Team NeuralNetflix

Milestone 2 - Data Collection and feature engineering

Note: The code is in a separate .ipynb

Description of our data

Our full dataframe includes

15 features from TMBD:

- id,
- genre_ids,
- poster_path,
- title,
- overview,
- release_date,
- popularity,
- original_title,
- backdrop_path,
- vote_count,
- video,
- adult,
- vote_average,
- original_language
- keywords

5 features from IMDB:

- director
- votes
- certificates
- stunt performer
- special effects department

We have collected 12,015 rows of this data from a few endpoints of the TMDB API and IMDB API.

Here are some summary statistics and descriptions of the columns that matter:

- Popularity: Range from 1.19 to 180.45, with an average of 1.99
- Vote_count: Range from to 0 to 11142, with an average of 269.8
- Video: All false
- Adult: All false
- Vote_average: Range from 0 to 10, with an average of 5.71
- Original_language: ['cn', 'da', 'de', 'el', 'en', 'es', 'fa', 'fr', 'hi', 'hu', 'id', 'it', 'ja', 'ko', 'lt', 'ml', 'nl', 'no', 'pl', 'pt', 'ro', 'ru', 'sr', 'sv', 'ta', 'th', 'tl', 'tr', 'zh']. 'en' is the most common, accounting for about 88% of the data.

- Keywords: There are 12997 unique keywords. The top 5 keywords and their counts are: ['woman director': 649, 'independent film': 471, 'murder': 418, 'based on novel': 362, 'sex': 356]. There are a large majority (54%) of keywords appearing only once. For each movie, there are on average 9 keywords (with a min of 1 and a max of 90).
- director - Name
- votes number of votes
- certificates (UK certificates PG, 12, 15, 18)
- stunt performer - number of people
- special effects department - number of people

Because video and adult are all false, we are not using those columns as predictors.

What does your choice of Y look like?

Y is the genre id. These come from the set: [10769, 10752, 80, 10402, 35, 36, 37, 53, 9648, 12, 10770, 14, 16, 18, 99, 878, 27, 28, 10749, 10751].

10769 does not have a genre name from the API, and we decided to drop it because: 1) There are only 0.9% of the data that uses this genre id, 2) 100% of the movies that have this genre id also have at least one other genre id, hence we don't end up dropping any movies.

Hence, Y will have 19 different categories, and we'll do a multi-label prediction with neutral networks for the next milestone.

Which features do you choose for X and why?

We end up choosing the following features for our baseline:

- Month (broken down from release_date)
- Year (broken down from release_date)
- Popularity
- Vote Count (tmdb)
- Vote Average
- Original_language
- Director
- Votes (imdb)
- Certificate
- Stunt performer
- Special effects department
- Top 2 keywords per genre (we looked at the keyword count per genre, and pick out the top 2 for each genre)

We decided to include the number of stunt performers and number of people in the special effects department as these could be indicators of genres such as 'Action' and 'Adventure'. In addition the movie certificate could be an indicator of genres such as 'Horror' and 'Action'.

Next steps to add more features:

We can improve the way we reduce the dimensionality of the keywords (instead of just choosing top 2 per genre) by doing the following:

- Use all the keywords, then run PCA for each genre

- We can model input as each keyword, and output as genre. Then, we can run a Random Forest classifier on this data from which we can find the most important features (or most important keywords) that are indicative of genre. This is better than just picking the top 2 keywords, because some genres may be best predicted not just by the *presence of predictors* but by also the *absence of predictors*.

We can also use the overview, which is a sentence of text:

- We can tokenize overview, and run sentiment analysis to predict if the overview is positive or negative, and use this as a feature.

We can also create a new predictor based on the cluster group done on all the keywords:

- We run PCA on the all the keywords to reduce dimensionality
- Run multiple types of clustering (PAM, Fuzzy, etc.) and figure the optimal number of clusters using GAP, Elbow plots and Average Silhouette Method
- The new predictor will be the cluster group number that this movie belongs to. By using multiple types of clustering we can get multiple predictors.

Discussion about the imbalanced nature of the data and how you want to address it

The genre 'TV Movie' only appeared 1.5% of the time, while the genre 'Drama' appeared 45.8% of the time. Specifically, these are the number of movies for each genre:

- ‘Action’: 2648,
- ‘Adventure’: 1667,
- ‘Animation’: 865,
- ‘Comedy’: 3675,
- ‘Crime’: 1501,
- ‘Documentary’: 306,
- ‘Drama’: 5505,
- ‘Family’: 1102,
- ‘Fantasy’: 995,
- ‘History’: 453,
- ‘Horror’: 1456,
- ‘Music’: 337,
- ‘Mystery’: 845,
- ‘Romance’: 1991,
- ‘Science Fiction’: 1220,
- ‘TV Movie’: 183,
- ‘Thriller’: 2977,
- ‘War’: 376,
- ‘Western’: 214

Hence the data is imbalanced, and we will oversample the minority genres, and undersample the majority genres. Given that there are 12,015 movies and 19 genres, we could aim for about 632 movies per genre. Hence, we bootstrap with replacement each genre to get 632 movies for that genre, such that we will end up with the approximately the same number of movies (12,008).

How do you sample your data, how many samples, and why?

As stated above, we will use a mixture of oversampling and undersampling on genres to get balanced data on genres. To split our data into train and test, we will use a stratified 60-40 split into train and test.

With this, we can run a multi-layer perceptron on the above features, with a few hidden layers with ReLU activation functions, and a final output layer with 19 output features (one for each genre), and a sigmoid activaton function to get a probability of each genre. We'll use adam, a variant of SGD, and the cross-entropy loss function to train our neural network. We can then use GridSearchCV to tune the hyperparameters: batch_size, epochs, number of neurons in the hidden layers, etc.