GENRE CLASSIFICATION

A REPORT

submitted by

GOLLAMUDI SRI HARSHA (18BCE1112)
N S V S ADITYA (18BCE1191)

for the course

NATURAL LANGUAGE PROCESSING
(CSE 4022)

under the professor

Dr. G BHARADWAJA KUMAR

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

DECEMBER 2021

BONAFIDE CERTIFICATE

Certified that this project entitled **“Genre Classification”** is a bonafide work of **Gollamudi Sri Harsha (18BCE1112)** and **N S V S Aditya (18BCE1191)** who carried out the J-component under my supervision and guidance.

Dr. G BHARADWAJA KUMAR

Professor

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING (SCSE)

VIT UNIVERSITY, CHENNAI CAMPUS - 600127

ACKNOWLEDGEMENT

We would like to express our sincere thanks and a deep sense of gratitude to our project guide, **Dr. G Bharadwaja Kumar**, professor, SCSE, for his consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

We are extremely grateful to the Dean of the SCOPE, VIT Chennai, for extending the facilities of the school towards our project and for the unstinting support. We also take this opportunity to thank all the faculty of the school for their support and their wisdom imparted to us throughout the course.

ABSTRACT

For a viewer who is searching for a certain type of story, reading through the plot may be quite taxing. As viewers have their preferences, it would be better to identify the genre based on the summary to quickly decide on. Our Genre Classifier aims to make a quick work of this problem by predicting the genre based on the summaries or plot of the story. The idea is to build a classifier that can predict the genre of the story using a trained model. Multi-label classification is done and more than one genre is generated based on the summary given for a movie. Four different methods have been used for this project, Term Frequency - Inverse Document Frequency, Word to Vector average, Word to vector sum, Bag-of-words.

TABLE OF CONTENTS

3. KEYWORDS

- **TF-IDF -** Term Frequency-Inverse Document Frequency is a method in Information retrieval that works based on numerical statistics that gives out the importance or weight of a word in a group of documents.

- **Word 2 Vector -** It is a technique for Natural Language Processing that uses a neural network model to learn word associations from a large corpus of text. It detects similar words or suggests additional words for a partial sequence.

- **Bag-of-words -** It is a representation used in Natural Language Processing. A text is represented as a bag of its words, it does not concentrate on the grammar and word order. This model is used when the frequency of words is important and is used as a feature for classification.

- **Machine Learning -** Machine Learning is a subset of Artificial Intelligence that works with data and learns through experience and predicts outcomes. It uses historical data as input to predict new output values.

- **Multi-label classification -** It is a type of classification in machine learning where the number of classification tags is more than one.

- **Deep Learning -** It is a type of Machine Learning and Artificial Intelligence that uses the concept of artificial neural networks and covers the gaps that a machine learning algorithm cannot do. It is used for image processing and Natural Language Processing.

4. INTRODUCTION

In the current scenario of COVID, "Over The Top" (OTT) services have become really popular. All kinds of languages and types of shows are available in these services and when one has to pick which one to watch over a huge bag of videos and movies, a genre can be a very helpful way to make one's search easy.

A genre is tagging a movie based on its category. It is a good way of sorting movies and series in a database so as to shortlist the type of movie one wants to watch. Examples of genres are drama, sci-fi, action, adventure, romance, etc.

Using information retrieval models and deep learning models, this project gives out all the genres of a movie based on the summary given in as input. This project uses multi-label classification as its core idea as giving out as many sub-categories of a movie as possible can help sort things better. And help viewers pick shows or movies easily based on the genre since reading every genre is a hectic process. The CMU dataset is used and TF-IDF, word2vec sum and product, bag-of-words are the methods used to classify the genre which will be discussed in the next sections of the report.

5. DATASET

CMU movies summary corpus has been used for the project. This dataset has 42,306 movie plots extracted from Wikipedia and Freebase. It is a large dataset with a lot of attributes consisting of Movie name, release date, box office revenue, runtime, languages, countries, genre, etc.

This is the dataset link:

http://www.cs.cmu.edu/~ark/personas/

6. METHODOLOGY

6.1 Pre-processing Techniques

There were three pre-processing techniques applied to the dataset:

- ➢ **Filtering Text:** In the plots of the movies, all the characters except English alphabets are filtered from the synopsis.
- ➢ **Stop-Words Removal:** The stop-words are removed using the ‘NLTK stop-words’ package.
- ➢ **Lemmatization:** The available words in the ‘NLTK wordnet’ package have been lemmatized.

6.2 Word Embedding Methods

Four Word Embedding methodologies have been implemented in the project to develop four models.

- **TF-IDF:** It is a word embedding method used to determine the significance of the words with respect to a document. Here the value of each word is the TF-IDF value of the word which is a product of the Term Frequency and Inverse Document Frequency of a word. Here, Term Frequency calculates how frequent a word is in a document while Inverse Document Frequency is the logarithm of the ratio of total documents in the dataset to the number of documents in which the given word is present.
- **Bag of Words:** This is one of the most direct word embedding methods used to represent textual data. Firstly, a vector the size of a total number of unique words is initialized with all zeros where each entry corresponds to a particular unique word. If the word is present ‘n’ a number of times in a document then the number corresponding to that word in the list will be changed to ‘n’. Because of this, the models developed using this method tend to have a very high number of input features which makes the process of training the model very time-consuming. This method also ignores the semantic relationship between the documents.
- **Word2Vector (Sum of Vectors):** This is one most popular and widely used

word embedding methodologies. The word vectors are created by using the entire dataset and being subjected to an unsupervised learning process with the use of deep learning models. This method produces a word vector that is not as large as other methods as the size of this vector is generally very less compared to the number of unique words present in the document. This means that the features of the textual data can be expressed in a much smaller set of features with more details. The Gensim word2vector model was used during the course of the project. The word vectors of all the available words in the document are added together to produce a final vector with a size of 300. This vector represents the given document in the dataset.

➢ **Word2Vector (Average of Vectors):** Much like the above-mentioned method, this model also converts all the available words into word vectors but instead of adding all of them the mean of all the word vectors is taken as the word embedding.

6.3 Multi-label Classification Model

The Machine Learning library used for this project is ‘SKLearn’. The model chosen for this scenario is Logistic Regression as the movie genre needs to be either one or zero. So the sigmoid activation function is very well-suited for the genre classification. Since this is a multi-label classification model, the ‘onevsrest’ classifier is used in conjunction with logistic regression. This model pits one genre against all others and sends the confidence to the sigmoid function which either predicts its ‘0’ or ‘1’ for that particular genre. Once this is done to all the available classes that are the genres, the model gives the output list with the genres predicted.

6.4 Evaluation Metrics

The metrics used in cross-validation are:

Accuracy: The measure of closeness between predicted and actual data.

Precision: It is the fraction of the number of positive predictions that truly belong to the positive class.

Recall: It is the fraction of the number of positive predictions made out of all positive data points.

Log Loss: Log-loss is a measure of how close the prediction probability is to the actual values.

F1 Score: It is the harmonic mean of precision and recall scores.

7. CODE

The python Jupyter Notebooks are zipped in the code folder. There are three files where one is about the TF-IDF model and this also consists of the pre-processing module. The Bag of Words model is another file and the last file deals with the two

Word2Vector models. This last file also encloses the testing module.

8. RESULTS

The cross-validation and the results and values for Accuracy, F1 score, Precision, Recall, and Log loss have been recorded for the 4 different models used for the project.

1) TF-IDF

Type of data	Metric	Value
Training data	Accuracy	0.06113909017915033
	F1 score	0.22874838015550467
	Precision	0.9602876083580141
	Recall	0.1298431260053659
	Log loss	59.402530370316
Testing data	Accuracy	0.0358923301721915
	F1 score	0.12411518090924578
	Precision	0.6945109253787436
	Recall	0.06818587652349684
	Log loss	47.595600597433524

2) Bag-of-words

Type of data	Metric	Value
Training data	Accuracy	0.9922828104716418
	F1 score	0.9983854735575697
	Precision	1.0
	Recall	0.9967762823723441
	Log loss	5.722458094408212
Testing data	Accuracy	0.04307022318998367
	F1 score	0.31274569039507694
	Precision	0.5065809513388303
	Recall	0.22651536349678564
	Log loss	76.97765288540411

3) Word2Vec Sum

Type of data	Metric	Value
Training data	Accuracy	0.21332810686330217
	F1 score	0.7391224000153741
	Precision	0.8392146893102602
	Recall	0.6603847556091423
	Log loss	41.737645335145146
Testing data	Accuracy	0.020818267770678738
	F1 score	0.3036897146331038
	Precision	0.32230184007816753
	Recall	0.28759004120586046

	Log loss	85.85982698536152

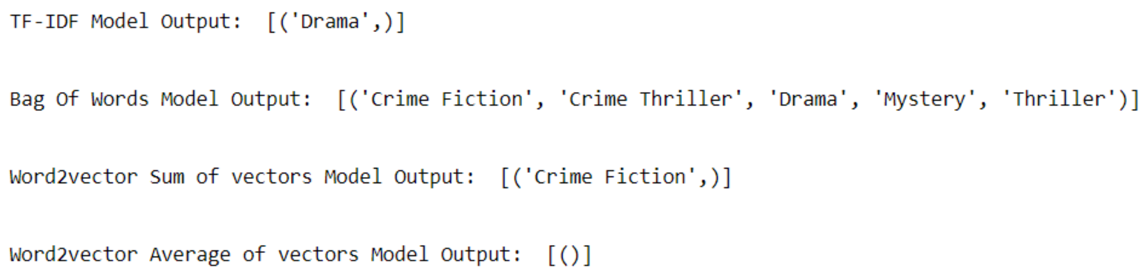
4) Word2Vec average

Type of data	Metric	Value
Training data	Accuracy	0.05557561898246283
	F1 score	0.2027928273603697
	Precision	0.6868528752008962
	Recall	0.11896419122649862
	Log loss	59.706683741814594
Testing data	Accuracy	0.05049021574076727
	F1 score	0.18829225111799863
	Precision	0.6370812145565228
	Recall	0.11060929235501223
	Log loss	59.93462712257597

Sample Test Cases

1) Movie Name: Jai Bhim

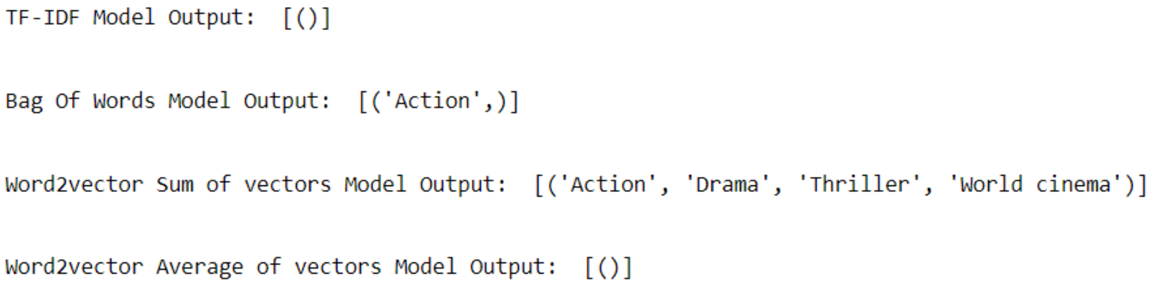
Prediction:



```
TF-IDF Model Output:  [('Drama',)]

Bag Of Words Model Output:  [('Crime Fiction', 'Crime Thriller', 'Drama', 'Mystery', 'Thriller')]

Word2vector Sum of vectors Model Output:  [('Crime Fiction',)]

Word2vector Average of vectors Model Output:  [()]
```

2) Movie Name: Dookudu

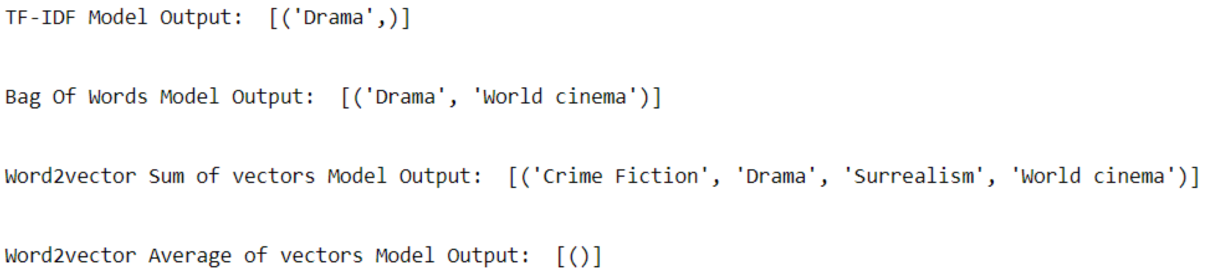
Prediction:



```
TF-IDF Model Output:  [()]

Bag Of Words Model Output:  [('Action',)]

Word2vector Sum of vectors Model Output:  [('Action', 'Drama', 'Thriller', 'World cinema')]

Word2vector Average of vectors Model Output:  [()]
```

3) Movie Name: Drishyam

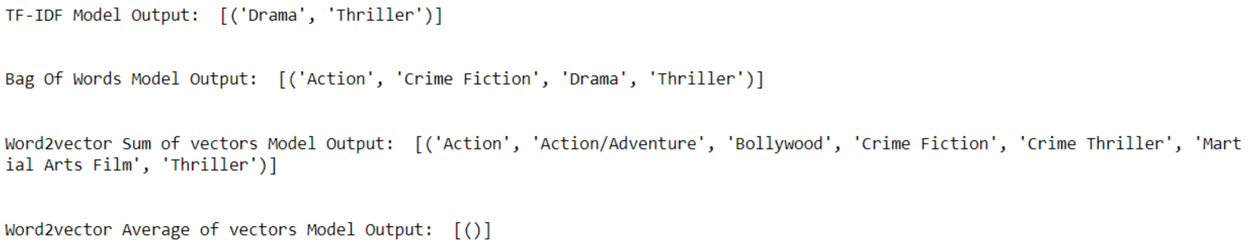
Prediction:



```
TF-IDF Model Output:  [('Drama',)]

Bag Of Words Model Output:  [('Drama', 'World cinema')]

Word2vector Sum of vectors Model Output:  [('Crime Fiction', 'Drama', 'Surrealism', 'World cinema')]

Word2vector Average of vectors Model Output:  [()]
```

4) Movie Name: Sahoo

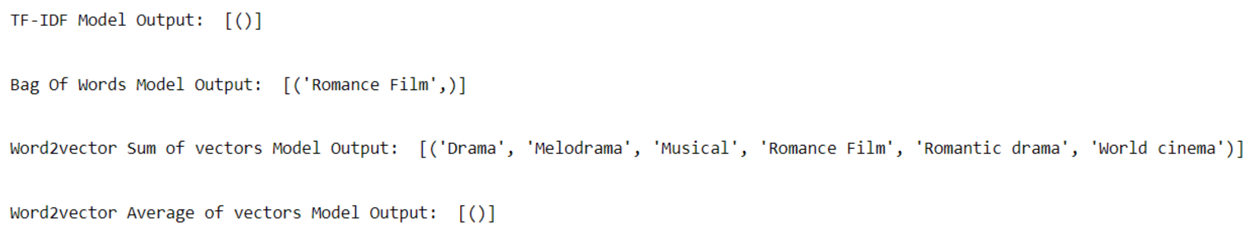
Prediction:



```
TF-IDF Model Output:  [('Drama', 'Thriller')]

Bag Of Words Model Output:  [('Action', 'Crime Fiction', 'Drama', 'Thriller')]

Word2vector Sum of vectors Model Output:  [('Action', 'Action/Adventure', 'Bollywood', 'Crime Fiction', 'Crime Thriller', 'Mart
ial Arts Film', 'Thriller')]

Word2vector Average of vectors Model Output:  [()]
```

5) Movie Name: Bommarillu

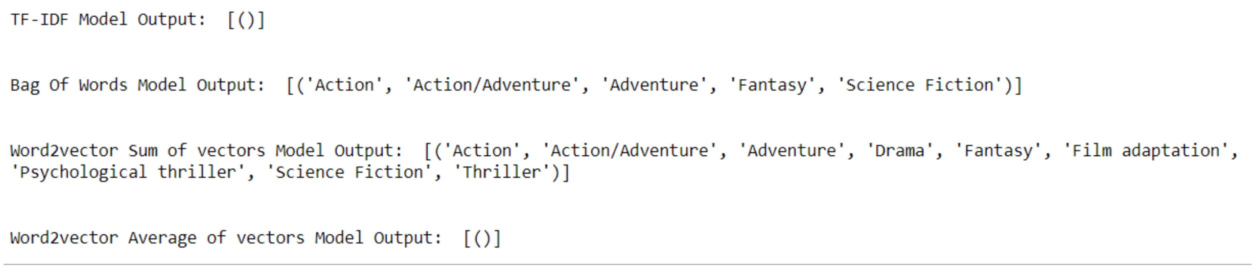
Prediction:



```
TF-IDF Model Output:  [()]

Bag Of Words Model Output:  [('Romance Film',)]

Word2vector Sum of vectors Model Output:  [('Drama', 'Melodrama', 'Musical', 'Romance Film', 'Romantic drama', 'World cinema')]

Word2vector Average of vectors Model Output:  [()]
```

6) Movie Name: Avengers: End Game

Prediction:



```
TF-IDF Model Output:  [()]

Bag Of Words Model Output:  [('Action', 'Action/Adventure', 'Adventure', 'Fantasy', 'Science Fiction')]

Word2vector Sum of vectors Model Output:  [('Action', 'Action/Adventure', 'Adventure', 'Drama', 'Fantasy', 'Film adaptation',
'Psychological thriller', 'Science Fiction', 'Thriller')]

Word2vector Average of vectors Model Output:  [()]
```

9. CONCLUSION

The whole new way of entertainment and the market leaders are the OTTs such as Netflix, Amazon prime video, Disney plus Hotstar, etc. All the leading OTTs organize the shows and movies based on their genre which makes the interface smooth and user-friendly. Using the four models, this project generates multiple genres for a movie.