Ben Guthrie, Jordan Henstrom, Ben Horrocks, Ryan West, “Determining Genre of Classical Literature with Machine Learning,” CS 478, Winter 2019.

[2] Holly Chiang, Yifan Ge, and Connie Wu, ”Classification of Book Genres”, CS 229 Fall 2015,Stanford University.

[3] Joseph Michael Worsham, “Towards Literary Genre Identification: Applied Neural Networks For Large Test Classification ,”B.S., University of Colorado Colorado Springs, 2014.

[4] Sayantan Polley,Suhita Ghosh,Marcus Thiel,Michael Kotzyba,Andreas Nurenberger “SIMFIC: An Explainable Book Search Companion.