Predicting Book Genre from Book Synopsis

Jainam Shah
Computer Science
Georgia State University
jshah23@student.gsu.edu

Abstract— This project predicts book genres using machine learning models based on book synopsis. Key steps include text preprocessing with cleaning and lemmatization, feature extraction using TF-IDF vectorization, and implementing Logistic Regression (84% accuracy), SVM (87%), and Random Forest (82%). The aim is to improve the accuracy and scalability of automated book genre classification based on the book's synopsis.

Keywords— Synopsis, Lemmatization, Logistic Regression, Support Vector Machine, Random Forest

I. INTRODUCTION

We often judge books by their covers or titles, but these first impressions can be misleading, as they do not always reflect the book's true content or genre. A synopsis, however, provides a clear summary of the story and themes, making it a more reliable way to understand what a book is about. By combining the insights from synopses with machine learning, we can accurately classify book genres and move beyond surface-level judgments.

The dataset is from Kaggle, and it consists of 8 columns and 1,539 rows with no missing values. The target variable is Genre and is classified into one of the following genres: thriller, fantasy, romance, horror, history, psychology, travel, science, sports, and science fiction.

So, the goal of my project is to develop an automated system that uses synopsis of books to predict the genre of the book.

There are three current methods to classify book genres. The first one is Image-Based Classification, which uses CNN models like AlexNet and VGG to extract features from book covers. It also uses Transfer learning with pre-trained models, such as ImageNet, which helps leverage prior knowledge. The next method is Title-Based Classification, which uses NLP techniques to extract features from titles, converting them into numerical vectors using word embeddings like GloVe and Word2Vec [1]. And the last method combining cover features from CNNs and title features from NLP. A logistic regression model then classifies genres using both sets of features for improved accuracy.

And the new method is classifying genres using the synopsis of the book.

II. RELATED PUBLICATIONS

Similar research has been done by using the book title and synopsis to predict the genre of the book. The researchers used K-Nearest Neighbor, Support Vector Machine, and Logistic Regression models [2]. The dataset they used consists of books consist of translated to English from Guajarati or Hindi originate books. Out of the 3 models, the most accurate model was SVM [3].

III. NEW METHOD

This method uses book synopsis to classify genres, focusing on content rather than titles or covers. The process begins with cleaning and preprocessing the text data by removing unnecessary characters, converting text to lowercase, and applying lemmatization. Next, the cleaned text is converted into numerical data using TF-IDF vectorization. Machine learning models like Logistic Regression, Support Vector Machines (SVM), and Random Forest are trained on these features to predict the genre of a book. Hyperparameter optimizes the models, ensuring better accuracy and scalability by analyzing the meaningful content from the synopsis.

IV. IMPLEMENTATION

1. Data Loading & Preprocessing: Read data.csv and select only the relevant columns (title, synopsis, genre). Display the first few rows using tabulate. Group the dataset by the genre column and count occurrences to understand the data distribution.

2. Text Cleaning: Implement text cleaning by removing unwanted characters, backslashes, converting text to lowercase, and removing extra spaces from synopsis.

3. Text Lemmatization: Lemmatize the words in the synopsis using WordNetLemmatizer to standardize variations of words.

   Lemmatization: process of converting a word to its base or dictionary form

Before Lemmatization

	synopsis
0	100,000 years ago, at least six human species ...
1	"Diamond has written a book of remarkable scop...
2	In the book, Zinn presented a different side o...
3	Author Erik Larson imbues the incredible event...
4	Discovered in the attic in which she spent the...
...	...
1534	Atticus O'Sullivan, last of the Druids, lives ...
1535	Charlie Bucket's wonderful adventure begins wh...
1536	"I live for the dream that my children will be...
1537	Rose loves Dimitri, Dimitri might love Tasha, ...
1538	The Prince of no value\nBrishen Khaskem, princ...

After Lemmatization

	synopsis
0	years ago at least six human species inhabited...
1	diamond has written a book of remarkable scope...
2	in the book zinn presented a different side of...
3	author erik larson imbues the incredible event...
4	discovered in the attic in which she spent the...
...	...
1534	atticus o sullivan last of the druids lives pe...
1535	charlie buckets wonderful adventure begins whe...
1536	i live for the dream that my children will be ...
1537	rose loves dimitri dimitri might love tasha an...
1538	the prince of no value brishen khaskem prince ...

4. Visualize Word Frequencies: Use nltk.FreqDist and seaborn to visualize the most frequent words across the cleaned summaries.

Before Lemmatization

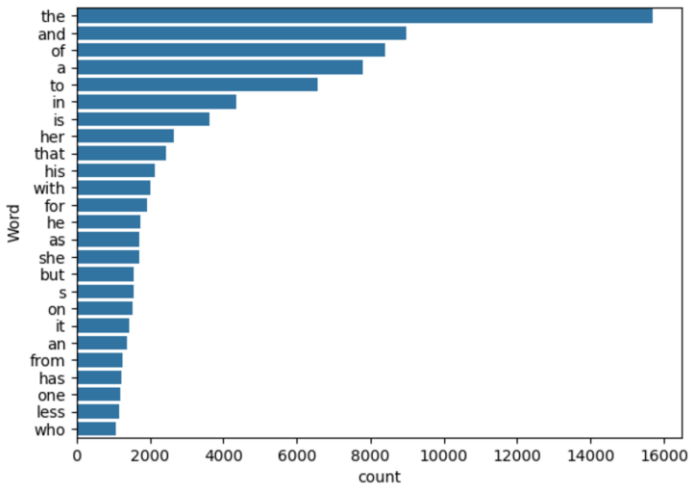

Figure 1. The frequency distribution of words before Lemmatization

After Lemmatization

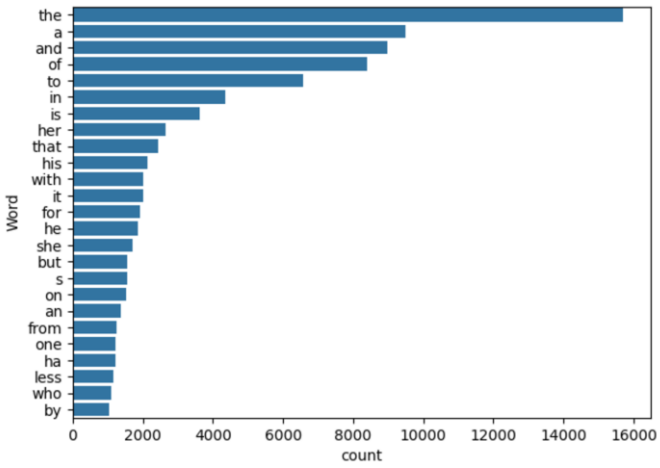

Figure 2. The frequency distribution of words after Lemmatization

5. TF-IDF Vectorization: Transform text data (synopsis) into numerical features using TfidfVectorizer for model compatibility. Include bi-grams (ngram_range= (1, 2)) for better contextual understanding.

6. Data Splitting: Split data into training and validation sets with an 80/20 split using train_test_split. Prepare features (X) and target (y) data.

7. Train Models:
   Logistic Regression using OneVsRestClassifier.

   Support Vector Machines (SVM) using svm.SVC(kernel='linear').

   Random Forest Classifier using RandomForestClassifier.

8. Model Evaluation: Evaluated models by using metrics like accuracy_score and classification_report. Compared Logistic Regression, SVM, and Random Forest for their performances on validation data before and after hyperparameter tuning.
   a. Logistic Regression: predicts genres by analyzing the relationship between word features and labels. It is efficient, works well with clear patterns, and is ideal for smaller datasets. While fast and simple, it may need feature adjustments to manage more complex relationships [4].
   b. SVM: Separates genres by finding the best boundary (hyperplane) in a high-dimensional space of word features. Excels with sparse, noisy data and manages complex decision boundaries. Performs well with clear genre separations and benefits from tuning parameters like the kernel type [5].
   c. Random Forest: Uses multiple decision trees to classify genres by identifying patterns in the synopsis. Manages non-linear relationships and provides insights into which words are most important for predictions. Works better with denser features and may require more data and computational resources compared to simpler models. Writes similar description in human language but for logistic regression [6].

9. Hyperparameter Tuning: Used GridSearchCV for hyperparameter tuning for Logistic Regression, SVM, and Random Forest models to find optimal parameters by using cross-validation method.

10. Visualize Results: Created confusion matrices and other relevant graphs for model analysis and comparison to better understand performance metrics.

11. Optimization using Feature Engineering: F-IDF vectorization was fine-tuned by incorporating bi-grams and setting a minimum document frequency (min_df) of 5, enabling the extraction of more meaningful and context-rich features from the text data.

12. Compare Models: Compare results from Logistic Regression, SVM, and Random Forest based on accuracy, classification reports, and confusion matrices.

This sequence uses preprocessing, text feature extraction, model selection, tuning, and evaluation to classify book genres based on their summaries.

V. RESULTS AND ANALYSIS

The data was split data into training (80%) and testing (20%) and ran the 3 models before and after Hyperparameter tuning.

Before Hyperparameter Tuning:

Logistic Regression:

Accuracy Score: 0.6298701298701299

Classification Report:

	precision	recall	f1-score	support
fantasy	0.78	0.93	0.85	69
history	0.71	0.28	0.40	18
horror	0.00	0.00	0.00	15
psychology	0.65	0.62	0.63	21
romance	0.00	0.00	0.00	24
science	1.00	0.29	0.44	21
science_fiction	0.00	0.00	0.00	5
sports	0.00	0.00	0.00	17
thriller	0.54	1.00	0.70	100
travel	1.00	0.33	0.50	18
accuracy			0.63	308
macro avg	0.47	0.34	0.35	308
weighted avg	0.56	0.63	0.54	308

Table 1. Logistic Regression accuracy Before Hyperparameter Tuning

SVM

Accuracy Score : 0.7727272727272727
Report :

	precision	recall	f1-score	support
fantasy	0.82	0.96	0.89	69
history	0.75	0.67	0.71	18
horror	1.00	0.13	0.24	15
psychology	0.86	0.86	0.86	21
romance	0.57	0.17	0.26	24
science	0.94	0.71	0.81	21
science_fiction	1.00	0.20	0.33	5
sports	1.00	0.59	0.74	17
thriller	0.69	0.99	0.81	100
travel	0.92	0.61	0.73	18
accuracy			0.77	308
macro avg	0.86	0.59	0.64	308
weighted avg	0.79	0.77	0.74	308

Table 2. SVM accuracy Before Hyperparameter Tuning

Random Forest

Accuracy Score: 0.6331168831168831

Classification Report:

	precision	recall	f1-score	support
fantasy	0.67	0.77	0.72	69
history	0.78	0.39	0.52	18
horror	0.00	0.00	0.00	15
psychology	0.72	0.62	0.67	21
romance	0.00	0.00	0.00	24
science	0.85	0.52	0.65	21
science_fiction	0.00	0.00	0.00	5
sports	1.00	0.24	0.38	17
thriller	0.57	0.96	0.71	100
travel	0.69	0.61	0.65	18
accuracy			0.63	308
macro avg	0.53	0.41	0.43	308
weighted avg	0.58	0.63	0.57	308

Table 3. Random Forest accuracy Before Hyperparameter Tuning

After Hyperparameter Tuning:

Logistic Regression:

Logistic Regression Accuracy Score: 0.7824675324675324

Classification Report:

	precision	recall	f1-score	support
fantasy	0.78	0.94	0.86	69
history	0.86	0.67	0.75	18
horror	0.70	0.47	0.56	15
psychology	0.85	0.81	0.83	21
romance	0.56	0.38	0.45	24
science	0.75	0.71	0.73	21
science_fiction	1.00	0.40	0.57	5
sports	1.00	0.71	0.83	17
thriller	0.77	0.89	0.82	100
travel	0.87	0.72	0.79	18
accuracy			0.78	308
macro avg	0.81	0.67	0.72	308
weighted avg	0.78	0.78	0.77	308

Table 4. Logistic Regression accuracy after Hyperparameter Tuning

SVM:

SVM Accuracy Score: 0.7662337662337663

Classification Report:

	precision	recall	f1-score	support
fantasy	0.80	0.96	0.87	69
history	0.80	0.67	0.73	18
horror	0.58	0.47	0.52	15
psychology	0.85	0.81	0.83	21
romance	0.44	0.33	0.38	24
science	0.79	0.71	0.75	21
science_fiction	0.67	0.40	0.50	5
sports	1.00	0.59	0.74	17
thriller	0.76	0.87	0.81	100
travel	0.86	0.67	0.75	18
accuracy			0.77	308
macro avg	0.76	0.65	0.69	308
weighted avg	0.76	0.77	0.76	308

Table 5. SVM accuracy after Hyperparameter Tuning

Random Forest

Random Forest Accuracy Score: 0.6883116883116883

Classification Report:

	precision	recall	f1-score	support
fantasy	0.77	0.72	0.75	69
history	0.70	0.78	0.74	18
horror	0.18	0.13	0.15	15
psychology	0.79	0.90	0.84	21
romance	0.41	0.38	0.39	24
science	0.77	0.81	0.79	21
science_fiction	0.50	0.60	0.55	5
sports	0.75	0.71	0.73	17
thriller	0.70	0.71	0.70	100
travel	0.75	0.83	0.79	18
accuracy			0.69	308
macro avg	0.63	0.66	0.64	308
weighted avg	0.68	0.69	0.68	308

Table 6. Random Forest accuracy After Hyperparameter Tuning

```
Logistic Regression Predictions:
Book: Sapiens: A Brief History of Humankind
Predicted genre: fantasy
Actual genre: fantasy
--------------------------------------------------
Book: Guns, Germs, and Steel: The Fates of Human Societies
Predicted genre: thriller
Actual genre: thriller
--------------------------------------------------
Book: A People's History of the United States
Predicted genre: romance
Actual genre: romance
--------------------------------------------------
Book: The Devil in the White City: Murder, Magic, and Madness at the Fair That Changed America
Predicted genre: thriller
Actual genre: thriller
--------------------------------------------------
Book: The Diary of a Young Girl
Predicted genre: romance
Actual genre: travel
```

Figure 3: Sample Logistic Regression Predictions after hyperparameter tuning.

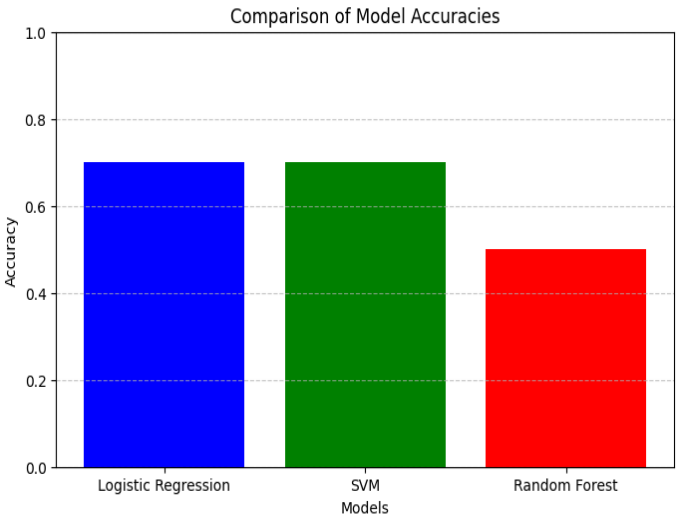

Figure 4. Accuracy of the 3 models After Hyperparameter Tuning

VI. CONCLUSIONS

	Before Hyperparameter Tuning	After Hyperparameter Tuning
SVM	0.7727	0.7662
Logistic Reg.	0.6299	0.7825
Random Forest	0.6312	0.6883

Table 7. Accuracy of the 3 models Before and After Hyperparameter Tuning

SVM achieved the highest accuracy (0.78), demonstrating its effectiveness in genre classification from book summaries before hyperparameter tuning.

After hyperparameter tuning, Logistic Regression had the highest accuracy of 0.78.

In the future, two features will be added. The first one is adding multilingual capabilities, so the model can manage book summaries in multiple languages. The next feature is highlighting which parts of the synopsis contribute most to the prediction.

REFERENCES

[1] What Is Text Classification? - Text Classification Explained - AWS, aws.amazon.com/what-is/text-classification/. Accessed 14 Dec. 2024.

[2] Shiroya, Parilkumar. “Book Genre Categorization Using Machine Learning Algorithms (K-Nearest Neighbor, Support Vector Machine and Logistic Regression) Using Customized Dataset.” International Journal of Computer Science and Mobile Computing, vol. 10, no. 3, 30 Mar. 2021, pp. 14–25, https://doi.org/10.47760/ijcsmc.2021.v10i03.002. Accessed 1 Apr. 2021.

[3] S. Li, “Multi-Class Text Classification Model Comparison and Selection,” Towards Data Science, Sep. 25, 2018. https://towardsdatascience.com/multi-class-text-classification-model-comparison-and-selection-5eb066197568

[4] “Text Classification Using Logistic Regression.” GeeksforGeeks, GeeksforGeeks, 4 Mar. 2024, www.geeksforgeeks.org/text-classification-using-logistic-regression/.

[5] Bedi, Gunjit. “Simple Guide to Text Classification (Nlp) Using SVM and Naive Bayes with Python.” Medium, Medium, 13 July 2020, medium.com/@bedigunjit/simple-guide-to-text-classification-nlp-using-svm-and-naive-bayes-with-python-421db3a72d34.

[6] Shafi, Adam. “Random Forest Classification with Scikit-Learn.” DataCamp, DataCamp, 1 Oct. 2024, www.datacamp.com/tutorial/random-forests-classifier-python.