Group 8: Project Movies

Abstract

This Recommendation system is nothing but a subclass of information retrieval and machine learning. In today’s world recommendation system plays an important role not only in video streaming platforms but in all the e-commerce industry such as recommending books, online products, songs, podcasts etc. it is prevalent in today’s world. The information overload problem is solved by todays search engines, but they do not provide us with the personalized information. This limitation is solved by recommendation system by providing proper and personalized information. There is a lot of research going on these systems. In our project we have implemented mainly 3 recommendation systems content based, collaborative and hybrid recommendation system. Google’s recommendation system is a content-based recommendation Netflix uses collaborative filtering. In addition, as we also preformed text classification for the genre in the movies. Since we also performed regression to for predicting the movie revenues. We have for almost 45000 plus movies in which there were 26 million plus ratings given by 27000 plus users.

1 Introduction:

Movies are the major source of entertainment and have been entertaining us since decades. Before online service providers we used to buy movies or rent movies from the nearby stores and watch them. After the Introduction of online media service providers such as Netflix, Amazon Prime etc. there has been a lot of growth in its subscriptions and also, we see almost everyone has a subscription to it. For these media service providers, it is not just streaming of movies, but they need to understand what the subscribes need and what their interests are which is a challenge for these online platforms. This is the reason why we need a recommendation system and these platforms are investing a lot of time in customizing and building recommendation systems. Our Project is mainly to build a personalized movie recommendation system for today’s movie lovers.

2 Research Questions:

In our Project, we are trying to address the following research questions:

1. Given the overview of a movie, we will predict the genre of the movie – Text Classification.
2. Movie recommendation system based on content based filtering and collaborative filtering with single value decomposition – Content and Collaborative Recommendation
3. Hybrid movie recommendation system combining the content based and collaborative recommendation systems – Hybrid Recommendation
4. Given the parameters like budget, revenue, language, average ratings etc. we will predict the popularity score assigned for that movie by TMDB – Regression.
5. K-Means Clustering Analysis

2.1 Genres Prediction using Text Classification:

In this research question, we aim to predict the genre of a particular movie given the overview (description) of the movie. Here we consider “movies\_metadata” of the dataset.

Data Preprocessing:

* We are considering the columns “genres” and “overview” for text classification. We observed that the genres column was in json format with other information and we converted it list format containing only genres.
* The overview column was converted to string format and then cleaned to contain only words by eliminating unwanted characters such as punctuations etc.
* In order to analyze the words in the overview column, we plotted frequency of each words as follows:

A screenshot of a cell phone

Description automatically generated

* In the above plot, we observed that the most frequent words in the overview column are insignificant words to our text classification model. These words such as “the, a, and” etc. are called Stop words, which needs to be eliminated for better performance of our model.
* We removed these stop words from the “overview” column using nltk library.
* The words contain inflections such as prefix, suffix to make the word grammatically correct but these words are redundant to our model. In order to extract the stem words, we perform stemming. For example, the words such as “cook, cooking, cooked” needs to be reduced to the stem word “cook”. We perform the stemming process using nltk library.
* Stemming removes the suffix of the word without much consideration to its meaning, leading to either under stemming or over stemming. In order to overcome this drawback, we need to perform Lemmatization. It resolves the word to its dictionary form (lemma), thus preserving the meaning of the word and also removing redundant words. Hence, we perform lemmatization.

Feature Engineering:

* The genres column is multivalued, hence there are multiple classes for text classification. To encode multiple classes of genres column we have used MultiLabelBinarizer.
* Text classification cannot be performed on raw text data; hence we need to convert it into vectors of numbers. To achieve this, we use Term Frequency- Inverse Document Frequency (TF-IDF) vectorizer.
* TF-IDF vectorizer accounts for the weightage of the words occurring in the overview column by providing a higher weightage for the words which occurs more in that particular document but does not occur frequently in other documents.
* It also accounts for the ordering of the words by specifying the n-grams in TF-IDF vectorizer. We ignore the high frequency words by specifying max\_df parameter and ignore the low frequency words by specifying min\_df parameter of TF-IDF and consider only the medium frequency words for our model.

Model Building and Evaluation:

* We split the data into test and train dataset with the input feature as overview and the target as genres.
* We transform the input feature “overview” using TF-IDF vectorizer.
* Now we build models with One Vs Rest classifier, which involves training a single classifier per class and treating samples of that class as positive and samples from all other classes are treated as negative.
* We build our first model with One Vs Rest classifier using naive\_bayes classifier and performed hyper parameter tuning using GridSearchCV to obtain the best hyper parameters for our model and evaluated our model. We obtained a F1 score around 0.36.
* We then built a model with One Vs Rest classifier using LinearSVC classifier and performed hyper parameter tuning using GridSearchCV to obtain the best hyper parameters for our model and evaluated our model. We obtained a F1 score around 0.5153.
* We have tried one more approach with pipeline using CountVectorizer and TfidfTransformer with OneVsRestClassifier using LinearSVC and obtained a F1 score around 0.528.
* Finally, we built a model with One Vs Rest classifier using Logistic regression with a default threshold of 0.5 and obtained the best hyper parameters using GridSearchCV and achieved a F1 score around 0.438.
* In the above model, we changed the threshold of logistic regression to 0.3 and achieved a F1 score around 0.574, which is a good F1 score for multi class text classification.

Result:

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

The above snapshot shows the detailed precision and recall as well as f1 scores to few of the samples within the dataset. The final f1 score achieved by the model is also depicted.

2.2 Recommendation system with content-based and collaborative filtering:

2.2.1 Content based filtering:

This can simply be defined as a system which recommends you content which has similar characteristics or have similar key words or meaning in them content-based filtering. Basically, this system suggests similar items to the items which user has liked. The major advantages which we have using this recommendation system is that the recommendations tend to be transparent and if there are any new items added into the data and are similar by some keyword these new items can be easily recommended. This is also much easier to implement than the other recommendation systems.

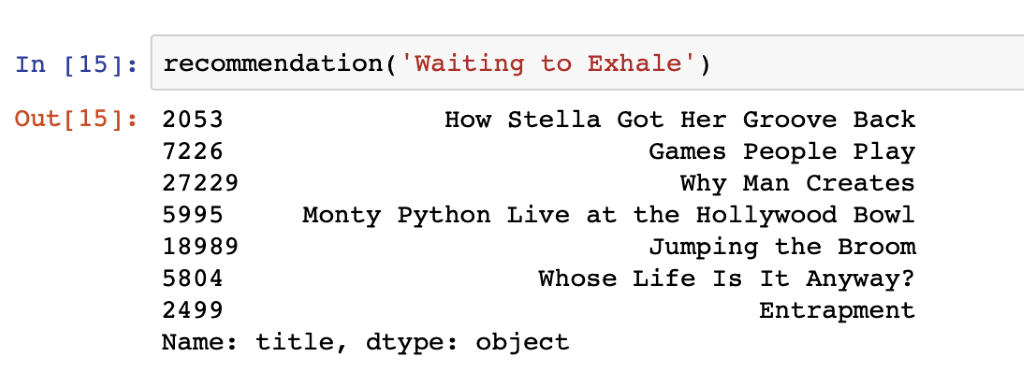
Implementation:

To implement this system, we have used our movies\_metadata.csv data set. After cleaning the data, Extracting data from json format etc. We have considered 5 columns which are overview, title, cast, crew, genres. The process is as follows: -

* Firstly, we remove stop words from each and every sentence. A,an,the,or etc are mostly the stopwords.These words wouldn’t give us much information and increase the complexity and the noise of or system so these words must be removed
* Now with words considered to be important in our sentences we take them and form tf-idf matrix. Tf matrix is nothing but the number of times words appear in the document divided by total number of terms in the document. The idf is calculated by considering the log of total number of documents divided by number of times word appears in the document.
* Now we find the cosine similarity for all the documents and ones with the higher similarity value would be taken as a group. The main reason we use cosine similarity is because when we are modeling texts, we will have many dimensions in thousands taking Euclidean for all of them is a bad idea, so we used cosine similarity.
* The documents with highest similarity value will be considered as a group which is basically the output of our system.

Results:

Our input to the system would be a movie name and the output of the system would be the movies which are similar to the movie we entered.

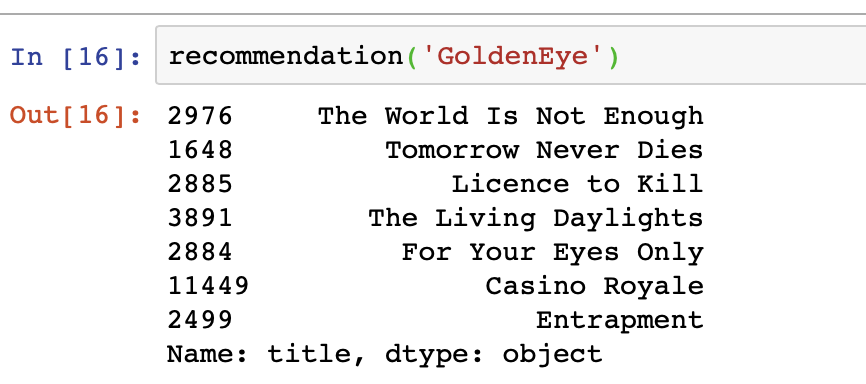


Our input was a movie named “Waiting to exhale” and the output we got are the movies which are similar to it. So, this is how content based filtering works and gives us results.

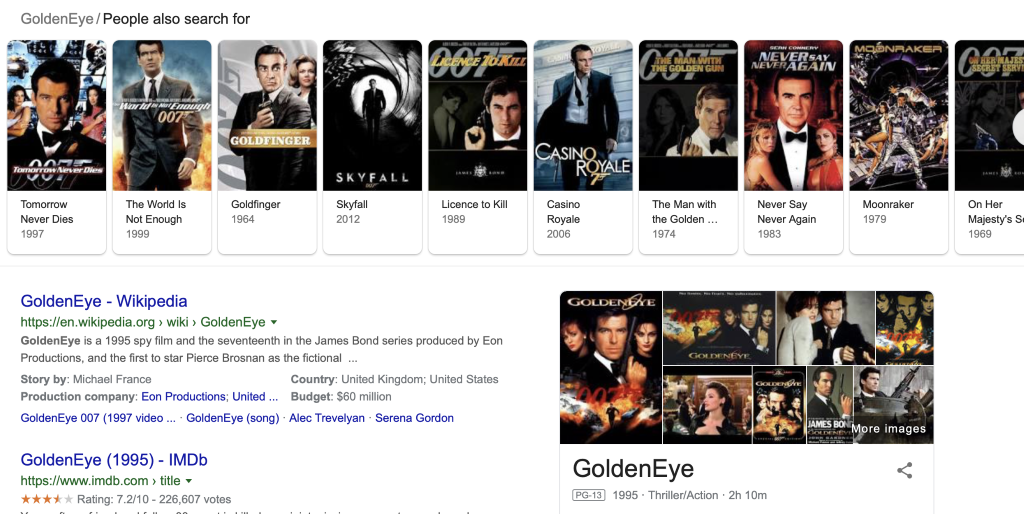
Evaluation:

To evaluate our system, we compared our recommendation system to Google’s recommendation system. Here is one of the results

Our system’s result for a movie with title “Goldeneye”:



Google’s recommendation for the movie with title “GoldenEye”:



We can see that our system is more or less on the same lines of googles recommendation system.

Disadvantage:

The major disadvantage of this system is that it doesn’t consider the user’s taste and give recommendations across genres. The system should also consider the user interests and personal taste of the user which content-based recommender fails to do.

2.2.2 Collaborative filtering with single value decomposition:

To overcome the disadvantages of content-based filtering they came up with collaborative filtering. Many of the movie streaming companies such as Netflix is currently using collaborative filtering. This system matches people with similar interests and give recommendations based on users’ personal interests and tastes. There are 2 types collaborative filtering techniques:

1. Memory Based Collaborative Filtering
2. Model Based Collaborative Filtering

Memory Based Collaborative filtering:

Memory Based Collaborative Filtering consists of User-Based and Item-Based Collaborative Filtering:

User based filtering basically matches and finds similarity patterns between all the users if it finds things similar to what one user liked he would recommend it to the other user of similar interests. Whereas item-based filtering finds similarity patterns between items and recommends them to the users based on the computed information.

Both User based and item based are very popular and widely used but each one has its own advantages and disadvantages. So, to overcome this we have used more popular approach which is using the SVD technique (which belongs to Model Based Filtering).

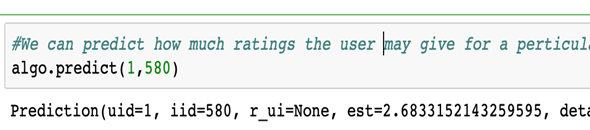
Implementation:

For the implementation of item based collaborative filtering we have used rating.csv dataset which includes UserId, MovieId and the rating given to that movie by the user. The ratings in this dataset are all on the scale of 5 points. The following steps were followed for the implementation of collaborative filtering: -

* we now performed 4-fold cross validation and measured the Rmse and Mac values these values will help us in evaluating our model. The minimum the RMSE the better is our model.
* Now from surprise library we bring in SVD algorithm. SVD is nothing but a matrix factorization technique which is generally used to reduce the number of features of the dataset by reducing the space dimension doing all this it also captures the similarity between the users and ratings.
* We have optimized our algorithm by hyper tuning the parameters which is nothing, but we found out the best set of parameters for the model and then based on these models we trained our model.
* We then trained and fitted the data with the model we initialized.

Results:

Here our input is UserID and the movieID for which we predict the ratings which the user would give. Here is one of the examples,



Here our input is the User ID=1 and the movieId for which user would give rating for

Here rating = 2.68 which means user ID 1 will give 2.68 rating for movie ID 580

2.3 Hybrid Recommendation System:

Content based recommendation recommends movies based on the similarities between the movies but does not take into account the user preferences. To overcome this drawback collaborative approach takes into consideration the user ratings for recommendations. Further, there are two approaches in collaborative recommendation; one accounts for similarities between users (user based collaborative filtering) for recommendation and the other accounts for similarities between movies (item based collaborative filtering) for recommendation. The drawback of user based collaborative filtering is that the user preferences may change over time while the item based collaborative filtering suffers from scalability issue when there are large number of movies and users. Collaborative filtering with single value decomposition captures the similarities between the users and the items and determines how a user might rate a particular movie.

In our project, we have implemented a hybrid recommendation system which uses both content-based and collaborative recommendation with single value decomposition for recommendation. This approach overcomes the drawbacks associated with both contend based recommendation and collaborative recommendation.

Implementation:

* We take the userId and the movie title, to recommend movies for a particular user.
* First, given a movie title, we find a set of similar movies by content-based recommendation.
* Then, we identify the set of movies which have not been rated by our current user.
* From this movie set, we identify unique movies which have not been rated by our current user.
* For each movie in this unique set, we get the corresponding movieId.
* By using this movieId and the userId of our current user, we predict the estimated ratings of our current user for each movie in this unique set of movies using collaborative filtering with single value decomposition.
* Then, we sort the estimated ratings of our current user obtained from collaborative filtering with single value decomposition to obtain the higher priority movies which is the result of our recommendation system.

Our hybrid recommendation system thus takes into account the similarities between movies as well as the preferences of our current user by considering movies which the current user might like the most.

Result:

For a user with userId 1 and movie with title ‘GoldenEye’ our recommendation system might will recommend the following movies along with their predicted ratings by the current user:

A screenshot of a cell phone

Description automatically generated

2.4 Popularity prediction with Linear Regression:

Linear Regression is one of the popular data modelling techniques. It basically means to find a relationship between more than one independent variable and one dependent variable. The advantage of linear regression is that it works great when there is a linear relationship between the dependent and the independent variables. The disadvantage of Linear Regression is that it forms many assumptions with the independent variables such as normality, no correlation etc.

In Movies Data set, we are trying to predict the popularity of the movies. We are considering various attributes such as Budget, Revenue, runtime of the movie etc. to predict the popularity.

Before training any model, the data has to be preprocessed. It involves data cleaning, feature selection, removing outliers etc. The process we followed in our Movies Dataset before applying Linear Regression can be summarized as follows.

1. Removing Features by intuition: Sometimes there isn’t a necessary of feature selection algorithms or other techniques for removing the features. We can do it by our common sense. We remove various features such as adult, genre, homepage, original\_tiltle etc.
2. Out of the remaining features, such as budget, revenue, runtime, many had null values which were partially removes and the rest was partially imputed with the median
3. Later the outliers in the various features was removed and non-numeric values were converted to numeric, Normalizing the features, etc
4. This final data was trained on Linear Regression and final popularity value was predicted

Result:

A screenshot of a cell phone

Description automatically generated

2.5 K-means Clustering Analysis:

K-means clustering is one of the popular Unsupervised algorithms. K means algorithm tries to find a cluster among the feature (or features). The number of clusters to be formed is determined by the value of K. Usually a technique called elbow method is employed to find the optimal value of K.

We tried to implement a cluster-based movie recommendation system. We took a subset of data that contains only the genres as romance and comedy. We took the users who rated both the type of genres. We took the Average ratings of them and then used K means algorithm to separate the set of users who like only romance movies and who like only comedy movies. We further wanted to increase the number of clusters and form a recommendation system with it but we couldn’t achieve satisfactory results due to which we had to stop at the stage after the clusters were formed

The results of the analysis are depicted in the below figures

A picture containing screenshot

Description automatically generated

A screenshot of a cell phone

Description automatically generated

Cluster differentiating the users who like romance and comedy genre type movies

Conclusion:

In the first research question, we have predicted the genre of movies given their overview using text classification and achieved an F1 score of around 57.49 which is a good F1 score for multi class text classification. In the second research question, we have built a content-based recommendation system which gave the recommendation in line with Google’s similar movie suggestion. Also, we have built a single value decomposition based collaborative filtering which will efficiently predict the possible ratings of the given user for the movies the user hasn’t rated. Finally, we have built a hybrid recommender, which combines the advantages of content-based and collaborative filtering and recommends similar movies to the given user based on movies that the user could like. We have also predicted the popularity score assigned to a given movie by TMDB and achieved a RMSE score of around 4.26.