Netflix Genre Predictor

Noah Ferrel, Raha Pirzadeh, Michael Kulinich
CS 530: Data Mining
Dr. Maoz
ferre150@mail.chapman.edu, pirzadeh@chapman.edu, kulinich@chapman.edu

***Index Terms*—Natural Language Processing, Neural Networks, Word Embedding, LSTM**

I. Introduction

Netflix has been the go-to streaming service for movies and TV shows since launching in the 2010s [1]. By having a recommendation system tailored to each of their users, Netflix's algorithm picks films and shows based on similarity and other users' votes [2]. Flixable, a third-party Netflix search engine, collected the data. The data contained 7,787 samples and 12 columns before data cleaning. The columns correspond to the show's unique identification number, type of content, title, director, cast, country, date added, release year, rating, duration, listed_in, and description. The 'type of content column' is categorical with two levels, TV show and Movie.

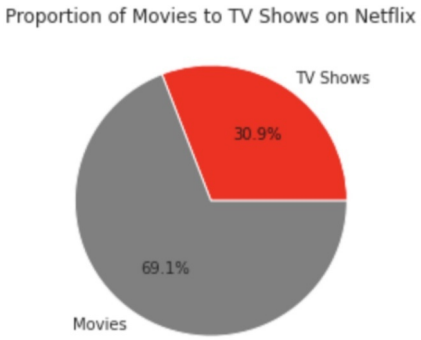

Fig. 1. Proportion of Movies vs TV shows

The 'date added' column contains dates in M, D Y format that corresponds to the date the content was added to Netflix. The 'rating' column is categorical, corresponding to the age content rating with fourteen levels. The cast, county, and listed in columns contain string lists. With the data-set, we built a model that would classify the genre of a given movie or TV show based on the description of the movie or TV show using natural language processing techniques.

We assumed that the vectorized word embeddings would provide insight into what genre the content is in. This model is used to either categorize new media added to Netflix's library by genre or suggest new content to users based on their genre preference.

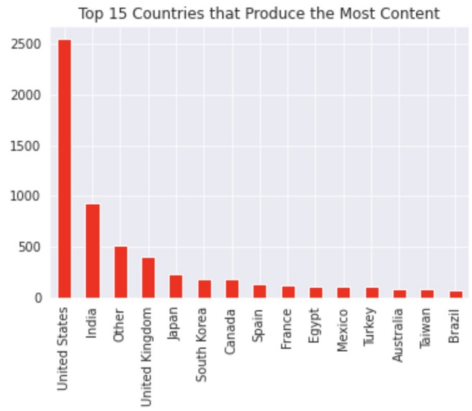

Fig. 2. Top 15 Countries

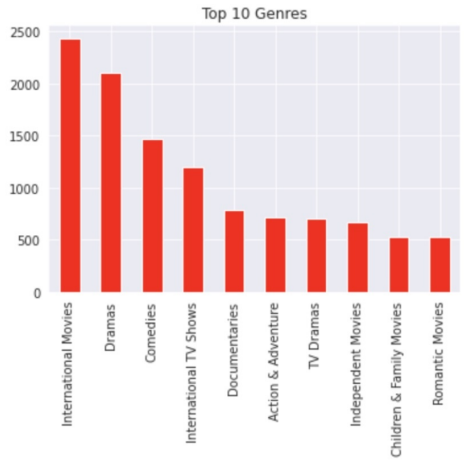

Fig. 3. Top 10 Genres

II. Methods

The data-set had over 2000 null values, primarily in the director, cast, and country columns. We opted to drop the rows containing the null values for the columns with ten or fewer data points missing. For the null values in the Cast and Director columns, we filled the nulls with the concatenation of the column name and the title of the feature. This was done because we wanted to arbitrarily fill in these nulls so that further analysis could be done using this information. The content rating column had fourteen columns which were converted into three levels; Mature, PG, and Kids.

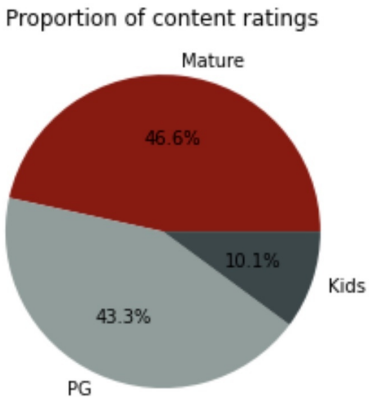

Fig. 4. Age Content Proportion

A problem encountered in the categorical columns, genre, and country. Multiple options were listed in each entry of the column. To build the genre predictor, we would need to convert the lists to a form the model could read. To achieve this, we added each genre and country to its own column. This allowed each movie or TV show in the data frame to have a binary value corresponding to its genres and countries (figure 6).

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	TV Show	3%	NaN	João Miguel, Bianca Comparato, Michel Gomes, R...	Brazil	August 14, 2020	2020	TV-MA	4 Seasons	International TV Shows, TV Dramas, TV Sci-Fi &...	In a future where the elite inhabit an island ...
1	s2	Movie	7:19	Jorge Michel Grau	Demián Bichir, Héctor Bonilla, Oscar Serrano, ...	Mexico	December 23, 2016	2016	TV-MA	93 min	Dramas, International Movies	After a devastating earthquake hits Mexico Cit...
2	s3	Movie	23:59	Gilbert Chan	Tedd Chan, Stella Chung, Henley Hii, Lawrence ...	Singapore	December 20, 2018	2011	R	78 min	Horror Movies, International Movies	When an army recruit is found dead, his fellow...
3	s4	Movie	9	Shane Acker	Elijah Wood, John C. Reilly, Jennifer Connelly...	United States	November 16, 2017	2009	PG-13	80 min	Action & Adventure, Independent Movies, Sci-Fi...	In a postapocalyptic world, rag-doll robots hi...
4	s5	Movie	21	Robert Luketic	Jim Sturgess, Kevin Spacey, Kate Bosworth, Aar...	United States	January 1, 2020	2008	PG-13	123 min	Dramas	A brilliant group of students become card-coun...

Fig. 5. Raw Data

rating	duration	listed_in	description	month_name_added	year_added	Other	United Kingdom	India	United States	Rating_Mature	Rating_PG	Rating_kids	type_Movie	type_TV Show	Cult Movies	Stand-Up Comedy & Talk Shows	Action & Adventure
TV-MA	4 Seasons	International TV Shows, TV Dramas, TV Sci-Fi &...	In a future where the elite inhabit an island ...	August	2020	1	0	0	0	1	0	0	0	1	0	0	0
TV-MA	93 min	Dramas, International Movies	After a devastating earthquake hits Mexico Cit...	December	2016	1	0	0	0	1	0	0	1	0	0	0	0
R	78 min	Horror Movies, International Movies	When an army recruit is found dead, his fellow...	December	2018	1	0	0	0	1	0	0	1	0	0	0	0
PG-13	80 min	Action & Adventure, Independent Movies, Sci-Fi...	In a postapocalyptic world, rag-doll robots hi...	November	2017	0	0	0	1	0	1	0	1	0	0	0	1
PG-13	123 min	Dramas	A brilliant group of students become card-coun...	January	2020	0	0	0	1	0	1	0	1	0	0	0	0

Fig. 6. Cleaned Data

As preliminary data exploration, We performed a random forest classifier and a multinomial logistic regression to see whether movies and TV shows could be classified by content rating based on where it was filmed, the genre, the type of feature, the release year, and when the content was added to Netflix. The random forest classifier had a mean validation accuracy of 0.67, and the multinomial logistic regression model had a mean validation accuracy of 0.68.

Another preliminary model we used was a Word2Vec analysis, a technique for natural language processing. The algorithm uses a neural network to learn word association from a large corpus of text. Our model used a pre-trained model from Google's news source. Using Word2Vec, we wanted to know about movie and TV show descriptions' relationships to one another through their respective embeddings. Using the pre-trained word embeddings, the model generated 300-dimensional word vectors.

	0	1	2	3	4	5	6	7	8	9	10
future	-0.080566	-0.012817	0.143555	0.051758	-0.062500	-0.042236	0.107422	-0.066895	0.200195	0.144531	-0.263672
elite	-0.101562	0.162109	-0.010132	0.138672	-0.003067	-0.021606	-0.343750	-0.484375	-0.159180	-0.080078	0.123047
inhabit	0.091797	-0.037598	-0.114258	-0.008545	-0.192383	0.100098	0.137695	-0.022583	0.057129	0.257812	-0.166016
island	0.037109	-0.056396	-0.126953	-0.055420	-0.025879	-0.050049	-0.161133	-0.202148	0.125977	0.231445	0.059570
paradise	0.103027	0.021851	0.033691	0.107422	0.163086	0.071777	-0.068848	-0.092773	-0.229492	0.185547	0.094238
far	-0.025391	-0.001587	-0.028320	0.054443	0.040771	-0.004089	0.069336	-0.085449	0.253906	0.133789	0.018921
crowded	0.019409	0.169922	-0.213867	0.135742	-0.178711	-0.257812	-0.156250	0.041748	-0.116699	0.089844	0.016724
slums	-0.094238	0.188477	0.320312	0.124023	-0.011963	0.018555	-0.074707	-0.032715	-0.130859	-0.139648	0.013123
get	0.033203	-0.089844	-0.294922	0.115234	-0.071289	-0.053955	0.010498	0.119141	0.024536	0.080078	-0.150391
one	0.045654	-0.145508	0.156250	0.166016	0.109863	0.007507	0.073730	-0.031006	0.157227	0.099609	0.052002

Fig. 7. Vectorized Words

The vectors are created by cosine similarity, allowing us to calculate vectors' similarity (figure 7) easily. To visualize the patterns, we dimensionally reduced word vectors using t-SNE. T-SNE creates two-dimensional maps from data with hundreds of dimensions, in our case 300.

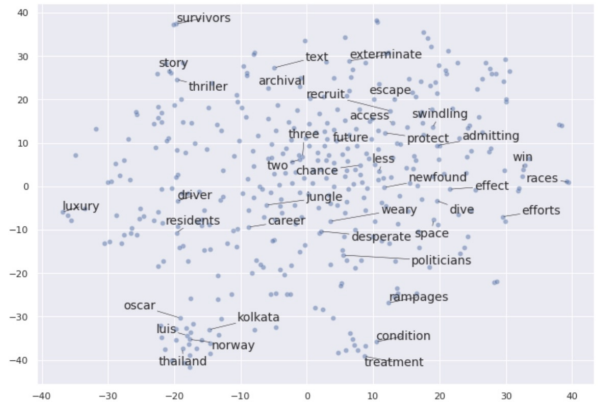

Fig. 8. Plot of Word Embeddings in two Dimensions

After making a list of the words in each string and removing duplicate words, stopwords, punctuation, and words not in the vectorized list, 15,000 words are left. T-SNE has many features that, with the smallest change, alter the entire plot. We found that a perplexity of 15 produced clean results. For better visualizations, we used smaller sets to see the clusters that appeared (figure 8). In figure 8, countries clustered in the lower-left corner, while numbers such as two and three clustered in the middle of the plot.

We decided to use a neural network for the genre predictor, but had to preprocess the data further. The dataset had 42 genres, with some being unrecognizable by just reading the description of the feature. Several of the genres were omitted, while others grouped together. For example, international movies were dropped altogether, and documentaries and docu-series were combined. Figure 9 shows the 13 genres we were

dramas	2808
comedies	1988
documentaries	1138
childrens	943
action and adventure	870
romantic	864
thrillers	541
standup comedy	381
horror	381
music and musicals	321
scifi and fantasy	293
reality	222
sports	196

Fig. 9. Counts per Category

left with that were used for the genre predictor.

After we are left with the final 13 genres, we use sklearn MultiLabelBinarizer to transform the text labels to a binary vector indicating the presence of a class label.

We then performed preprocessing on the input description text data to get it ready for the neural network. First, we removed special characters, punctuation, and 'stopwords' from the text. Stopwords are common words that carry far less meaning than other keywords in the text and add noise to the data. Neural networks do not input raw text, so we need to convert the text into numeric tensors. Vectorization is the process of transforming text into numeric tensors. We perform this by breaking the text by each word into tokens. Tokens are a numeric representation of the word; thus, we are left with sequence tensors in the place of text descriptions, which are fed into the neural network. To associate a vector with a token, we use dense word vectors known as word embeddings. Word embeddings are a low-dimensional floating-point vector that represents a given word. These embeddings are learned through training and create a structured embedding space. The embedding space holds geometric relationships between each vector that should reflect the semantic relationships between words. Since our dataset is small, we use pre-trained word embeddings. Pre-trained word embeddings are trained on millions of samples and create a highly structured spatial representation of words in the English language. A popular database with word embeddings that we use is called Global Vectors for Word Representation, GloVe [7]. For our models, we used the smallest database of 400,000 words that are 50 dimensional. Each word in the database is represented using a 50-dimensional vector.

Moving on to our model building, we started with a few simple neural networks, one with LSTM layers and another with Conv1D layers. We chose to try LSTM and 1D CNN because both have been proven to work very well with text sequence data. They can understand context along with the sequences of text and incorporate time series calculations. Each model that we built also included a pre-trained embedding layer. For the simple LSTM model, we chose to have one LSTM layer with 32 units, followed by two dense layers with dropout in between each layer. Dropout is a regularization technique for reducing overfitting by preventing complex co-adaptations on training data. A dropout rate of 0.5 has been proven to be effective in most scenarios [4]. We chose to have two Conv1D layers for the simple CNN model, each with 32 filters, followed by two dense layers with dropout in between each layer. The results of these two models will be discussed in the results section, but they did not perform as well as our final model.

Our final model consisted of a multi-channel neural network with a combination of LSTM layers and Conv1D layers. The three parallel channels read in the input text and then are merged together. We used this architecture to capture different abstractions of the text data. The first channel included two Conv1d layers with a kernel size of 2 and relu activation. A kernel size of 2 means that each filter scans 2 words at a time. We also use MaxPooling1D to perform max pooling operations over 1D temporal data. Max pooling reduces the dimensionality by a factor of 2 by taking the max value from each pool. We used the same architecture as the first channel for the second channel but with different hyperparameters. The conv1D layers have a kernel size of 6. The reason for the change of the kernel size is because we want both subnetworks to capture different numbers of words at a time. This can help the model learn contexts of varying sizes and understand various abstractions. The third and final channel consisted of a simple LSTM network. We added an LSTM channel to take the benefits of an LSTM and apply it to our network. Combining both Conv1D and LSTM layers into our model allows it to get the performance advantages from both architectures packed into one model. This channel included just one Bidirectional LSTM layer with 32 units. A bidirectional LSTM is an extension of the regular LSTM layer that can help improve model performance. It improves performance because it handles inputs in two ways, one from past to future and one from future to past. The layer has 32 units because any sizes smaller or larger did not perform as well. Next, each of these channels was flattened then merged together. The combined layer has a total of 2576 units which are then fed into a 64 unit dense layer and then finally into the output 13 unit dense layer. Between each of these layers is a dropout of .5. The output layer uses a sigmoid activation function. We are using sigmoid because our classes are not mutually exclusive, and each class is independent. An element belonging to a particular class should not influence the decision for another class. With this reasoning, we also use the binary cross-entropy loss function because a multi-label classification problem is set up as a binary classification problem for each of the 13 classes. Figure 10 on the next page shows the whole architecture of our model.

III. Results

The metrics used for evaluating the accuracy of our models was Micro-Average F1 Score. Since our classification problem is Multi-Label, we need to change our accuracy evaluation method. The reasoning behind this is because the model should still be rewarded if it predicts a subset of the genre categories correctly. Micro average F1 score aggregates the contribution of classes to compute the average metric

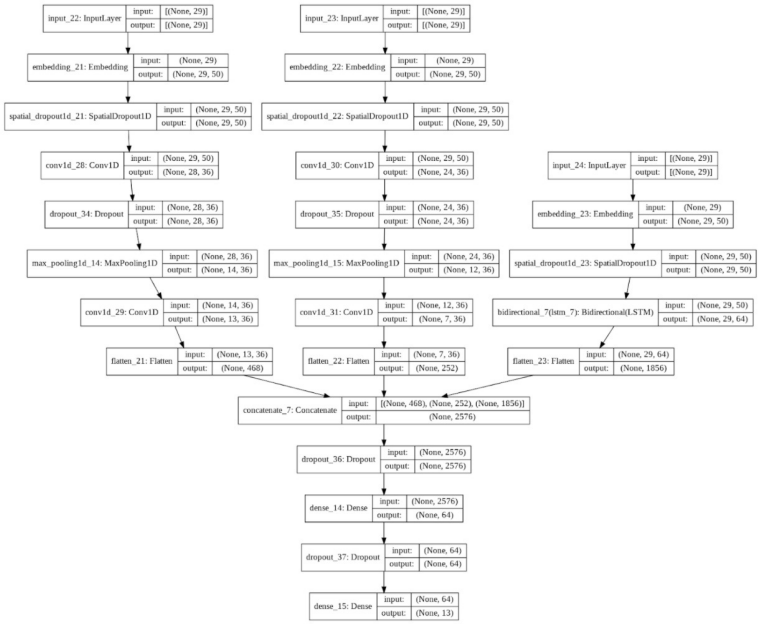

Fig. 10. Ensemble Architecture with three channels

For our initial models, after implementation of the genre predictor using an LSTM model and a CNN, it was found that the LSTM performed better based on the preliminary results. The micro-average score of the Conv1D network was performed at .45. The LSTM model had a micro-average score of .51. Both of these models were trained for 15 epochs and a batch size of 64. Also, both models lead to overfitting quickly after just a few epochs. After implementing the genre predictor using our final model, which was a combination of both CNN and LSTM layers, we received the best micro-average score of .61. The model was trained for 30 epochs and with a batch size of 64. After about 15 epochs, the model began to overfit, but since we used built-in keras checkpoints, we could save the weights of the model when the validation accuracy was at its highest. To examine the benefit of using pre-trained GloVe word embeddings, we ran the same model without the GloVe word embeddings. This means that the embedding layer was randomly initialized and was learned from scratch on our dataset. The results are evident and demonstrate the performance boost of pre-trained word embeddings.

Since the GloVe word embeddings were trained on millions of samples, they are able to capture the semantic and syntactic representation of words in the English language. It is shown in Figure 13 how the validation accuracy is much higher than the validation accuracy in Figure 8, and the model does not become severely overfitted to the training data.

We also looked at another evaluation metric for our model. Instead of allowing the model to predict one or several genres per sample, we limit the prediction to just one genre. This is done by choosing the class in which the model is most confident. With this evaluation, the model correctly predicted

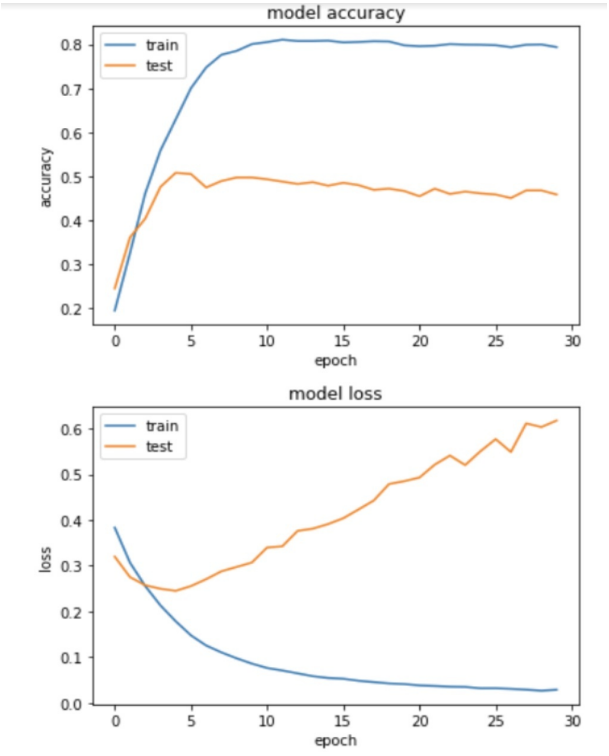

Fig. 11. Without pretrained word embeddings

a genre of a sample 67% of the time. To understand this accuracy, we need to compare it with the random chance level. Since we have 13 classes, each sample can fall under one - three classes. Therefore, randomly selecting a single class per

sample would result in accuracy between 7% - 23%. This is much lower than our model's performance.

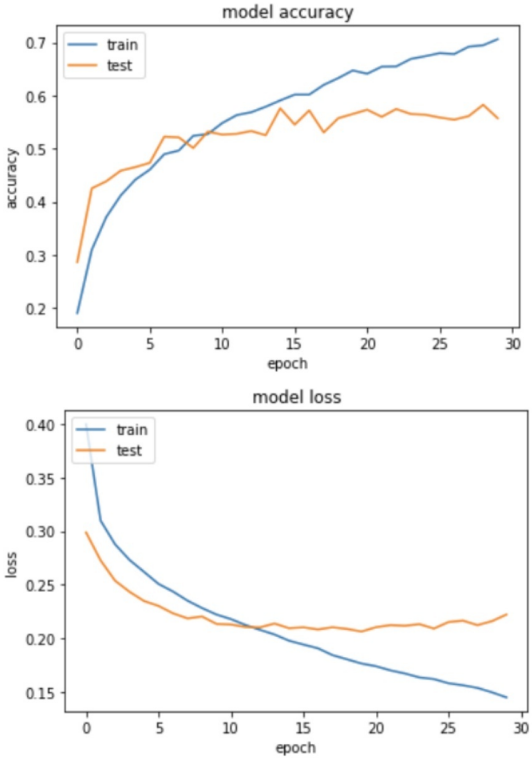

Fig. 12. With pretrained word embeddings

IV. Discussion

For future work, we plan to use a fully connected neural network to classify the genres. There was also a class imbalance, so we plan to add weights to the different classes to classify better. Additionally, with our genre predictor, we could build a recommendation model. The recommendation model would use the genre predictor, user ratings, content age ratings, the cast, and other features to recommend TV shows and movies to users.

V. References

[1] Shattuc, J. (2020). Netflix, Inc. and Online Television. A companion to television, 145-164.

[2] Bennett, J., & Lanning, S. (2007, August). The netflix prize. In Proceedings of KDD cup and workshop (Vol. 2007, p. 35).

[3] Kingma, D. P., & Ba, J. L. (2015). Adam: a Method for Stochastic Optimization. International Conference on Learning Representations

[4] Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. Proceedings of the Conference on Empirical Methods in Natural Language Processing, 1746–1751.

[5] Ruder, S., Ghaffari, P., & Breslin, J. G. (2016). A Hierarchical Model of Reviews for Aspect-based Sentiment Analysis. Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP-16), 999–1005.

[6] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. [pdf] [bib]

[7] Melamud, O., McClosky, D., Patwardhan, S., Bansal, M. (2016). The Role of Context Types and Dimensionality in Learning Word Embeddings. In Proceedings of NAACL-HLT 2016 (pp. 1030–1040).

[8] Plank, B., Søgaard, A., Goldberg, Y. (2016). Multilingual Part-of-Speech Tagging with Bidirectional Long Short-Term Memory Models and Auxiliary Loss. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics.