**MOVIE RECOMMENDATION SYSTEM USING CONTENT BASED FILTERING ALGORITHM**

*Submitted for*

**Machine Learning Techniques (CS 7602)**

*Prepared by*

**S. Harish** (2015103553)

Department of Computer Science and Engineering

College of Engineering Guindy

Anna University

Chennai 600 025

**ABSTRACT**

Given the huge amount of movies are available all over the world, it is challenging for a user to find the appropriate movies suitable for his/her tastes. Different users like different movies or actors. It is important to find a method of filtering irrelevant movies and/or find a set of relevant movies*.* Movie recommendation system is a process of exactly doing above tasks. Such a system has lot of implications and is inspired by the success of recommendation systems in different domains such as books, TV program, jokes, news articles. It is one of the most important research in the digital television domain. The most well known recommendation systems are mainly based on Collaborative Filtering (CF) and Content-based Filtering. CF first tries to find out the groups of similar users automatically from a set of active users. The similarities between users are computed using correlation measure. It then recommends items to a user based on the opinions of the users groups. On the other hand, Content-based Filtering works such that only Single User are considered for Recommendation as opposed to Multiple Users in case of CF. It works by computing cosine similarity between the current movie that user has watched and the rest of list of movies that are not watched by the user. In this project, I have implemented the Content based Filtering algorithm for Movie Recommendation System.

**OVERALL ARCHITECTURE:**

DATA

CLEANING

FEATURE ENGINEERING

FEATURE EXTRACTION

COSINE

SIMILARITY

SORTING SIMILARITY

**FIGURE 1:** OVERALL ARCHTIECTURE DIAGRAM

**MODULE DESCRIPTION:**

The different modules involved in the Movie Recommendation System using Content based Filtering are namely Data Cleaning, Feature Engineering, Feature Extraction, Cosine Similarity and Sorting Similarity. The following section details about each of these modules.

**DATA CLEANING:**

Data Cleaning is the process of cleaning the data in appropriate form. This step is essential in sense that whatever the garbage in is garbage out. If the algorithm tries to predict on unclean data, then it definitely fails to produce correct results. This step also affects the accuracy of the system. We replace the Missing Values (N/A) with an Empty String for better processing.

**FEATURE ENGINEERING:**

Feature Engineering is the process of selecting the features that best helps to produce the target values. For example, consider you have 10 attributes and 1 target. We may use all 10 attributes to modeling, but in reality only 5 attributes may be enough for entire modeling. Using all attributes provides suboptimal results leading to inefficient system. In our system, we have features namely index, budget, genres, homepage, id, keywords, original\_language, original\_title, overview, popularity, production\_companies, production\_countries, release\_date, revenue, runtime, spoken\_languages, status, tagline, title, vote\_average, vote\_count, cast, crew and director. From these many features, we focus only on Index, Keywords, Cast, Genres and Director and leaving out the rest. We consider these features because they help more towards finding the similarity between the current movie and “to watch” movies.

**FEATURE EXTRACTION:**

Feature Extraction is the process of extracting features from the dataset. In our project, the input is of the form text. It is difficult to work with text in Machine learning. So, we need to convert it to a form that can be used suitably for the algorithm. This is achieved by creating the Document Term Matrix representing each word with its index and score value of how important that word is contributing towards the target. In sklearn, this is performed using CountVectorizer module. The fit\_transform method is used on the combined features from the previous step, Feature Engineering.

**COSINE SIMILARITY:**

The similarity between two points in two dimensional space, for example, can be computed using the Cosine Angle formula. This same idea is applied in our system by calculating the cosine value between the movie that user had watched and the movies that user would watch after watching the current movie. This is performed using sklearn’s, cosine\_similarity method. This method takes as input the Count Matrix from previous module and spits out a list of tuples containing Movie Index and Cosine Score of that movie with the input movie.

**SORTING SIMILARITY:**

After finding the similarity score, we need a way to display those movies with highest score. Note, that cosine similarity value of input movie with itself (present in dataset) has maximum value of 1.00, so we need to ensure to drop this value from the list after sorting the list in descending order of the Cosine Values from the list of tuples obtained through the previous step. Finally, the top ‘n’ movies that user might watch are displayed in order of highest likelihood.

**SCREENSHOTS:**

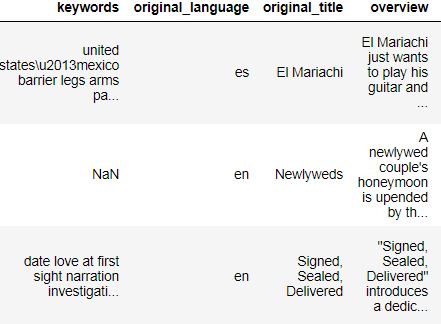
****

Figure 2 Data with Missing Values

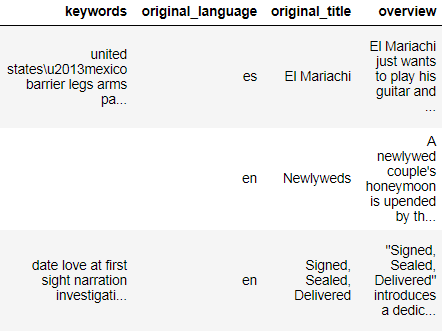


Figure 3 Data with Missing Values Replaced

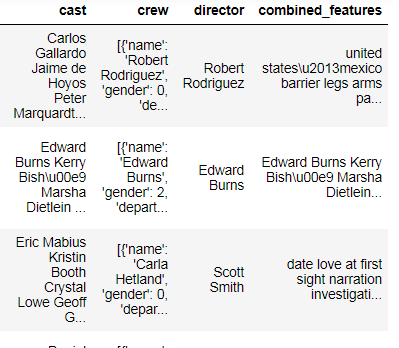
****

Figure 4 Feature Engineering

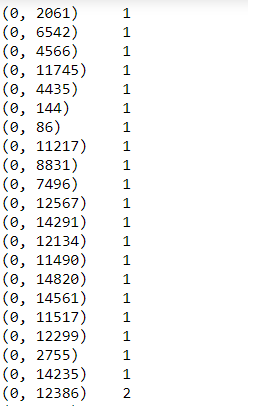


Figure 5 Count Vectorization

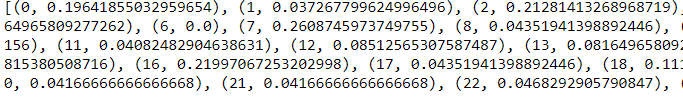
****

Figure 6 Cosine Score List

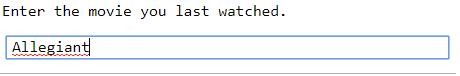


Figure 7 Input Movie

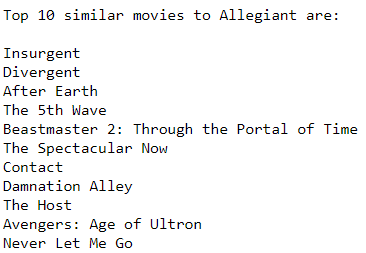


Figure 8 Similar Movies that user can watch

**CONCLUSION:**

Thus, we have finally been able to develop a Movie Recommendation System using Content based Filtering Algorithm that uses Cosine Similarity between One versus Rest of All movies. This system can be further developed to large scale with more dataset and other advanced algorithms.