Movie Recommendation

Search

Term Frequency

Term Frequency of term t for document d is the number of term frequency in the documents.

Inverse Document Frequency

Inverse Document Frequency is defined by following equation where N is the total documents having term t and df is the term frequency for document d.

Term Frequency- Inverted Document Frequency

Term Frequency and Inverted Document Frequency is combined to produce composite weight for each term and in each document using following equation.

Stemming

Stemming usually refers to a crude heuristic process that chops off the ends of words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes.

Similarity

To find similarity between search query and movies first I have generated word vector for each movie and using word vectors inverted term frequency was calculated.

Now for new query we need to generate word vector of query and then calculate inverted term frequency. After that we need to calculate cosine similarity between inverted term frequency of query and of each movie. Then we will find top results whose cosine similarity is maximum.

Classification

Multinomial Naive Bayes Classifier

Naive Bayes classification works on probability of individual features within document. For movie classification movie description is data and each word or term in description is feature.

Formula

t1, t2, t3, ... = Terms in data

$P(Genre|[t1, t2, t3, ...]) = P(Genre) * P(t1|Genre) * P(t2|Genre) * P(t3|Genre) * ...$

Genres

- Comedy
- Action
- Animation
- Romance

- Adventure
- Horror

Multilabel Classification

Movies can belongs to many classes such as an animation movie can be comedy as well. Multilabel classification is little bit tricky. The model should return multiple genres. Following are the few techniques I have used.

Top Results

In top result technique it returns top n genres which has highest probability.

Pros: Easy and fast.

Cons: In most cases movies does not belongs to exactly n genres, it might belongs to less than or greater that n genres.

Threshold

In threshold it returns genres which has probability greater than threshold. We can find good threshold value by applying and evaluating different values.

Cons: In naive bayes with text the range of probability changes with the number of terms in document. For example for one test case probability for genres might between 1.0e-1 to 1.0e-5 and for other it might be between 1.0e-10 to 1.0e-20 so there might be cases when it classify to all genres and their might be case where it does not classify to any genre as well.

Mean

In this technique it returns the geners which has higher probability than other genres. For movie it first calculate probability using naive bayes theorem than it perform log and find mean of each genres, then it will return genres which has higher value than mean. In addition it has one variable which is multiplied by mean while comparing with each genre's value and we can find good value by applying different values in this case we can use F1 Score and evaluate on validation data.

Challenges

Accuracy

Improving accuracy was major challenge for this project. Following are the techniques used to improve accuracy.

Stopwords

Stop words are the words which does not play significant role in solving problem such as "the", "a", "this", etc. These words does not give any clue about genre and also these words repeats very much and that creates problem in classification.

Accuracy with stopwords: 0.5024773649566582

Accuracy without stopwords: 0.5178779592087157

Lemmatization

Lemmatization is a process of finding root word for example "having" word is converted into "have".

Accuracy after applying lemmatization: 0.527629565354579

Contribution

Multinomial Naive Bayes
Implemented multinomial naive bayes algorithm from scratch and tried to optimize as much as possible for quick response.

Multi Label Classification
Tried different techniques to get multiple genres as a result such as top k, threshold, and created new method for retrieving multiple genres.

Recommendation

Item Based Collaborative Filtering Recommender
In item based collaborative filtering similarity between item's are used to recommend items to users. In this method if user is looking for item or has liked any item than similar items are recommended to that uers. We use cosine similarity between items to find similarity.

Tf-idf
In this movie recommendation it uses tf-idf to find similarity between movies. Cosine similarity between user's selected movie to all other movies are calculted and top 20 results are shown to the users.

Challenges

Time Consuming

To calculate cosine similarity of a movie to all movies for every new movie selection is very inefficient so instead of calculating cosine similarity at runtime we can calculate and cosine similarity of all movies to all movies before hosting and can use that matrix at runtime that will not take too much time.