Identifying movie genre compositions using neural networks and introducing GenRec - a recommender system based on audience genre perception

Aditya Pal
Computer Science and Engineering
Birla Institute of Technology, Mesra
Ranchi, India
aditya.pal.science@gmail.com

Abhilash Barigidad
Computer Science and Engineering
Birla Institute of Technology, Mesra
Ranchi, India
abhilash.ub@gmail.com

Abhijit Mustafi
Computer Science and Engineering
Birla Institute of Technology, Mesra
Ranchi, India
abhijit@bitmesra.ac.in

***Abstract*—Film industries all around the world continue to produce thousands of movies every year. Each such movie is generally classified into 2-5 genres on popular movie websites and streaming platforms. However, such genre classifications are based on the nature of the movie script or movie scenes alone and do not usually provide a measurement of the percentage of genres in the movie. In this work, we present an approach to break down a movie into its respective genre compositions based on viewer opinion using neural networks and introduce GenRec – a content-based movie recommender system using these opinion-based genre metrics. To train machine learning models, we have created the world's largest movie reviews to genres dataset with the help of the popular Internet Movie Database (IMDb). We analyze our dataset with ensemble classification algorithms such as Nearest Neighbors and Naïve Bayes and move on to ensemble logistic regression models in order to breakdown a movie into its genre compositions. We have also trained neural network models – DNN, CNN and RNN and found that DNN outperforms the rest with an R2-score of 0.825 and RMSE of 0.060. Finally, we designed our recommender system from thousands of movies by applying similarity algorithms on the audience perceived genre compositions of movies as obtained from the neural network. In addition, we also discuss use cases of how our genre composition system can be used for analyzing movie reception and talk about some of the future work that may be conducted with our dataset.**

Keywords— neural network, recommendation system, movie genre prediction, audience perception, machine learning

I. Introduction

Recommendation systems (RS) have now become an essential component of many online businesses, be it movie recommendations by video streaming platforms [1] or shopping product recommendations by retail giants [2]. These RS typically make use of an algorithm that is able to determine certain relationships between users and products in order to create recommendations using similarity models. Traditional recommendation systems can be classified into 3 main categories based on the algorithm used. The first technique is content based filtering (CBF) which uses the characteristics of an item and assumes that each user has a specific taste. CBF approaches work best where the characteristics that distinguish items are known or can be easily derived [3]. Next comes collaborative filtering (CF), which predicts similar items based on the preferences or opinions of multiple users (collaboration). It analyses the relationships between users and items to obtain similar profiles. However, CF approaches typically face a few challenges as outlined by Suganeshwari and Ibrahim [4]: a) Limited Content Analysis, b) Over-Specialization, and c) Cold Start. These problems are partly taken care of by hybrid systems which make use of a combination of CF and CBF approaches to predict items [5].

The term genre defines the category of a motion-picture based on similarities in narrative elements or in emotional response to a film. We find that most of the popular movie database websites such as the Internet Movie Database (IMDb) or other video streaming platforms like Netflix and Amazon Prime provide only 2-5 genres per movie that are used to denote the categories of the movie based on narrative content. Also, these platforms do not generally contain any well-defined metrics to evaluate the existence of a genre in a movie. It is a common experience that a movie cannot be completely broken into only the few genre elements as specified by these platforms. For example, let us consider a popular IMDb top 10 movie, The Dark Knight. The themes for the movie as given by IMDb are: 1. Action 2. Crime 3. Drama. A person who has watched the movie can confirm that these are indeed the main genres of the movie. However, this categorization of the movie into only the 3 genres is not very informative of the uniqueness of the emotional aspects of the movie as multiple movies have exactly the same genre set. A first-time viewer may have the following insightful questions when selecting a movie like The Dark Knight:

- Is the movie only made up of 3 genres? Are there no hints of any other elements, like comedy or horror?
- Is there any metric that can define the percentage of the genres? Is the movie more Action or Crime based?
- Were movie reviews taken into consideration to get the genres as perceived by the general audience?
- Did the genres perceived by users change before and after movie release due to any external event?

The vast majority of previously designed recommender systems utilize the ratings of a movie [6] to derive the algorithm for movie recommendation. Makita and Lenskiy [7] have previously proposed a Naive Bayes model for the prediction of movie genres with the help of user ratings of a movie. Recently, with advances in NLP techniques, a few review-based recommendation systems have been implemented to solve rating sparsity [8]. An attempt has been made in [9] to classify movie scripts using logistic regression and feature-extraction. Some features include comparing dialogue frames to non-dialogue frames or the ratio of descriptive language to nominals. The model has been trained on a very small dataset containing 399 scripts with the best

978-1-7281-9180-5/20/$31.00 ©2020 IEEE

feature subset achieving an F1 score of only 0.56. Some studies have also been conducted to use a combination of ratings, sentiments of user reviews and other aspects as well [10]. A few approaches to derive genres of a movie based on scene selections from movie trailers [11] and plot summaries [12] have also been proposed. However, there has been very little work done to establish audience opinion-based genre compositions of a movie. In this paper, we introduce a process to evaluate the genres of a movie based on viewers' opinions and recommend movies using this evaluation. We show how this process can answer the questions put forth previously.

II. Dataset Generation

The first step to train our machine learning models is to obtain a dataset with the following properties:

- Movie Reviews
- Movie Names
- Genre(s) of the movie

After extensive searching on popular dataset platforms and search engines, we found that majority of online movie review datasets contain only reviews for sentiment analysis with neither the genre nor the movie name listed in them. A few movie reviews to genre datasets have been created - however most of these have less than 30 movies with around 5000 reviews only. So, we generated our required dataset ourselves using the most popular and extensive source of movie data - the Internet Movie Database (IMDb) website. We used Python scraping libraries to extract movie details from the list of top 100 IMDb movies for 17 movie genres – Action, Adventure, Animation, Biography, Comedy, Crime, Drama, Fantasy, History, Horror, Music, Mystery, Romance, Sci-Fi, Sport, Thriller and War. In total, we have collected nearly 1 million unique reviews from 1150 unique movies, making it the world's largest public movie reviews to genres dataset. We have made our dataset openly available on IEEE DataPort[1]. "Fig. 1" contains a statistical description of movies and reviews count of each genre in our dataset. Note that the sum of individual review counts per genre does not equal the total reviews count because many movies are present as part of 2 or more genres. We have also collected reviews from 100 TV shows to test our model and recommender system.

III. Data Pre-Processing

Data pre-processing is the first step towards information retrieval from the input data. Movie reviews scraped from the IMDb webpages are noisy and have redundant text elements. We make each review go through 3 text pre-processing steps before they are used for further analysis:

A. Tokenization and Case Transformation

Sentences have been 'tokenized' or split into words based on whitespace, tabs, new lines and the text has been converted into a uniform lower case for pre-processing.

B. Noise Reduction

Raw data from websites contains HTML tags, numerical values, punctuations and URLs that are considered as noise for our ML models. These have been cleaned from reviews.

TABLE I. Dataset Description

Genre	Movie Count	Review Count
Action	300	338792
Adventure	279	319123
Animation	100	51446
Biography	150	76651
Comedy	304	157872
Crime	133	121386
Drama	563	420497
Fantasy	109	126248
History	108	53553
Horror	103	103770
Music	96	31799
Mystery	107	111049
Romance	135	84552
Sci-Fi	124	180867
Sport	100	27731
Thriller	151	143707
War	99	41134
Total	**1150**	**932464**

*Note: Sum of movie counts is not equal to total movies due to movies with multiple genres

Fig. 1. Dataset Description containing movie and review counts per genre

C. Normalization

Normalization makes text more uniform and involves removal of stop words and lemmatization of words to convert them into their base word on the basis of usage and context.

IV. Analysing The Dataset

In this section, we have provided a descriptive and quantitative analysis that we performed on our dataset to verify that the dataset is capable of producing features that can be trained by machine learning and deep learning models.

A. Keywords

We applied term frequency–inverse document frequency (TF-IDF) to represent the words in our dataset as a feature vector. Salton and Buckley, 1988 [13], Berger, et al, 2000 [14] have stated that, given a word w, document collection D, and an individual document d ϵ D, we calculate

$$w_d = f_{w,d} * \log(|D|/f_{w,D}) \qquad (1)$$

where $f_{w,d}$ equals the count of w in d, |D| is the number of documents in corpus, and $f_{w,D}$ equals the number of documents in which w appears in D. In our case, each review signifies a document and each genre is analogous to the corpus of text under consideration. We have shown a summary of some of the top relevant words of a few genres derived using TF-IDF on our dataset in "Fig. 2".

B. Baseline Classification Models

In this section, we evaluate the performance of a few strong baseline Machine Learning classification models on our dataset. We use these models to perform a multi-label classification into the 17 genres for any review taken out of our dataset. We have randomly shuffled the dataset and split it into train and test datasets having a ratio of 4:1. We train our baseline classification models on this train dataset and test our

[1] https://ieee-dataport.org/open-access/imdb-movie-reviews-dataset

final predictions using the test dataset. We have not used a validation dataset when doing the initial exploratory analysis.

TABLE II. KEYWORDS

Genre	Top 10 keywords
Action	densely, chatty, executed, thrill, slick, pumping, exciting, joyously, throbbing, superwoman
Crime	gangster, videographer, organized, consultation, docu, backup, psychological, committed, drama, rise
Drama	docu, dramatic, melodrama, intrigue, thriller, foxcatcher, gripping, exciting, caper, thrilling
Romance	tale, drama, fantasy, romantic, element, core, existent, pure, fairy, genre
War	anti, vietnam, semitic, demonized, readjustment, horror, condones, conflict, semitism, casualty

Fig. 2. Top relevant keywords per genre

We now go ahead with representing each review as a feature vector. We have used TF-IDF scores as a measure of feature values and fed the scores to some of the most popular classification techniques from the Python Machine Learning library scikit-learn [15]. We have applied Nearest Neighbours, Support Vector Machine with a linear kernel, Random Forest Classifier and Naïve Bayes models on our dataset. "Fig. 3" summarizes the performances of each of the text classifiers on the test dataset and evaluates the results into the following metrics – micro-average precision, micro-average recall and micro-average F-score. The definition of these metrics as stated in [12] are:

$$Recall = \frac{TP}{TP+FN} \tag{2}$$

$$Precision = \frac{TP}{TP+FP} \tag{3}$$

$$F-score = \frac{2 \times Precision \times Recall}{Precision+Recall} \tag{4}$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{5}$$

here, TP = True Positive, TN = True Negative, FP = False Positive, FN = False Negative. We have chosen subset accuracy as the accuracy metrics, which means that a correct prediction is indicated by a success on every genre of a review. Even if one genre is predicted incorrectly, the result is treated as an incorrect prediction which is stricter than the evaluation metric in the model defined by Makita and Lenskiy [7], where a prediction is regarded as a success if at least one of the genres matches with the true labels. We find that KNN has the highest classification accuracy of 89%. We can safely say that our movie review dataset is capable of providing features for genre extraction by baseline classification algorithms.

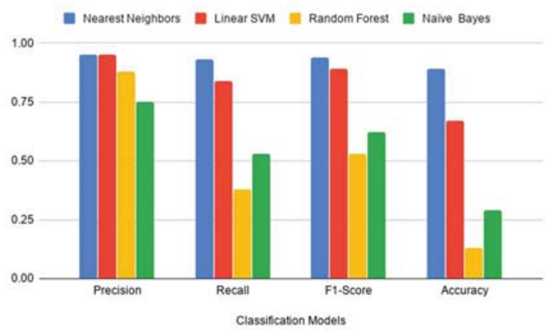


Fig. 3. Performance of Baseline classification models

V. GENRE COMPOSITION

In this section, we describe in detail the process of decomposing a movie into its respective genre compositions. This process differs from the typical multi-label genre classification of a movie and we cannot use the same binary labelled training and test datasets, since we want to obtain the percentage of genres available and not just the presence of the genre. To obtain an initial percentage of genres in reviews for our models, we transform the target values of our dataset from binary to continuous real values between 0 and 1. We initially assume that every movie review has an equal share of each of the predefined genres for that movie in IMDb. For example, "The Dark Knight" has genres Action, Crime and Drama – instead of a binary positive value in each of these genres, we transform the target values into Action – 0.33, Crime 0.33 and Drama 0.33. We use the following relation to establish input genre target values for each review:

$$gt_{d,i} = g_{d,i} / \sum_{j=1}^{j=k} g_{d,j} \tag{6}$$

where gd, i denotes the 0-1 binary value indicating the presence of genre i in document d (review in our case), and k denotes the number of genres under consideration (17 in our case). We choose d ϵ D which represents a document that appears in our genre reviews list D. Note that the sum of all the genre target values for one such review equals 1.0.

A. Baseline Regressor Models

Now that we have established a continuous target for each of our movie review, we can use this transformed dataset for training our model to get the desired genre compositions. First, we divide the dataset into a shuffled train and test sets in the ratio of 4:1. We generate TF-IDF matrix to model the feature values and feed the values to 4 baseline scikit-learn logistic regression models, namely Gradient Boosting Regressor, Random Forest Regressor with estimators (number of trees in the forest) as 5, Bagging Regressor with 10 base estimators and Extra Trees Regressor with 10 trees in the forest. The results of these standard models on our dataset have been provided in "Fig. 5". The accuracy metrics system chosen for multi-class logistic regression was R-squared score [16] for each model which is basically a statistical measure for calculating the proximity of data points from the fitted regression line. We have also calculated the Root Mean Squared Error (RMSE) for each of these baseline models, which is widely considered as a good measure for evaluating the error of a regression problem. Based on our experimental analysis, we find that Extra Trees Regressor with 10 trees has outperformed the other standard regressors with an R2-score of 0.542 and an RMSE of 0.1. Note that we have not used a validation dataset for hyperparameter tuning because these baseline models have R2-scores which are significantly lower than the Neural Network models trained in the next section.

B. Neural Network Based Models

After training the baseline regressor models, we explored neural networks for improved performance on the dataset. We have used the popular Python library TensorFlow [17] for training our neural networks. All neural networks have been trained after selecting hyperparameters on a validation dataset as shown in "Fig. 4". A description of implementation details of each neural network model – DNN, CNN and RNN have been depicted below and results shown in "Fig. 5".

TABLE III. HYPERPARAMETER TUNING OF NEURAL NETS

Hyperparameter	DNN	CNN	RNN
Learning Rate	0.001, 0.01	0.001, 0.01	0.001, 0.01
Number of Epochs	10, 15, 20	10, 15, 20	10, 15, 17
Nodes per Layer	128, 256, 512	128, 256, 512	128, 256, 512
Dropout Probability	0.3, 0.5, 0.7	0.3, 0.5, 0.7	0.3, 0.5, 0.7
Batch Sizes	64, 128, 256	64, 128, 256	64, 128, 256
GRU Units	-	-	32, 64, 128
Optimizers	SGD, Adam	SGD, Adam	SGD, Adam

Fig. 4. Hyperparameter ranges used during training neural network models

1) Deep Neural Network (DNN)

We used the TensorFlow framework to construct our Deep Learning Model. TF-IDF has been used to represent words as feature vectors. After testing various hyperparameters, including count of hidden layers, neurons per layer, learning rate of network, number of epochs and so on, our final DNN model consisted of 6 dense layers and 5 dropout layers having 512 nodes per layer, creating a total of 38 million trainable parameters with 0.986 validation AUC score. We used Adam optimizer over SGD to obtain the best R2-score.

2) Convolutional Neural Network (CNN)

A CNN model using TensorFlow was trained, but this time we used GloVe for word vector representation over TF-IDF. After training our model using the widely used pre-trained GloVe [18], we switched to a GloVe trained on our own dataset for much better results. Our final CNN model consisted of 7 Conv1D layers with filter sizes ranging from 2 to 6, an equal number of MaxPooling layers, 4 dropout and 3 dense layers having 128 nodes per layer, creating a total of 7.6 million trainable parameters with 0.97 validation AUC score.

3) Recurrent Neural Network (RNN)

To train our RNN model, we again opted for the GloVe previously trained as our word vector representation. We found the Gated Recurrent Unit (GRU) architecture [19] to perform slightly better than Long Short-Term Memory (LSTM) [20] nodes. We trained the RNN model with 4 GRU layers having 32 GRU nodes each, 4 dropout and 2 dense layers. At higher layer counts, training time of RNN increased exponentially due to the complexity of RNN architecture. We found that beyond 17 epochs training accuracy of the model started to decrease. Our final model produced 8 million trainable parameters with 0.937 validation AUC score.

We have provided training logs for the three neural networks with their AUC score per epoch in "Fig. 6".

TABLE IV. REGRESSION AND NEURAL NETWORK RESULTS

Model Name	R2-score	RMSE
Gradient Boosting Regressor	0.435	0.109
Random Forest Regressor	0.502	0.105
Bagging Regressor	0.498	0.105
Extra Trees Regressor	0.542	0.1
Deep Neural Network	0.825	0.060
Convolutional Neural Network	0.610	0.073
Recurrent Neural Network	0.592	0.076

Fig. 5. Results of experimental analysis of standard ensemble regressor models and neural network models

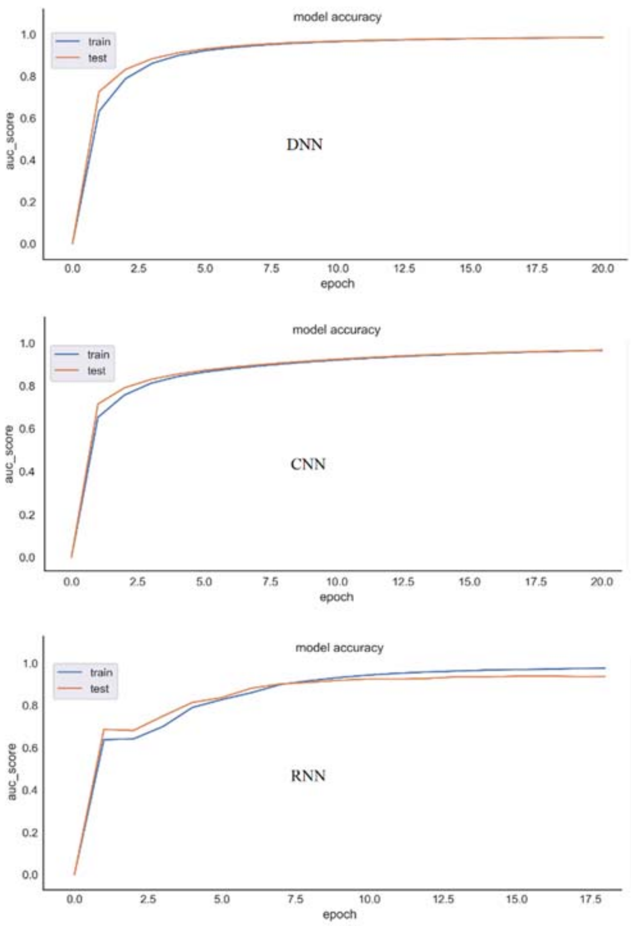


Fig. 6. Training logs and AUC score of neural network models

VI. GENREC RECOMMENDER SYSTEM

In this section, we present GenRec – world's first content-based recommender system using audience-perceived genre compositions of a movie. We explain the implementation details of each component in the GenRec architecture.

A. Base Model

The base machine learning model of the GenRec recommender system is the Deep Neural Network trained previously. We have trained the model with about 1100 movies and saved the model for recommending movies.

B. Input Parameters

GenRec takes two main parameters as inputs – the number of movies to be recommended and a variable second parameter of three types:

1) Name and year of a movie

A user may enter the name and year of the movie as the first input type to GenRec. We check if the user perceived genre compositions of the input movie are already stored in our database. If present, we apply similarity methods with all the stored movie genres from our database and output top similar movies. If the movie is not present in our database, we first download the user reviews of the movie from IMDb, pre-process them, create TF-IDF feature vectors and generate its genre compositions using our Deep Learning Model. An additional step here is to persist this genre composition along with the movie name and year in the database, thus ensuring that future searches of the movie do not go through the repetitive steps of downloading and running the model.

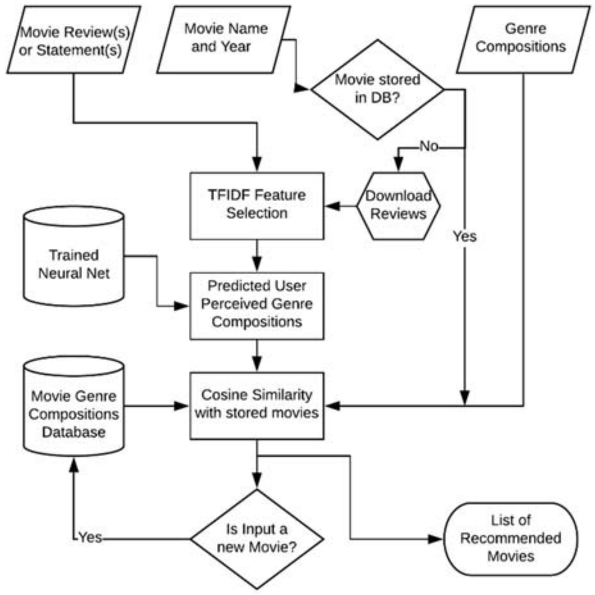


Fig. 7. GenRec recommender system high-level architecture

2) Statement(s) / Review(s) of a movie:

A user may choose to enter review(s) or statement(s) to GenRec. After pre-processing, we feed the TF-IDF feature vector into our model and genre compositions are generated. Similarity methods are applied on the input genre compositions with the ones stored in our movie database and top movies are recommended. Note that we do not persist this data to database in this case because these are not genre compositions from a full set of reviews of a new movie.

3) Genre Compositions of a movie:

The final input type is simply the genre compositions itself. The user may choose real values in the range 0-100 for all the 17 genres such that the summation of all the individual genre elements equals 100. A simple example of such an input would be "Action – 20%, Horror – 50%, Thriller – 30%" and rest 14 genres as 0%. We would apply similarity techniques directly to this input genre compositions with the stored movies in our database and recommend top matching movies.

C. Similarity Methods

As recommender systems have improved with time, a number of similarity methods have been defined as explained in the study [21]. Among all the various methods established, we have used Cosine Similarity – a widely used method for calculating similarity between two vectors. A few advanced similarity measures like the inverse Euclidean distance have been proven to be better for movie recommender systems [22] when the users have rated null or not reviewed a movie. However, in our case, we have established the fact that all 17 genres have a certain non-negative value and the input genres are non-negative real values having sum equal to 100. Hence, we chose cosine similarity for GenRec to derive the movies having the highest match with the input audience-perceived genres. Cosine Similarity is defined as follows:

$$\mathrm{sim}(\mathrm{a},\mathrm{u}) = \cos(\mathrm{r_a},\mathrm{r_u}) = \frac{\sum_{j=1}^{j=m} \mathrm{r_{a,j}} * \mathrm{r_{u,j}}}{\sum_{j=1}^{j=m} \mathrm{r^2_{a,j}} * \sum_{j=1}^{j=m} \mathrm{r^2_{u,j}}} \quad (7)$$

Here, r_a and r_u denote the genre compositions of movie a and user input movie or review u. m denotes the number of elements in a genre composition which equals 17 for our case.

D. Database persistence

We use a database to store user-perceived genre compositions of movies. In the first run of GenRec, we have stored around 1150 movies, their year of release and their genre compositions as trained by the model. As users seek recommendations of newer movies, we keep adding the new movie, its year of release and its genre composition. We prefer standard RDBMS over NoSQL databases since we need ACID properties for fast read and write into the database.

E. Output - Movies Recommended

Using all these components together, we are able to generate the desired number of recommended movies which are displayed in a descending order of similarity values.

"Fig. 7" shows the individual components in the GenRec recommendation system architecture.

VII. Results and Use Cases

In this section, we present the user-perceived genre compositions obtained from a few IMDb movie and TV series. Note that we have not trained our model on any TV series-based reviews. Hence, we wanted to test our model on both movie and series reviews to check if user perceived genres match closely with the genres stated by IMDb. "Fig. 10" contains a list of sample reviews taken from movies and TV shows, their true labels as identified by IMDb and the genre compositions provided by our Deep Neural Network model. Note that we have only plotted the genres having at least 10% contribution to the overall genre composition while the remaining genres are cumulated in an "Others" category to prevent visual clutter. We have also used GenRec to predict top 5 Database movies based on these user-perceived genre compositions. It is important to mention that these individual reviews often provide skewed information about a movie and it is advisable to use the full movie reviews set for best results.

Here is a brief description of a few additional use cases where we show how our prediction model can provide some valuable insights and analysis on movie reception.

A. Comparison of User Reception post vs pre release

Few movies are pre-screened for testing by audiences before the actual release takes place. The viewers may express their opinions and write their reviews for the director and crew before the actual release takes place. This list of reviews can be gathered from the users and analysed for their genre compositions before the release. Later, we can collect user reviews post the release and analyse the opinions of the audience pre and post the release. This comparison may yield valuable insights such as effect of time of release on the reception of a movie. Director and his movie crew may find the analysis helpful in identifying why the audience may have perceived a horror-comedy movie to be more on the comical side rather than horror, when he may have intended the script to be more horror based. Directors may analyse their script and improve it based on genre perception by target audiences.

B. Impact of an external event on movie genre perception

A movie's success or failure may depend on several external factors or events as well. Our genre perception system will be able to identify the impact of an event on the movie.

While testing our model on several movies and TV series, we stumbled upon one such tragic event that had a major impact on audience perception. A Bollywood movie named "Chhichhore" was released in September 2019. The movie is about a middle-aged man who reminisces his college days after his son attempts to commit suicide. The main genres of the movie as outlined by IMDb are Comedy and Drama. However, when we applied our model on the IMDb movie reviews, we found that biography turned out to be one of the main genres of the movie. The movie was never intended to be biographical. But when we took a closer look, we found a large number of reviews after June 2020 talking about the lead character Sushant Singh Rajput's tragic death by suicide on 14th June 2020. We figured that the comparisons between the life of the actor and that of the movie character had caused the genre compositions to be more biographical than expected. In "Fig, 8", we compare the genre compositions of Chhichhore derived from user reviews before the actor's demise versus all reviews after his death. The clear difference in the audience perception of Chhichhore before and after Sushant Singh Rajput's death provides an insight into change in user genre perceptions due to a single event.

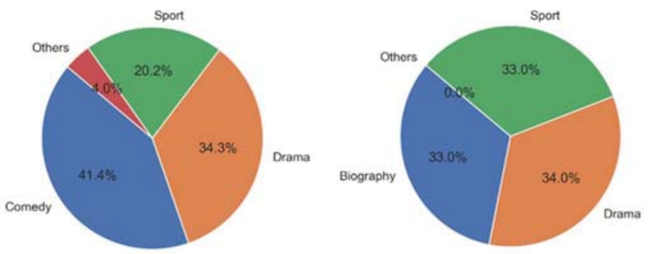


Fig. 8. Chhichhore's genre compositions before and after Sushant's demise

C. *Identification of genres missed by directors or movie platforms, but captured by user reviews*

Our genre-based composition metrics system has been successfully used to identify genres that have been missed by IMDb. One such example is the 2020 released TV series "The Last Dance". The series shows the rise of the 1990's Chicago Bulls, a basketball team led by Michael Jordan. Basketball was a major theme of the show and so, sport should have been one of the main genres of the series. However, the genres listed by IMDb are Drama and Biography. Our model accurately identifies Sport as a top genre as shown in "Fig. 9".

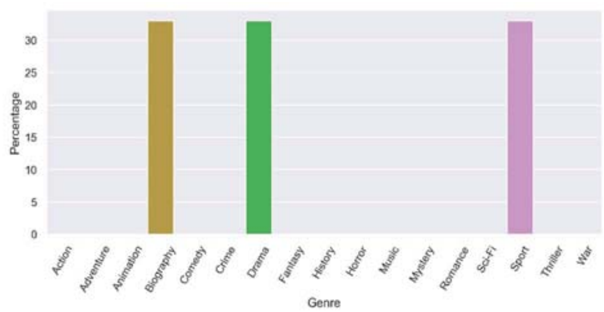


Fig. 9. Audience perceived genre compositions of "The Last Dance (2020)"

VIII. FUTURE WORK

Although we have used popular Machine Learning models for genre breakdown, we would like to explore the latest Neural Network models like RCNN and RMDL. Our model can also be integrated with movie metadata like plot summaries, movie posters as well as user reviews from other platforms like Google. We have manually evaluated GenRec for a few reviews, but we would like to perform further analysis and comparison with other recommender systems. Even though GenRec is a content-based recommender system, it suffers from the cold start problem because new movies may not have sufficient user reviews for processing. To avoid such cases, we can use a combination of director / movie platform-based genres and genre compositions from our model to breakdown the movie until more user reviews are available.

ACKNOWLEDGMENT

We would like to thank Google Colaboratory [23] for providing powerful hardware for training our ML models. The research would not have been possible without the Internet Movie Database's (IMDb) publicly available movie reviews.

REFERENCES

[1] Gomez-Uribe, Carlos A., and Neil Hunt. "The netflix recommender system: Algorithms, business value, and innovation." *ACM Transactions on Management Information Systems (TMIS)* 6.4 (2015): 1-19.

[2] Linden, Greg, Brent Smith, and Jeremy York. "Amazon. com recommendations: Item-to-item collaborative filtering." *IEEE Internet computing* 7.1 (2003): 76-80.

[3] Van Meteren, Robin, and Maarten Van Someren. "Using content-based filtering for recommendation." *Proceedings of the Machine Learning in the New Information Age: MLnet/ECML2000 Workshop*. Vol. 30. 2000.

[4] Suganeshwari, G., and SP Syed Ibrahim. "A survey on collaborative filtering based recommendation system." *Proceedings of the 3rd International Symposium on Big Data and Cloud Computing Challenges (ISBCC–16')*. Springer, Cham, 2016.

[5] De Campos, Luis M., et al. "Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks." *International journal of approximate reasoning* 51.7 (2010): 785-799.

[6] Wei, Shouxian, et al. "A hybrid approach for movie recommendation via tags and ratings." *Electronic Commerce Research and Applications* 18 (2016): 83-94.

[7] Makita, Eric, and Artem Lenskiy. "A multinomial probabilistic model for movie genre predictions." *arXiv preprint arXiv:1603.07849* (2016).

[8] Chen, Li, Guanliang Chen, and Feng Wang. "Recommender systems based on user reviews: the state of the art." *User Modeling and User-Adapted Interaction* 25.2 (2015): 99-154.

[9] Blackstock, Alex, and Matt Spitz. "Classifying Movie Scripts by Genre with a MEMM Using NLP-Based Features." *Citeseer* (2008).

[10] Diao, Qiming, et al. "Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS)." *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2014.

[11] Simões, Gabriel S., et al. "Movie genre classification with convolutional neural networks." *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2016.

[12] Hoang, Quan. "Predicting movie genres based on plot summaries." *arXiv preprint arXiv:1801.04813* (2018).

[13] Salton, Gerard, and Christopher Buckley. "Term-weighting approaches in automatic text retrieval." *Information processing & management* 24.5 (1988): 513-523.

[14] Berger, Adam, et al. "Bridging the lexical chasm: statistical approaches to answer-finding." Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval. 2000.

[15] Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." *the Journal of machine Learning research* 12 (2011): 2825-2830.

[16] Cameron, A. Colin, and Frank AG Windmeijer. "An R-squared measure of goodness of fit for some common nonlinear regression models." *Journal of econometrics* 77.2 (1997): 329-342.

[17] Abadi, Martín, et al. "Tensorflow: A system for large-scale machine learning." *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*. 2016.

[18] Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global vectors for word representation." *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014.

[19] Chung, Junyoung, et al. "Gated feedback recurrent neural networks." *International conference on machine learning*. 2015.

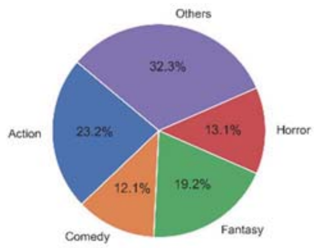
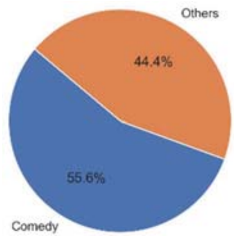
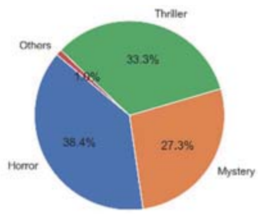
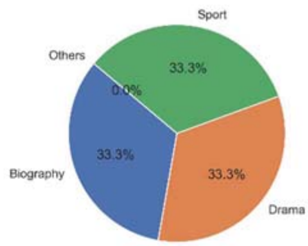
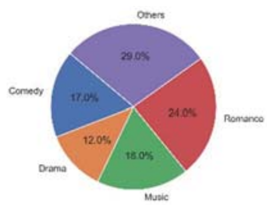
[20] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." *Neural computation* 9.8 (1997): 1735-1780.

[21] Agarwal, Ajay, and Minakshi Chauhan. "Similarity measures used in recommender systems: a study." *International Journal of Engineering Technology Science and Research IJETSR, ISSN* (2017): 2394-3386.

[22] Schwarz, Mykhaylo, Mykhaylo Lobur, and Yuriy Stekh. "Analysis of the effectiveness of similarity measures for recommender systems." *2017 14th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM)*. IEEE, 2017.

[23] Bisong, Ekaba. *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. Apress, 2019.

TABLE V. EXAMPLES OF GENREC RECOMMENDATIONS FOR SELECTED USER REVIEWS

<table>
<tr><th>User Review</th><th>IMDb Genres</th><th>Predicted Genre Compositions</th><th>GenRec Recommended Movies</th></tr>
<tr><td>I saw Green Lantern with my son because it looked safe enough and I used to collect the comic books when I was a kid. I felt like I was the only guy over 20 at the theater, which I guess isn't surprising anymore. Did think there would be more adults, however, because the Green Lantern was something we read in my generation. Not sure if it's even around anymore. Being a fan of the comic book I had to go see it, and I was intrigued with the movie. After I got home I jumped on the computer and read the reviews. I was very disappointed with the professional reviews that cast stones at it, while at the same time, regular folk like myself thought it was a pretty good stab at the DC comic book version. I guess I look for something different when I go to the movies. I go to unwind and enjoy myself for a couple of hours, and Green Lantern worked.</td><td>Action
Adventure
Sci-Fi</td><td>
</td><td>This is Ghost Rider (2007)
Hellboy (2004)
Underworld (2003)
Blade II (2002)
Underworld Awakening (2012)</td></tr>
<tr><td>There were more comedies like that - crossing over the political correctness line, but not over the line of good taste, so as to make a movie hard to watch.
The film is quite pointy, it is for a moderate audience and will make all kinds of extremists quite unhappy. The plot is well thought out and funny. It'll nail you to the screen.
I like the unorthodox character of this film, Sacha Baron Cohen's movies are definitely one of the kind. I wouldn't consider the humor crude, but it is certainly honest. Some of the unthinkable things you see are likely to have happened in reality.</td><td>Comedy</td><td>
</td><td>This is the End (2013)
The Hangover Part II (2011)
The Hangover (2009)
The Devil Wears Prada (2006)
The Breakfast Club (1985)</td></tr>
<tr><td>I'm quite a horror junkie, and it's so rare to find such an engaging horror series. Fantastic atmosphere, the creepy bits weren't overdone so none of it seemed hokey, acting was spot on, and in the end I didn't feel cheated by the plot line (no holes or strings left dangling, I really hate when that happens). Usually I can tear apart horror movies because of inconsistencies or impractical things but I saw none when I binge watched this show. Just fantastic all around, and I really enjoyed the way it was told, the past and present and how it flowed so well. I recommend this 100%.</td><td>Drama
Horror
Mystery</td><td>
</td><td>Us (2019)
The Others (2001)
The Conjuring (2013)
The Conjuring 2 (2016)
Sinister (2012)</td></tr>
<tr><td>This is a fascinating documentary into modern football. I remember watching a similar programme about Neil Warnock's Sheffield Utd years ago and I suppose if you put this side by side with that it would paint a glaring picture of the way the sport, its players and its managers have changed forever. It's obvious from the outset that money drips off every part of the top level of this sport but it's interesting to see how it has been used to shape footballers into almost superhuman athletes with the best marketing in the world. The reason I enjoyed this documentary so much was mainly the insight it gives into Pep Guardiola. I have obviously heard all the cliches about the man before but this was the first time I felt actually understood why he has been so successful. Yes he has great teams and yes he gets huge amounts of financial support but his passion and outstanding man management are undeniable. Every scene featuring Guardiola is hugely watchable and often gripping.</td><td>Sport</td><td>
</td><td>Unbroken (2014)
The World's Fastest Indian (2005)
The Hurricane (1999)
The Greatest Game Ever Played (2005)
The Fighter (2010)</td></tr>
<tr><td>Ironically, Kumail Nanjiani, the lead actor in this film stands out as a terrible actor against the fantastic talent of Zoe Kazan, Ray Romano, and Holly Hunter. I say, "Ironically," because it is the lead actor's project and story! Zoe Kazan is adorable and charming as his love interest, but he is flat and wooden, which has the effect of creating zero chemistry between the lovers in this romance.</td><td>Romance
Drama
Comedy</td><td>
</td><td>To Rome With Love (2012)
The Wedding Singer (1998)
Music and Lyrics (2007)
A Night at the Roxbury (1998)
The Legend of 1900 (1998)</td></tr>
</table>

*Note: Only genres having at least 10% contribution have ben plotted. The rest have been cumulated in an "Others" category for a cleaner chart

Fig. 10. Examples of User-Perceived Genre Compositions and Movie Recommendations Using GenRec