iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** X **Month of publication:** October 2021

DOI: https://doi.org/10.22214/ijraset.2021.38409

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue X Oct 2021- Available at www.ijraset.com

Book Genre Prediction

Priyal Desai[1], Ghata Saraiya[2], Maliha Nan[3]
[1, 2, 3]*Computer Engineering Dept., Dharmsinh Desai University, Nadiad, Gujarat, India*

Abstract: The present work aims to classify the genre of the books automatically using the Python programming language. A genre is a subset of art, literature, or music that has a distinct form, substance, and style. In many instances, a book can be classified as belonging to more than one genre. It's difficult to categorize a book or piece of literature as belonging to one genre over another. Many novels end up badly categorized or pushed under the super-genre umbrella of fiction since there is no clear criterion to determine how much of a book belongs to a given genre. Therefore, it's critical to develop a system for categorizing books and determining their relevance to a particular genre. Therefore, the current study tries to solve this challenge by combining various text categorization approaches and models to come up with the best solution.

I. INTRODUCTION

Big Data technologies are intimately linked to the science of artificial intelligence. Natural language analysis is one of its specialties. Computers can be taught to recognize particular patterns in processed texts and classify words, phrases, or even entire documents into predefined groups based on these patterns. One can easily configure such a project utilizing open-source instruments, which are capable of classifying text based on a preceding automatic learning phase and preset input data.

Many natural language processing (NLP) machine learning algorithms include a statistical model, in which judgments are determined using a probabilistic approach. Deep learning algorithms have also been used recently, with excellent results. Text fragments make up input data, which can be basic word sequences, whole sentences, or even entire papers. The text is altered, and different data features are given varying weights.

Machine learning models are developed using the input data and can then be applied to fresh, unexpected data. In comparison to alternative linguistics models, such algorithms can learn from data and are better at interpreting new or erroneous input, such as spelling problems or missing words. Linguistic models are built on a set of established grammatical rules that are prone to errors when dealing with unfamiliar or wrong input, as well as being more difficult to maintain when dealing with big and complicated systems.

The accuracy of machine learning models is proportional to the amount of the input data. Providing additional texts from which the model can learn will boost the new processed data's prediction outcomes.

Natural Language Processing encompasses a wide range of disciplines, including part of speech tagging and named entity recognition (which aims to locate and identify named entities such as people, places, and organizations), machine translation, speech recognition, question answering, sentiment analysis, etc.

The present work focuses on using the Python programming language to forecast the genre of a book. This study is based on how books are classified based on their summaries. The proposed theory is that novels can be classified based on their written summaries' word content. The model will be used to categorize new books into predefined categories once it has been trained using a dataset. One of the long-term goals is to make book classification easier and to inform consumers about possible genres and genre overlap. This may make it easier to categorize novels as belonging to multiple genres. The classification of literary works differs greatly from the classification of ordinary texts. The length of books, which is far greater than most other text media, is one of the major reasons behind this. As a result, we'll be working with book summaries rather than the complete text.

II. LITERATURE REVIEW

Topic segmentation and recognition are two capabilities of Natural Language Processing. This necessitates a set of input data as well as some machine learning models capable of categorizing the text into various subjects.

Unsupervised and supervised learning are two approaches to the problem. Unsupervised algorithms employ input data that has not been hand-annotated with the correct class or topic, whereas supervised algorithms use data that has been labeled with the relevant class or topic. Unsupervised learning is generally more difficult and produces less accurate outcomes than supervised learning. Nonetheless, the volume of data that has not been labeled is much bigger than the data that has been allocated to the correct classes, and in some cases, an unsupervised approach is the only alternative.

Unsupervised NLP methods use machine learning clustering algorithms to divide text data into segments and identify the class of each group.

Surveillance algorithms, on the other hand, necessitate a set of textual data with the appropriate labels pre-filled. What is the best way to get a text that has been labeled? You can use data that has already been assigned to a category, such as movies, a genre, product categories, document categories, or comment subjects.

An NLP supervised classification algorithm will examine the input data and should be able to determine which topic or class a new text should belong to based on the current classes found in the train data.

To some extent, the level of confidence is proportional to the amount of accessible training data, as well as the similarity between the new incoming text and the ones from which the model has learned. The majority of algorithms additionally show the amount of confidence in a correct match, which is usually between 0 and 1. The user can choose a forecast accuracy criterion, which will lead to a decision about when to dismiss the result. This outcome can take the form of one or more classes determined by the model.

A supervised NLP algorithm's end-to-end flow includes acquiring labeled data, data preparation, developing a classification model, and utilizing the model to forecast the topic of a new text.

III. METHODOLOGY

Evaluating and discovering the appropriate dataset is a very crucial step. In the current study, two datasets were discovered that are discussed in the further subsection.

A. CMU Book Summary

The first dataset is “CMU Book Summary” which includes plot summaries for 16,559 novels collected from Wikipedia, as well as aligned metadata from Freebase, such as author, title, and genre. The dataset contains 179 distinct genres with a very asymmetric distribution. To balance the distribution, we only preserve mainstream genres and delete books from genres such as "anti-war." The number of books published in each genre continues to remain unbalanced. In order to make the distribution more even, we will use data augmentation in subsequent rounds. The number of genres kept, and their distribution are as follows.

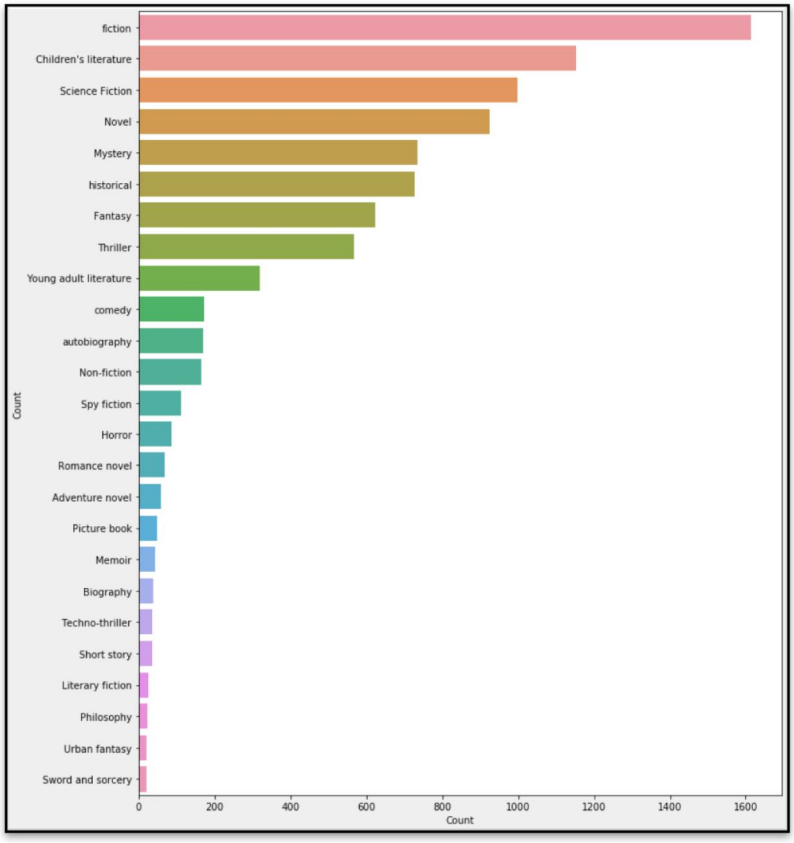


Figure 1: Distribution of the CMU Book Summary Dataset

1) *Data Preparation:* A list of documents made up of word sequences or entire phrases is the starting point for document classification. In their original form, they can't be used as machine learning features. These articles must be divided and turned into useful features for a machine learning model. In this concern, the Natural Language Toolkit was used to clean the summaries. The following steps were followed for the pre-processing process:

a) Conversion to lowercase
b) Eliminating the punctuation marks
c) Eliminating Stop Words
d) Word-to-number conversion
e) Stemming

2) *Data Augmentation:* In computer vision, data augmentation is often utilized. It may very likely flip, rotate, or mirror a picture in vision without risking affecting the original label. However, there is a distinction when working with text, particularly summaries. For data augmentation, the application of a single basic process that does not modify the book's category was done. Moreover, synonym replacement was done i.e., Pick n words from the sentence that doesn't stop words at random. Each of these words should be replaced with a synonym picked at random. According to the distribution, there will be an increase in records in genres with fewer records. Figure 2 depicts the distribution of the CMU dataset after the data augmentation process. It is to be noted that this distribution is more balanced than the previous distribution.

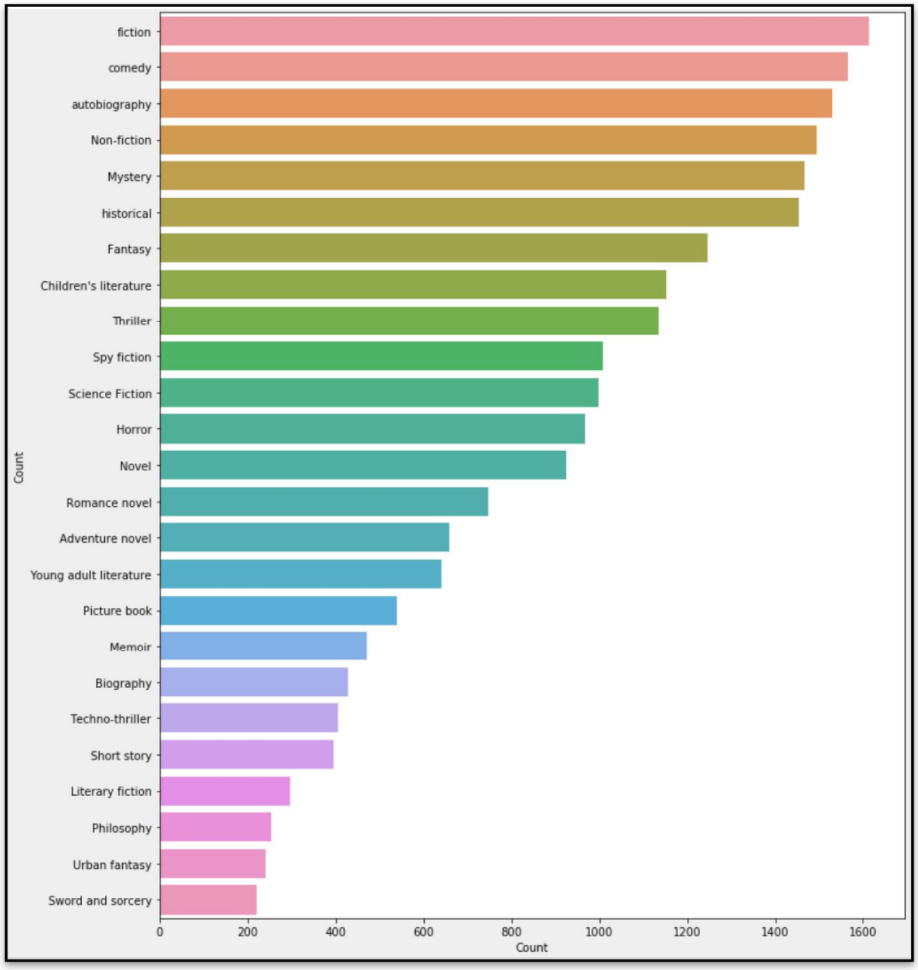


Figure 2: Distribution of the CMU dataset after data augmentation

The next step was to develop a classifier and fit the dataset. To perform that, a pipeline was utilized that vectorizes the data before it is fed to the classifier.

B. The Blurb Genre Collection

The second dataset is "The Blurb Genre Collection" which contains blurbs (advertising descriptions of books) and aligned metadata for 91,982 books collected from Penguin Random House. The data distribution of The Blurb Genre Collection is demonstrated in figure 3. Furthermore, multiple genres were assigned to each summary that is shown in figure 4.

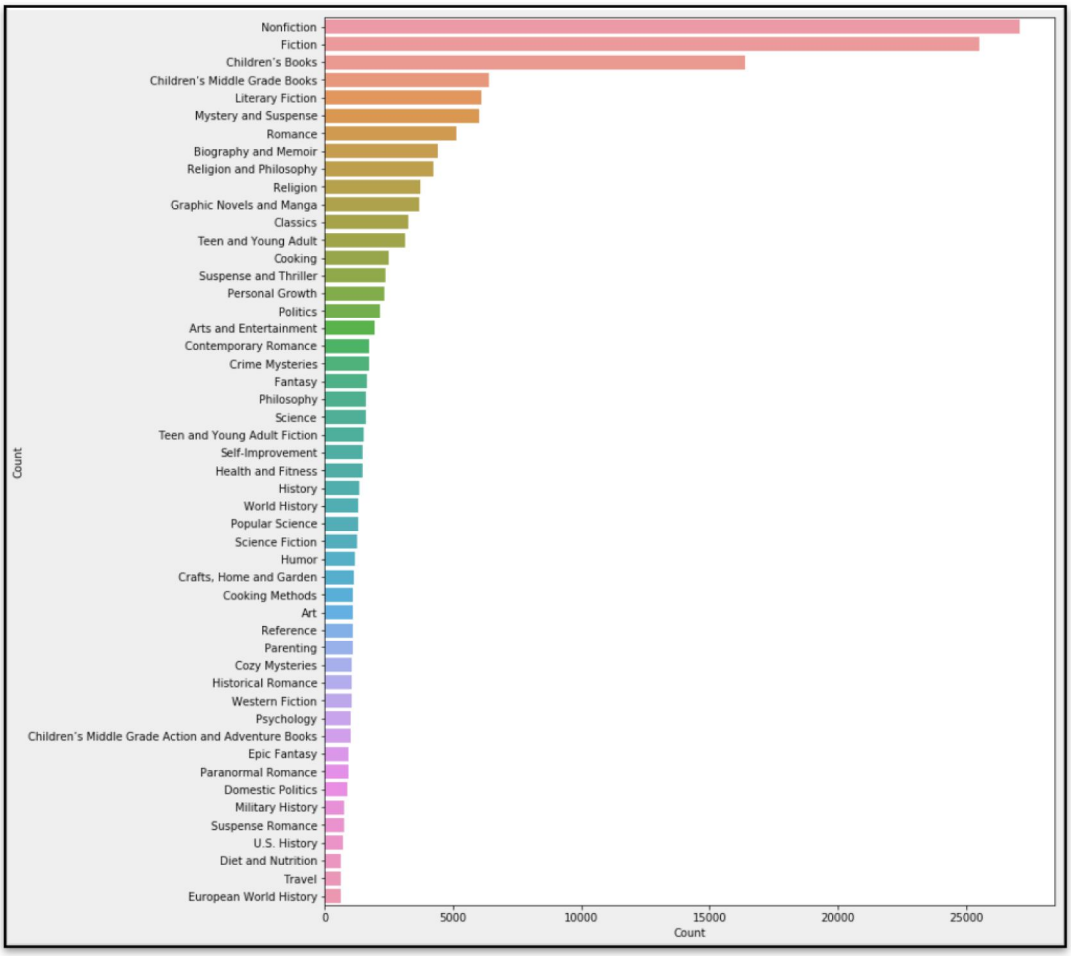


Figure 3: Distribution of the Blurb Genre Collection

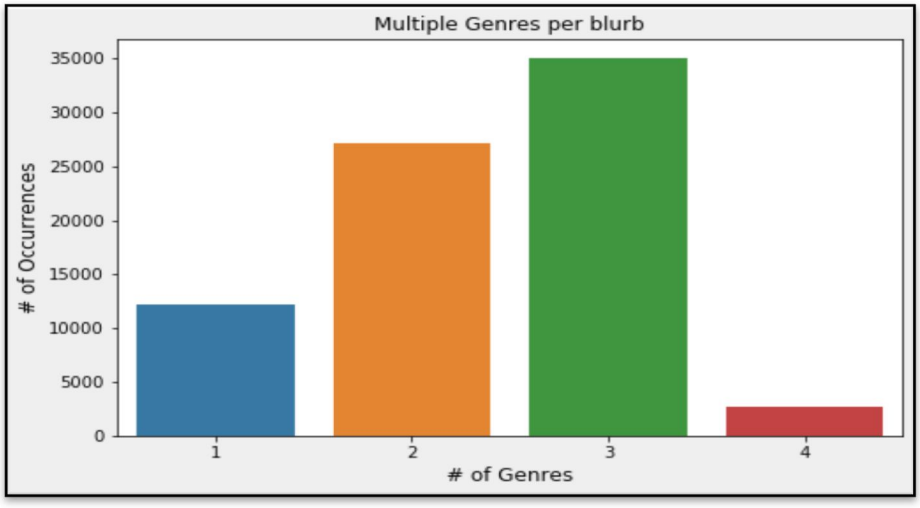


Figure 4: Genres as per summary

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue X Oct 2021- Available at www.ijraset.com

C. Data Preparation

As the data is in XML format, the first step was to extract the information from the file and turn it into a data frame. Every blurb is divided into one or more categories. The categories are arranged in a hierarchical order. Each document in the collection must be assigned at least one category, according to the minimum coding policy. Every ancestor of a document's label gets allocated as well, thanks to the hierarchy policy. We prepared lists of genres to which each blurb belongs to make data access easier.

D. Multilabel Classification

Multi-label classification is a generalization of multi-class classification, which is the single-label problem of categorizing instances into precisely one of more than two classes. In the multi-label problem, there is no limit to how many classes an instance can be assigned to, so the output data used for training could have one, two, or many labels. The F1 metric was used. The harmonic mean of precision and recall is used to determine the score. (F1 Score = 2 * (precision * recall) / (precision + recall)). This F1 score is micro averaged before being used as a multi-class classification metric. The value of true positives, false positives, true negatives, and false negatives is counted to calculate it. In this scenario, all of the projected outputs are column indices, and they are utilized in sorted order by default. To modify the dataset to the multilabel, initially, the modification of the dataset was done to create a binary matrix such that there is a separate column for the genre. There are 139 different genres to choose from. If a blurb belongs to a genre, it will have a value of 1 in that column; otherwise, it will have a value of 0. Another method takes advantage of scikit-multilabel binarized. The intuitive format is converted to the supported multilabel format, which is a (samples x classes) binary matrix indicating the presence of a class label.

The next step was to use the TF-IDF vectorizer from the scikit-learn library to vectorize the summaries. For multi-label prediction, the use of a One vs. Rest classifier was done after vectorizing. This technique, often known as one-vs-all, consists of fitting one classifier per class. The class is fitted against all the other classes for each classifier. This technique has the advantage of being interpretable, in addition to being computationally efficient (just n classes classifiers are required). Because each class is represented by only one classifier, inspecting its matching classifier can provide information about the class. This is the most frequent multiclass classification approach, and it's a good starting point. This approach may also be used for multilabel learning, which involves fitting a classifier to a 2-d matrix in which cell [i, j] is 1 if sample i contains label j and 0 otherwise. It is needed to provide an estimator as a parameter to the OneVsRestClassifier, an estimator is an object implementing fit and one of decision_function or predict-proba. In the context of the present study, two estimators can be used i.e., Linear Support Vector Machine and Logistic Regression. After this point, it was trained the classifier using through the fit function.

IV. RESULT

A. CMU Book Summary

To find the accuracy of the developed system for book genre prediction in the CMU Book Summary dataset, the predict function was applied to predict the multi-class targets using underlying estimators. The output of the results is presented in figure 5. Apparently, the SGD classifier achieves an accuracy of 0.81 for single label genre prediction (Stochastic Gradient Descent). Except for genres with fewer records, we get balanced precision and recall levels for all genres. However, for books, a single genre prediction is typically insufficient; we need numerous genre predictions for the model to be useful.

accuracy 0.8125285779606767

	precision	recall	f1-score	support
Adventure novel	0.93	1.00	0.96	211
Biography	0.83	1.00	0.91	112
Children's literature	0.67	0.50	0.57	330
Fantasy	0.81	0.77	0.79	361
Horror	0.87	1.00	0.93	299
Literary fiction	0.86	0.99	0.92	85
Memoir	0.90	1.00	0.95	145
Mystery	0.74	0.83	0.78	452
Non-fiction	0.87	1.00	0.93	466
Novel	0.46	0.09	0.15	296
Philosophy	0.94	0.99	0.96	74
Picture book	0.89	0.98	0.94	162
Romance novel	0.86	1.00	0.92	213
Science Fiction	0.79	0.72	0.75	283
Short story	0.89	1.00	0.94	107
Spy fiction	0.88	0.99	0.93	312
Sword and sorcery	0.88	1.00	0.93	63
Techno-thriller	0.91	1.00	0.95	131
Thriller	0.80	0.69	0.74	343
Urban fantasy	0.87	1.00	0.93	76
Young adult literature	0.76	0.65	0.70	191
autobiography	0.83	0.99	0.90	443
comedy	0.87	1.00	0.93	473
fiction	0.62	0.43	0.51	512
historical	0.72	0.81	0.76	421
accuracy			0.81	6561
macro avg	0.82	0.86	0.83	6561
weighted avg	0.79	0.81	0.79	6561

Figure 5: Output

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue X Oct 2021- Available at www.ijraset.com

B. The Blurb Genre Collection

For the Blurb Genre Collection, to estimate the probability, predict_proba function was used that gives the output of 0.7136495754642689. Therefore, using this model, the F1 score achieved is 0.713. The estimated values for all classes are sorted by class label. It's worth noting that each sample in the multilabel instance can have any number of labels. The marginal probability that the provided sample contains the label in issue is returned. It is perfectly consistent, for example, that two labels have a 90% chance of applying to the same sample.

Now, to identify or affirm that whether the book belongs to that genre, the threshold value was utilized. Using various thresholds on the same prediction allows us to find the best value, which in our instance is 0.25. When tags are chosen based on a lower threshold value, too many tags are chosen, lowering the F1 metric score, and when the threshold value is very big, nearly no tags are picked, lowering the performance metric score. In the next step, inverse_transform was used to obtain the string values of the classes from the binarizer. Figure 6 depicts the actual label and the predicted labels of the books. In addition, the accuracy of correct prediction is also obtained which is demonstrated in figure 7.

```
Book:  The Art of Betty and Veronica
Predicted  genre:  [('Art', 'Arts and Entertainment', 'Nonfiction')]
Actual genre:  ['Art', 'Arts and Entertainment', 'Humor']

Book:  Strong Enough
Predicted  genre:  [('Contemporary Romance', 'Fiction', 'Romance', 'Teen and Young Adult')]
Actual genre:  ['Suspense Romance', 'Romance', 'Fiction']

Book:  Strange Relations
Predicted  genre:  [('Children's Books', 'Children's Middle Grade Books')]
Actual genre:  ['Teen and Young Adult Fiction', 'Teen and Young Adult']

Book:  Accidentally in Love
Predicted  genre:  [('Contemporary Romance', 'Fiction', 'Romance')]
Actual genre:  ['Contemporary Romance', 'Romance', 'Fiction']

Book:  Tell Them Who I Am
Predicted  genre:  [()]
Actual genre:  ['Nonfiction']
```

Figure 6: actual label and predicted labels

```
... Processing Nonfiction
Test accuracy is 0.9399559071456361
... Processing Fiction
Test accuracy is 0.9297108027493192
... Processing Children's Books
Test accuracy is 0.9569446245623136
... Processing Children's Middle Grade Books
Test accuracy is 0.946958889897549
... Processing Literary Fiction
Test accuracy is 0.9425496044611594
... Processing Mystery and Suspense
Test accuracy is 0.9712099597976916
... Processing Romance
Test accuracy is 0.9744520814420957
... Processing Biography and Memoir
Test accuracy is 0.9561665153676566
... Processing Religion and Philosophy
Test accuracy is 0.9728958630527818
```

Figure 7: Accuracy of prediction

C. Multi-feature Classification

The book's summary is solely used to predict its genre in the overall model. However, other factors such as the book's title and author may have an impact on the genre. As a result, the use of various features to classify the data is explained in this section. The first step was to convert the row into a dictionary. Then, applying the various classifier. It was observed that the best accuracy comes from the Naive Bayes classifier. Naive Bayes classifiers, a subset of classifiers based on the well-known Bayes' probability theorem, are well-known for producing simple yet effective models, particularly in the field of document categorization. The Naive Bayes method was shown to be the most efficient in this investigation as well.

The accuracy of 0.72 (Output: 0.7204800995992686) using the multi-feature model was achieved. Furthermore, the relevant predictions were also obtained while using this model to predict the genres of random summaries outside the dataset. This model was stored in the pickle files for further applications. To provide a good interface, a flask web application was developed. An AJAX request is performed to the server when the user inputs a summary, and the response is shown by parsing the JSON data received. The preview of the webpage is demonstrated in figure 8.

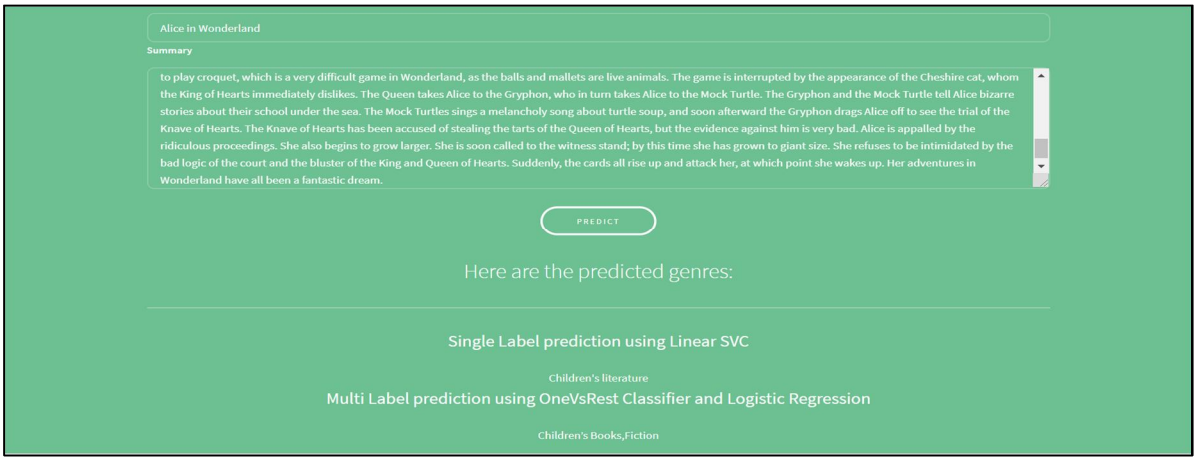


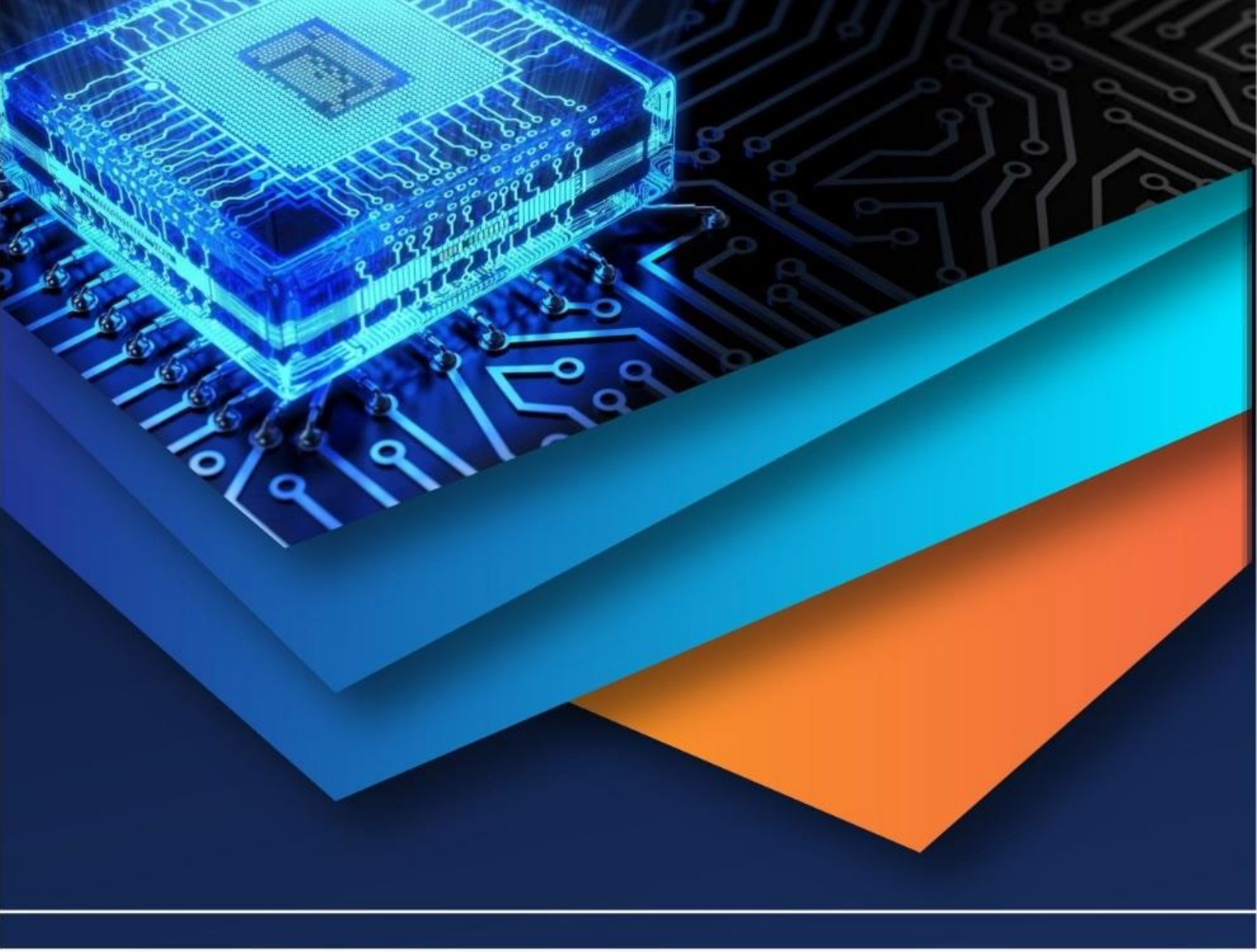
Figure 8: Preview of flask web application

V. CONCLUSION

The author of the paper worked on two datasets namely CMU Book Summary and the Blurb Genre Collection. After performing various predictions of both the dataset sit was observed that the author was able to develop high-accuracy models that are efficient. Not only into the datasets, nevertheless, the model also worked effectively with random inputs outside of the dataset. The various approaches were used to solve the identical problems including single-label classification, multi-label classification, and multi-feature classification were found to be accurate. During the study, it was noted that it is important to maintain the context of the words in the summary as well as to ensure that we are accounting for overlaps between various genres. This was achieved using the multi-label model. Although the classification of the book genre prediction is complex in nature, however, the above models proved to be successful and accurate to predict the book genre.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.
[2] Leo Breiman. 2001. Random forests. Machine learning, 45(1):5–32. Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, pages 785–794, New York, NY, USA. ACM.
[3] Thomas Cover and Peter Hart. 1967. Nearest neighbor pattern classification. IEEE transactions on information theory, 13(1):21–27.
[4] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In Proceedings of the 28th international conference on machine learning (ICML-11), pages 513–520.
[5] Sepp Hochreiter and Jurgen Schmidhuber. 1997. Long short-term memory. ¨ Neural Comput., 9(8):1735–1780, November.
[6] Anna Huang. 2008. Similarity measures for text document clustering. In Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand, pages 49–56.
[7] Brett Kessler, Geoffrey Numberg, and Hinrich Schutze. 1997. Automatic detection of text genre. In ¨ Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics, pages 32–38. Association for Computational Linguistics.
[8] Yoon Kim. 2014. Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.
[9] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
[10] Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In Advances in neural information processing systems, pages 3294–3302.
[11] Aleksander Kolcz, Vidya Prabakarmurthi, and Jugal Kalita. 2001. Summarization as feature selection for text categorization. In Proceedings of the tenth international conference on Information and knowledge management, pages 365–370. ACM.
[12] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In AAAI, volume 333, pages 2267–2273.
[13] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In Proceedings of the 31st International Conference on Machine Learning (ICML-14), pages 1188–1196.
[14] Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. 2015. A hierarchical neural autoencoder for paragraphs and documents. arXiv preprint arXiv:1506.01057.
[15] Ying Liu, Han Tong Loh, and Aixin Sun. 2009. Imbalanced text classification: A term weighting approach. Expert systems with Applications, 36(1):690–701



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY