International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 06 | Jun 2022 www.irjet.net p-ISSN: 2395-0072

Tags Prediction from Movie Plot Synopsis Using Machine Learning

Nagareddy Deepthi[1], Tulasi N[2], Mrs Mona[3]

[1]Student Dept. of Information Science and Engineering, BNM Institute of Technology, Karnataka, India
[2] Student Dept. of Information Science and Engineering, BNM Institute of Technology, Karnataka, India
[3]Assistant Professor, Dept. of Information Science and Engineering, BNM Institute of Technology, Karnataka.

---***---

Abstract - Movies can be tagged with various details such as genre, plot structure, soundtracks, and visual and emotional experiences etc. This information can be used to build automatic systems to extract similar movies, enhance user experience and improve the recommendations. We describe a way that allowed us to create a fine-tuned set of various tags that expose the varied characteristics of movie plots. T h e correlation between tags is varying which we make a study on different types of movies. By means of this project we will be solving a Multi-LabelClassification problem. This system was built to study the inferring tags from plot synopses. We hope it will be usefulin other tasks related to narrative analysis.

Key Words: Plot Synopsis, Machine Learning, Logistic Regression, One Vs Rest, SGD Classifier etc

1. INTRODUCTION

Various features of movies, such as genre, plot structure, soundtracks, and emotional responses, visuals and so on can be labelled Because of the massive volume of multimedia data generated these days, it' s difficult to imagine systems that can automatically analyze the content to determine the validity and classify them. Manual processing is important for extracting needed information from the data and assigning suitable tags. As a result, tag quality is based on a subjective criterion that differs from person to person. This generated metadata makes it difficult to gain complete insights intomajor elements of a movie and due to lack of precision, there are irregularities and hence data is less accurate which could impact user experience. This data may be utilizedto create automated algorithms that extract related movies, dealing with the growing problem of information overloadand improve the user experience and suggestions as well as provide users with a preview of what to expect from a movie. This tag prediction model has various applications like object detection, automatic subtitles generation and optimization of movie search engines and mainly in content censorship.

Only a machine learning based algorithm can efficiently complete such tasks. Here, we will address this problem by creating an automated engine that can extract tags from the plot of the movie which is a detailed description of a movie (synopsis of movie story line) or summary of a movie. A movie can have one or more tags this is when multi-label categorization comes into play where each sample is givena set of target labels. e. g., classifying a dataset which may be adventure or action, comedy, horror, flashback. Machine learning's remarkable breakthroughs have paved the road for discovering patterns in data with high accuracy, the completion of tasks with a machine learning based algorithm is that's whyefficient. We describe a way that will allow us to create a fine-tuned set of various tags that expose the varied characteristics of movie plots. After this we check as to how these tags relate to different sorts of films. We will use this model to assess if we could somehow infer tags from plot synopsis.

Fig-1: Movies

Fig-2: Tagcloud

2.Related Work

We feel that there has been a low attention towards the tag prediction and categorization of movies in the literature. The relevant work in this sector is mainly focused on small-scale image, video, blog and other content-based tagging. S. Kar, et. al. [1], proposed a model which used plot analysis for tag prediction. They proposed a novel neural network model that merges information from synopsis and emotion flows throughout the plots to predict a set of tags for movies. In order to create the tag set, they tackled the challenge of extracting tags related to movie plots from noisy and redundant tagspaces created by user communities in Movie Lens and IMDb.

Tag set and mapping the tags to the plot synopses and also provided an analysis, to find out the correlations between tags.These correlations seem to portray a reasonable set of movie types based on what we expect from certain types of movies in the real world. They also tried to analyze the structure of some plots by tracking the flow of emotions throughout the synopses, where it is observed that movies with similar tag groups seem to have similarities in the flow of emotions throughout the plots. They created a system to predict tags from the synopses using a set of hand-crafted linguistic features. For each movie, the algorithm forecast a limitednumber of tags.

Ali Mert Ertugrul, et. al. [2], proposed 'Movie Genre Classification from plot summaries' system where they performed movie genre classification using Bi-directional LSTM. They used sentence level approach. Since plot summaries contains many sentences, they first divide each plot summary into sentences. Then using word representationsof sentences, they trained Bi-LSTM networks to estimate genres of each sentence separately. Since there are many sentences and many genres, they used majority voting forthe final decision by considering the probabilities of genres assigned to sentences. The results reflect that, training Bi- LSTM network after dividing the plot summaries into their sentences and fusing the predictions for individual sentences outperform training the network with the whole plot summaries with the limited amount of data.

Antonius Christiyanto Saputra, et. al. [3], proposed the 'Classification of the Movie Genre based on Synopsis of the Indonesian Film' system that aims to classify a document content into the correct label. Synopsis text data in Indonesian films is used as a features to determine the appropriate genre of film by utilizing machine learning algorithms. They used document level approach. The text classification system aims to classify a document content into the correct label. They have used several feature extraction methods and machine learning models to classify the movie genre. Their model gave the best results with the Support Vector Machines classification algorithm with TF-IDF extraction. There are still shortcomings and errors in the results of the classificationof the film genre when conducting summary synopsis input due to the lack of the datasets.

Jingcheng Wang.et.al[4],Proposed the Identification ofMovie genres through online movie synopses' system that aims to identify the genres of movie through movie synopses. The movies and corresponding synopses in database are downloaded from the Kaggle and rotten tomatoes websites. Inthis system accessing useful information from online movie synopsis can be time consuming, misunderstanding and misjudge the genre of the movie. In this system two supervised learning models KNN and SVM and two deep learning models CNN and RNN are used to classify the genres of movie through movie synopses. SVM or KNN can be used when the number of samples in training set is limited, but SVM is always better. Sample data of the training set increases, the training speed of SVM or KNN increases rapidly, and the improvement of accuracy is less, so it is not applicable to the actual situation. CNN and RNN with LSTM layer in deep learning are used which are more suitable toact as a text analysts for huge data. In this system RNN with LSTM layer is the best model overall.

Darren Kuo,et,al.[5],proposed the 'On word prediction methods' This system represents a simple task of word prediction given a piece of text. This system includes clustering of documents, querying for related documents, similarity of documents, and capturing the topics given a set of documents. This model used a co-occurrence approach to generate tags based on the words in the post and their relationship to tags. The model was created for next-word prediction in big datasets and then modified by limiting the predicted next word to just tags. This co-occurrence algorithm correctly predicts one tag per post with a classification accuracy of around 40 percent. The system emphasize on predicting words, which are mixtures of topics.

Gilad Mishne,et,al.[6],proposed the 'Autotag: A collaborative approach to automated tag assignment for weblog posts' The system describes autotag,

A tool which suggests tags for weblog posts using collaborative filtering methods. autotag is used for simplifying the tagging process and in improving quality. autotag identifies tags for a post by examining tags assigned to similar posts. autotag was used to tag 6000 of the tagged posts. In addition to the collaborative approach the investigation of local approach to tag suggestion required.

Teh Chao Ying, Shyamala Doraisamy, Lili Nurliyana Abdullah,et,al.[7], proposed the 'Classification of Lyrics –based genre'. The system aim is to improve the performance of lyrics-based musical genre classification. Music documents are often classified based on genre and mood. The features from lyrics text are used for classification of musical documents and the feasibility of lyrics features to classify musical documents. From the analysis of the lyrics text in the data collection, correlationof terms between genre and mood was observed. In this system term frequency and inverse document frequency values are used to measure relevance of words to the different mood categories. The tf-idf weighting scheme for lyrics text was used to describe the relative importance of a term for a particular musical mood class. In this system dataset size is limited which was 145 songs with lyrics and in fact genre and mood provide complementary descriptionsof music.

3. PROPOSED METHODOLOGY

There are various datasets available that contain movies and their plots, but our main concern is to retrieve a dataset with certain expected attributes, such as tags should be closely related to the plot and not some metadata that is completely irrelevant to the plot, and redundancy in tags should be avoided because we need to assign unique tags, so having tags that represent the same meaning would be ineffective. So, we have gathered data from various internet sources that contains nearly 14,000 movies with unique tag set of 72 tags. here, each data point has six attributes including IMDB id to get the tag association information for respective moviesin the dataset, Title of the movie, Plot of the movie, Tagsassociated with each movie, Split attribute indicating whether the data belongs to test, or train set and source which is either IMDB or Wikipedia.

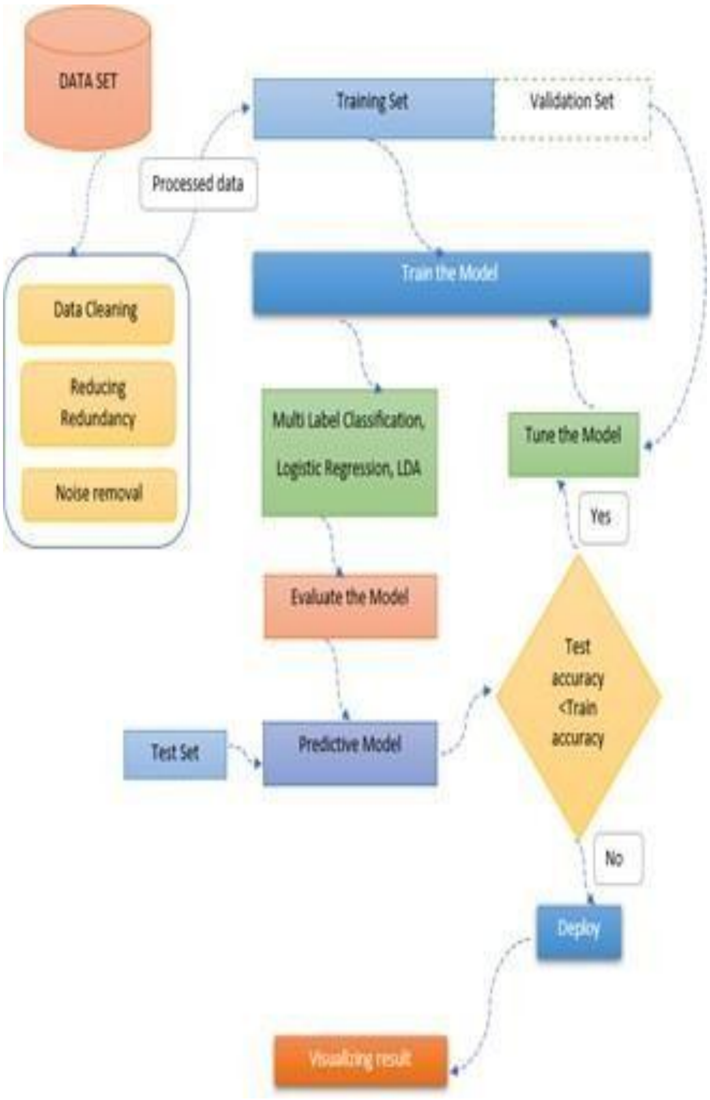
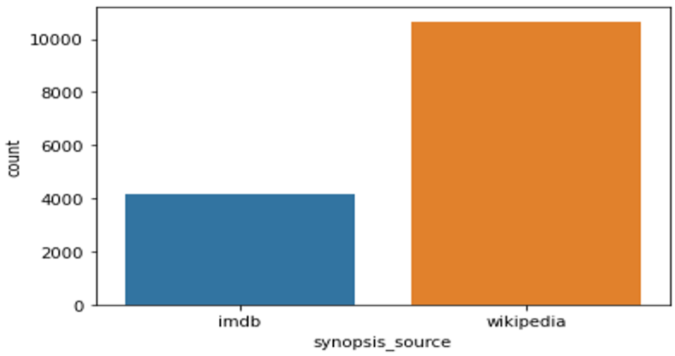


Fig-3: Flow diagram 1

Data Distribution:

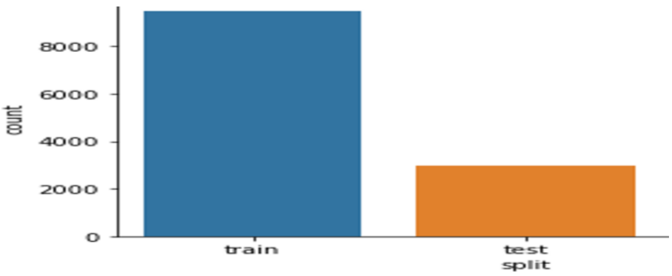
SOURCE:

Most of the data is taken from Wekipedia and some fromIMDB



SPLIT:

Train and test split



TAG ANALYSIS:

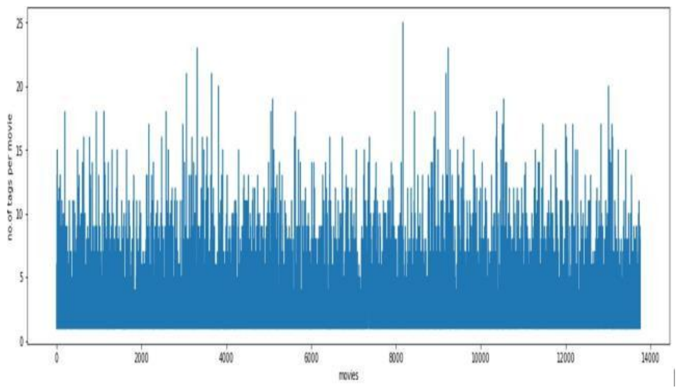


Fig-4: Movie vs number .of tags per movie

Moreover, Plot synopses should not contain any noise suchas HTML tags or IMDB alerts and include sufficient information because understanding stories from extremely short texts would be challenging for any machine learning system, each overview should include at least 10 sentences. Although the text is unstructured data, it is often created by individuals for the purpose of being understood by others. So, how can we handle a big volume of text and convert it into a representation that can be used to predict and classify using machine learning models? There are a variety of methods for cleaning and preparing textual data, and we used a few of them here.

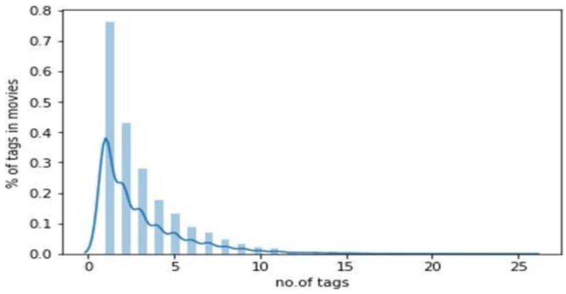


Fig-5:percentage of tags in moves vs no.of tags

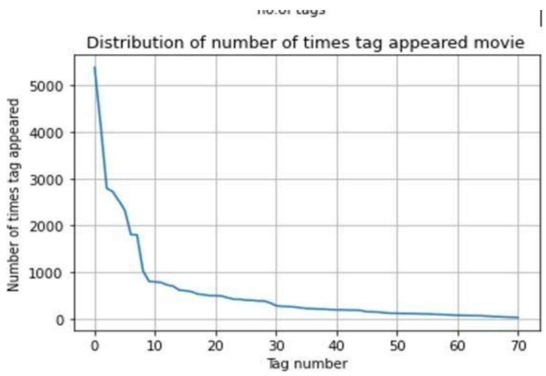


Fig-6: Tag Distribution

- Converting every word to lowercase and removing HTMLtags or other irrelevant elements present in the dataset.
- De-Contraction of words like can't to can not and remov-ing any stop words if present such as "the", "a", "an", "in" as we would not want these terms to take important processing time or space in our database.
- Lemmatization of words, which typically refers to per- forming things correctly with the help of a vocabulary and returning a word to its root form, for instance, convertingwords that are in 3rd person to 1st person and future and past tense verbs to present tense.
- Stemming refers to reducing words to their word stem that affixes to prefixes and suffixes like "-ing", "-es", "- pre", etc.

Exploratory analysis for data of tags-data distribution and feature engineering.

we created a SQL database file of the given source CSVfile and delete the duplicate entries and modify the same by adding a new custom attribute (6+1) tag-count which indicatesthe number of tags associated per movie so, we can find the exact count, how many movies are associated with how many tags. we also checked the number of unique tags present inthe dataset using the BOW (bag of words) technique whichis implemented using the count vectorizer method. We must transform it into vectors of numbers since machine learning algorithms do not accept the raw text as input data. Bag of words is the easiest way to represent Text Documents. In other words, it will determine how many times a specific word appears in a given document. This method yielded a tag cloud such as flashback, violence, murder, romantic and cult. This way we produced a more generic version of the common tags relevant to the plot of the movie. On the other hand, tags such as entertaining, and suspenseful are slightly less common got filtered out.

Machine Learning Approach for Predicting Tags using Plot Synopses

In this part, we'll go through some basic tests we've done using the corpus to predict movie tags. Using all tags:

TFIDF Vectorizer Here, less frequent words are assigned comparatively more weight. TFIDF is the product of Term Frequency (the ratio of the number of times a word appearsin a document to the total number of terms in the document)and Inverse Document Frequency (the log of the number of times a term appears in a document) (the total number of documents to documents with term present in it). We tested several approaches in the baseline model construction portion, including the multinomial Naive Bayes classifier, which is good for discrete feature classification, such as word counts for text categorization in a document. Integer feature counts aretypically required for multinomial distributions, making them robust and simple to build, While Logistic Regression is used when the dependent variable (target) is categorical, and the question arises Why Linear Regression is not used for classification? Two things explain this. The first one is that classification problems mandate discrete values whereas linear regression only deals with continuous values which makes it not suitable for classification. The second thing is the threshold value for this considering a situation where we need to determine whether an email is spam or not. If we use linear regression here, we'll need to select a threshold by which we may classify the data. If the actual class is malignant with a predicted value of 0.45, and the threshold is 0.5, then it willbe classified as non-malignant, which can result in serious consequences in real-time. So, it can be seen that linear regression is unbounded that's why we need logistic regression for instance, in a binary classification where we need to predict if a data point belongs to a particular class or notand the class with the highest probability is where the data point belongs. So, to fit a mathematical equation of such type we cannot use a straight line as all the values of output will be either 0 or 1. That's when sigmoid function comes into picturethat transforms any real number input, to a number between 0 and 1.

Sigmoid equation:

S(x) = 1/1+e^-x

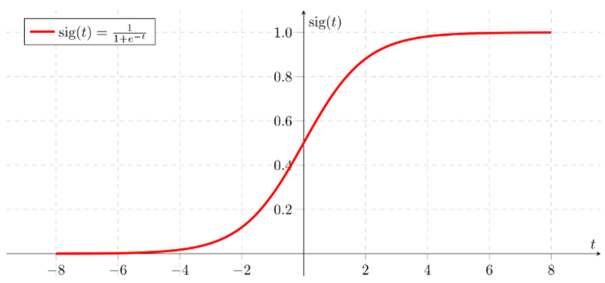


Fig-7: Sigmoid function

$Hypothesis => Z = WX + Bh(x) = sigmoid(Z)$

(2)

Here if Z becomes close to infinity, Y will become 1 andifZ is close to negative infinity, Y will be predicted as 0.

Sigmoid equation for multiple features:

Here, we are computing output probabilities for all K-1classes. For Kth class = 1 — Sum of all probabilities of k-1 classes.

So, we can say that multinomial logistic regression usesK-1 Logistic regression models to classify data points for K distinct classes.

Another method that we tried is SGD classifiers whichis an optimization method, while Logistic Regression is a machine learning model that defines a loss function, and the optimization method minimizes/maximizes it.

In all the cases our goal is to maximize the micro averagedF1 score. We used micro-averaging as here, a movie might have more than two tags/labels associated with it. Micro averaged F1 score is the harmonic mean of micro-Recall and micro-Precision. Micro-precision is the sum of all true positives to the sum of all true positives and false positives.

$$Microprecision = \frac{TP\,1 + TP\,2 + TP\,3 + \ldots}{(TP\,1 + TP\,2 + TP\,3 + \ldots) + (FP\,1 + FP\,2 + FP\,3 + \ldots)}$$

Micro-recall is calculated by first finding the sum of all true positives and false positives, over all the classes. Then we compute the recall for the sums.

$$micro - recall = \frac{TP\,1 + TP\,2 + TP\,3 + \ldots}{(TP\,1 + TP\,2 + TP\,3 + \ldots) + (FN\,1 + FN\,2 + FN\,3 + \ldots)}$$

Logistic Regression with Outliers was the model that provided us the highest micro averaged F1 score. One Vs Rest, sometimes known as one-vs-all, is a method that involvesfitting a single classifier to each class. The class is fitted against all the other classes for each classifier. Despite the fact that this technique cannot handle multiple datasets, it trains fewer classifiers, making it a faster and more popular option.

AVGW2V

By capturing semantic information, word embeddings have been proved to be successful in text classification problems. Asa result, we average the word vectors of each word in the plot to capture the semantic representation of the plots. Thus, we get a 1D vector of features corresponding to each document. By using this model also logistic regression gave the highest micro averaged F1 score.

According to the exploratory data analysis, a movie is typically associated with three tags. As a result, we attempted to create a model that could predict the top three tags. We utilised the same set of features this time, but the number of tags was set to three. By using TFIDF Vectorizer the highest F1 score was achieved by using SGD Classifier with log loss as well as accuracy score is also improving and by usingAVGW2V (average word2vec) Logistic Regression model gave the highest F1 score. Similarly, for the next training of our model, we have set the tags to the top 5 and observed the

F1 score for each model. We read the dataset and vectorize the tags using the BoW algorithm in the following training step to see which word appears how many times. Then, to construct a new data frame, we sorted these tags in decreasing order depending on how many times they appeared in the document. Out of the 71 distinct tags, we manually chose the top 30 tags based on their frequency. Then, except for the 30 tags present per movie, we eliminated all other tags alongwith their respective rows and repeated the procedure to train the model further. For the next stage, we employed Python's Topic Modelling and Latent Dirichlet Allocation (LDA).

We used Latent Dirichlet Allocation (LDA) to classify text in a document to a specific topic. Topic Modelling is a statistical model for discovering the abstract "topics" present in a collection ofdocuments, and it is a commonly utilized for the discovery of hidden semantic meaningful structures in the body of text. It generates a topic per document and word per topic model based on the Dirichlet distribution. (LDA) is a common topic modelling approach with great Python implementations in the Gensim package. The LDA model above is made up of ten separate topics, each of which is made up of several keywordsand given a specific amount of weight to the subject thatrepresent the importance of a keyword to the particular topic. It determines the dominant subject of a particular text and is one of the practical applications of topic modelling.

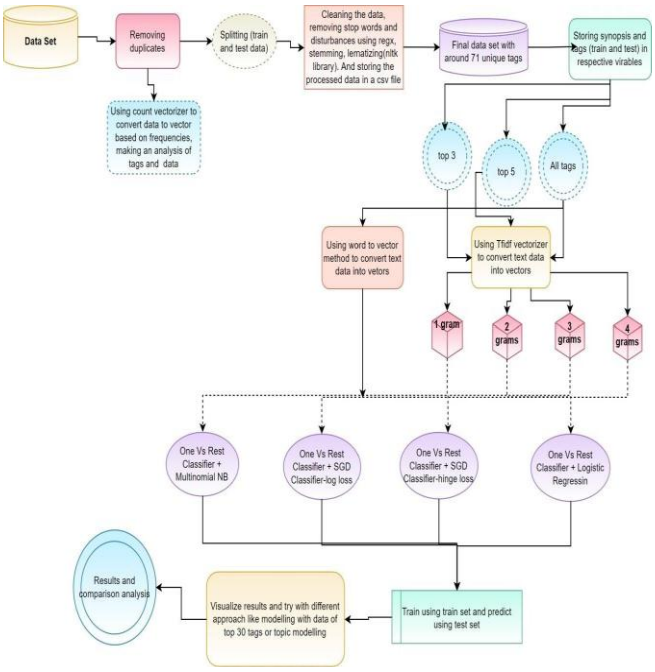


Fig-8: DataFlow diagram 2

Model	Vectorizer	precision	recall	F1-score	n-grams
Multinomial NB	Tfidf Vectorizer	0.467	0.686	0.556	1
		0.468	0.686	0.556	2
		0.468	0.686	0.556	3
		0.468	0.686	0.556	4
SGD Classifier Log-loss	Tfidf Vectorizer	0.502	0.68	0.578	1
		0.514	0.674	0.583	2
		0.521	0.662	0.583	3
		0.502	0.674	0.575	4
SGD Classifier Hinge-loss	Tfidf Vectorizer	0.492	0.7	0.578	1
		0.501	0.677	0.576	2
		0.499	0.684	0.577	3
		0.492	0.68	0.571	4
Logistic Regression	Tfidf Vectorizer	0.508	0.684	0.583	1
		0.508	0.685	0.583	2
		0.508	0.684	0.583	3
		0.508	0.685	0.583	4

Fig-9: Top3-Tag-Prediction

To accomplish so, we search for the topic with the highest percentage contribution. Then we saved these dominating topics to a CSV file and concatenated it with our original data and used the same method to train the model even further to improve accuracy.

4. RESULT AND ANALYSIS

To solve this Multi Label classification problem we used One VS Rest Classifier combining with different types of binary classification algorithms. First time we took data containing all tags (complete pre-processed data set). To make the model understand the plot synopsis we trained the model with different word counts like 1 grams, 2 grams, 3 grams etc. Here firstly Tfidf Vectorizer is used to convert the synopsis data into numeric data. After that we tried to train the model with One VS Rest classifier Multinomial NB, SGD Classifier-log loss, SGD Classifier-hinge loss and logistic regression with 1 grams (initially) and the same process is repeated with 2, 3 ,4 grams. Our aim here is to maximize the f1 score.

Model	Vectorizer	precision	recall	F1-score	n-grams
Multinomial NB	Tfidf Vectorizer	0.116	0.646	0.197	1
		0.117	0.645	0.198	2
		0.117	0.645	0.198	3
		0.117	0.645	0.198	4
SGD Classifier Log-loss	Tfidf Vectorizer	0.157	0.597	0.249	1
		0.159	0.588	0.25	2
		0.16	0.589	0.252	3
		0.157	0.594	0.248	4
SGD Classifier Hinge-loss	Tfidf Vectorizer	0.153	0.584	0.243	1
		0.154	0.583	0.244	2
		0.152	0.586	0.241	3
		0.148	0.58	0.26	4
Logistic Regression	Tfidf Vectorizer	0.167	0.584	0.259	1
		0.167	0.584	0.26	2
		0.167	0.584	0.26	3
		0.167	0.584	0.26	4

Fig-10: All-Tag-Prediction

In case of Multinomial NB the score decreased from 1 to 2 grams and stayed same through 2, 3, 4 grams. And when it comes to SGD Classifier-log loss it decreased from 1 to 2 to 3 and increased in 4 grams (and this is the max). SGD Classifier-hinge loss increased gradually and highest is at 4 grams. And now comes the logistic regression it's score increased from 1 to 2 grams and remained unchanged later and logistic regression showed highest scores in individual models of 1, 2, 3, 4 grams and also in the overall case (and it is 0.26). And on a overview 4 grams model gave the better results when compared with the rest. Figure "fig 9".

Now to improve the model even further we try to implement this procedure with top3 and 5 tags as we saw in analysisof data, we take maximum features as 3 and

5 in these respective trainings. And follow the above that we have done with data with all tags.

Model	Vectorizer	precision	recall	F1-score	n-grams
Multinomial NB	Tfidf Vectorizer	0.411	0.676	0.511	1
		0.412	0.676	0.512	2
		0.412	0.676	0.512	3
		0.412	0.676	0.512	4
SGD Classifier Log-loss	Tfidf Vectorizer	0.448	0.676	0.539	1
		0.455	0.66	0.539	2
		0.444	0.665	0.532	3
		0.445	0.675	0.536	4
SGD Classifier Hinge-loss	Tfidf Vectorizer	0.435	0.674	0.529	1
		0.441	0.672	0.533	2
		0.434	0.668	0.526	3
		0.443	0.649	0.526	4
Logistic Regression	Tfidf Vectorizer	0.447	0.668	0.535	1
		0.447	0.668	0.535	2
		0.447	0.668	0.535	3
		0.447	0.668	0.535	4

Fig-11: Top5-Tag-Prediction

Multinomial NB stays the same in all variants. The f1 of SGD Classifier-log loss increases from 1 to 2 grams and decreases from 3 to 4 grams (but still ¿ 1gram). SGD Classifier-hinge loss doesn't show much variation and logistic regression also stays the same in all cases. But this time SGD Classifier-log loss shows highest score (of all variants) in case of 2 grams, that is 0.568. Here 2 grams performance is better compared to the rest."fig 10"

Let's go a step further and try to predict top 5 tags "Fig

11. The f1 score of Multinomial NB it increases from 1to 2 grams and remains unaltered. SDG Classifier-log loss f1 decreases in 3 and 4 grams where as equal in 1 and 2 grams. SDG Classifier-hinge loss increases the score gradually in all grams. Logistic Regression shows same scores with all 1 2 34 grams. But here also logistic regression is the highest f1 score achiever and the score is 0.535."fig 11" Now let's compare among these models which try to predict all, top 5 and 3 tags. We can clearly see the significant increase in scores of f1 in top three and five tag prediction compared to all tags. And top 3 model is showing the best results among these.

Model	Vectorizer	precision	recall	F1-score
SGD Classifier Log-loss	AVG-W2G	0.12	0.632	0.201
SGD Classifier Hinge-loss	AVG-W2G	0.103	0.615	0.176
Logistic Regression	AVG-W2G	0.129	0.63	0.214

Fig-12: All-Tag-Prediction-w2v

We also tried to use word to vector as it converts the data into more meaningful sense compared to Tfidf."fig 12", "fig 13", "fig 14"

Model	Vectorizer	precision	recall	F1-score
SGD Classifier Log-loss	AVG-W2G	0.41	0.68	0.512
SGD Classifier Hinge-loss	AVG-W2G	0.398	0.694	0.506
Logistic Regression	AVG-W2G	0.418	0.674	0.516

Fig-13: Top3-Tag-Prediction-w2v

And performed the operations just as above for all, 5 and three tag predictions, we also applied this for the data set containning top 30 tags and got the following f1 scores as follows.

All-tag prediction - Logistic regression (gave highest score thatis 0.214)

Top 3 - Logistic Regression (gave highest score that is 0.563) Top 5 - Logistic Regression (gave highest score that is 0.516) Top 30 - data tags Logistic Regression (gave highest scorethat is 0.32)

As you can see this followed the same trend like when we used Tfidf. And Logistic Regression gave the best model outcomes in case of both vectorizers. Also top 3 prediction gave some what satisfying results in both. We also used Topic modelling (which finds the topics with in our synopsis) and LDA (latent dirichlet allocation) model to Implement this. Here also Logistic regression gave the highest result (0.366).

5.COMPARATIVE ANALYSIS

Here the approach that we have stated utilises One-Vs- Rest classifier to solve the problem as a multi-label (classification). A set of people [22] Khalid Haseeb, Najm us Sama, MiguelÁ. MartÍnez-Del-Amor, Adnan Ahmed, Umair Ali Khan,

Model	Vectorizer	precision	recall	F1-score
SGD Classifier Log-loss	AVG-W2G	0.443	0.764	0.561
SGD Classifier Hinge-loss	AVG-W2G	0.395	0.723	0.511
Logistic Regression	AVG-W2G	0.477	0.688	0.563

Fig-14: Top5-Tag-Prediction-w2v

Saleh M. Altowaijri, Naveed Islam and Atiq Ur Rehman who belongs to various fields and branches of Computer Science worked on the same problem (movie-tag prediction) with a different approach. The similar thing between us is we both felt that this field(predicting tags for movies) received less attention over years. They tried to predict the tags using segmentation of movies (movie-frames) and CNN (Convolutional-Neural- Network). We prepared a tag set of 71 tags(based on mostly used or occurring tags) and this team on the other hand prepared a tag set of 50 tags (dataset with combined information of each tag with 700 images and prepared a datasetwith semantic information).

We tried combinations (two algorithms) of different classifyingalgorithms like Logistic regression, Multinomial-NB, GCD with One Vs rest classifier to train our model using N grams (to get a semantic sense) and Vectorizers (Count, TFIDF, Word

to - Vec), they used Inception V3 - pretrained model (CNN) and modified the final layers of classification in the sense with Soft-max classification for pridictions (transfer learning) .

In our case we tried to predict based on the plot synopsis and tried to gather the feel of plot and predicted tags accordingly. But this team used segmentation to achieve this. In this processthey use videos of the movie and divide it into different frames and gather a single slide from the frame and try to predict 3 tags for this slide and they combine all the frames information and select top thre tags having higher weights or probabilities (dominant). They also designed a algorithm for boundary and key-frame detection and extraction. In our case we tried to classify our problem as a multilabel model and trieddifferent combination of One-vs-rest classifier(which classifies the model considering one feature to rest all the features)and other multiclass classification algorithms and tried with n- gram approach and used different type of vectorizers to get meaningful numerical data.

Our highest f1-score is a decent 0.583 and they have a really efficient score of 0.88 as their highest. Both these are in top three tag predictions. So there is still a improvement scope when compared with other variant models.

6. CONCLUSION

We described a way that allowed us to create a fine-tuned set of various tags that expose the varied characteristics of movie plots. We took on the task of extracting tags relating to movie plots from noisy and repetitive tag spaces which needed to be pre-processed. We present an analysis where we tried to predict movie tags from plot synopsis. The highest micro averaged F1 score that we obtained from the entire project is 0.583. As, we observed in the EDA section that on an average amovie contains 3 tags. So, w tried to train our model multiple times and each time we set the tags to different values liketop 3 and top 5 and we also selected the top 30 tags manuallyto analyze our model to a greater extent and we also used various semantics vectorizations such as LDA (Latent Dirichlet Allocation) and word2vec to improve our model. Although we had a small dataset, we still got a decent F1 score. By means of this project we are solving a Multi-Label Classificationproblem. We hope it will be useful in other tasks related to narrative analysis.

REFERENCES

[1] Sudipa Kar, Suraj Maharjan, A.Pastor Lopez-Monroy and Yhamar Solorio,23 Feb 2018,"Predicting tags for movies from plot synopsis using emotion flow encoded neural network" In 2018 international conference on semantic computing.IEEE.

[2] Ali Mert Ertugrul, Pinar Karagoz,2018,"Movie genre classification from plot summaries using Bidirectional LSTM".In 2018 12th international conference on semantic computing.IEEE.

[3]Antonius Christiyanto Saputra,Anjelina Br Sitepu,Stanley,Yohanes Sigit Purnomo W.P,2019," The Classification of the Movie Genre based on Synopsis of the Indonesian Film". In 2019 international conference on semantic computing. IEEE.

[4] Jingcheng Wang ,2020," Using machine learning to identify movie genres through online movie synopsis". In 20202nd international conference on information technology and computer application. IEEE.

[5] Darren Kuo,December 16,2011," On Word Prediction Methods". In 2011 international conference on semantic computing. IEEE.

[6] Weblog Posts,Gilad Mishne,May 23,2006," AutoTag: A Collaborative Approach to Automated Tag Assignment for Weblog Posts". In 2006 World wide web conference committee. IEEE.

[7]Teh Chao Ying, Shyamala Doraisamy, Lili Nurliyana Abdullah,2015," Lyrics-Based Genre Classification Using Variant tf-idf Weighting Schemes". Journal of Applied Sciences,15:289-294. IEEE.