SCRAPING IMDB MOVIES REVIEW DATA & PERFORMING KEYWORD ANALYSIS

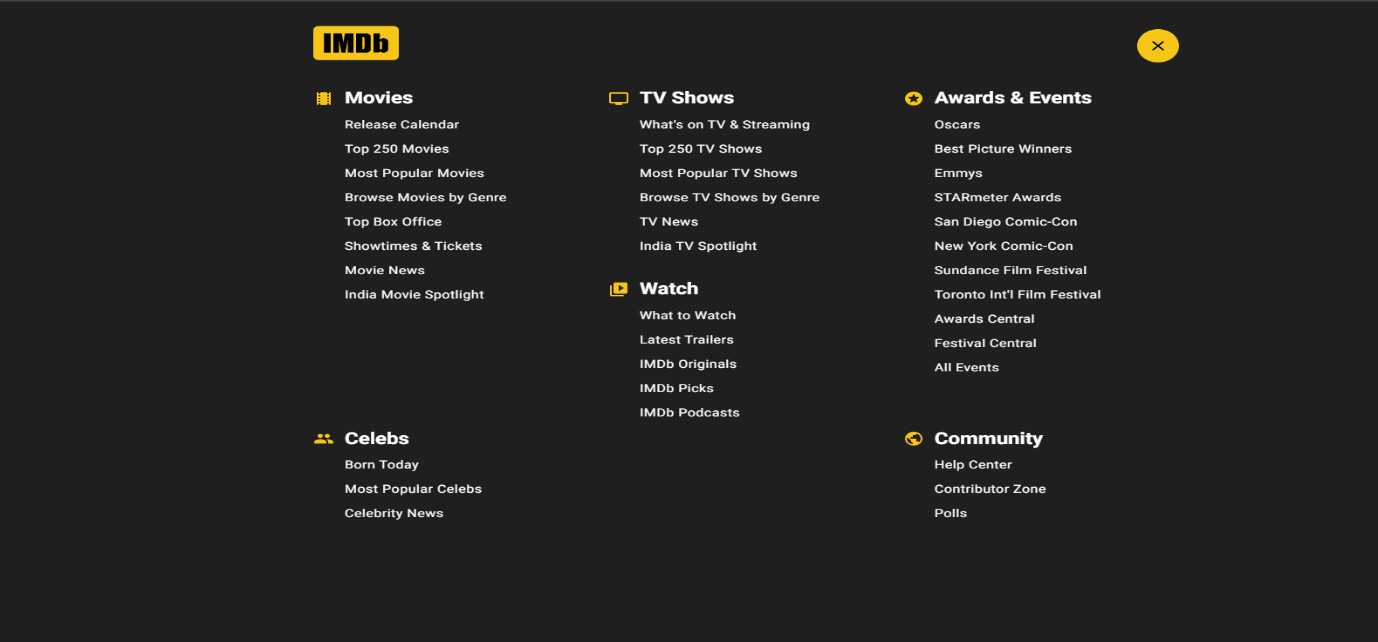
**SUBMITTED BY**

**Sudhanshu Shekhar Lal**

**Abstract.**

IMDB is a popular movie-related search engine. It provides a sizable database of movies, actors, and directors, as well as details on each movie and reviews. Because of its popularity, people use it to find out more about their favourite actors and films. The IMDB website, however, is challenging to use and doesn't always provide the most accurate information about a particular actor or film. If one wants to build a recommendation engine for movies that suggests movies based on user preferences, one needs data sets of various movies and TV shows from various genres. It is possible for programs to automatically extract all of this data from IMDB and then use models to analyze it. This makes it possible to get the results you want.

Keywords: Sentiment analysis, Keyword Extraction, LDA



**Fig:- IMDB Search Menu**

**I. INTRODUCTION**

The proposed idea is to scrape data of any 20 Movies listed on IMDB site and based on user/PVR Owner desire to select a movie for it’s theatre based on Keyword extraction and sentiment analysis of various movies parameters like ratings, votes, genres, WOM(Word of Mouth of reviewers, gross revenues and other search results on Google Trends and Google Ads.

Assuming that Reviewer’s WOM has impacts on Movies based on the research conducted by a small community stating that Various external online data sources that include: Google Trends, Wikipedia, Google Search, Technical indicators, Macroeconomic indicators and the models using the features from these external sources along with the traditional data about the movie performance will help Theatre owners to have a Movie recommendation for their theatres for making better profits

# DATA COLLECTION AND PREPROCESSING

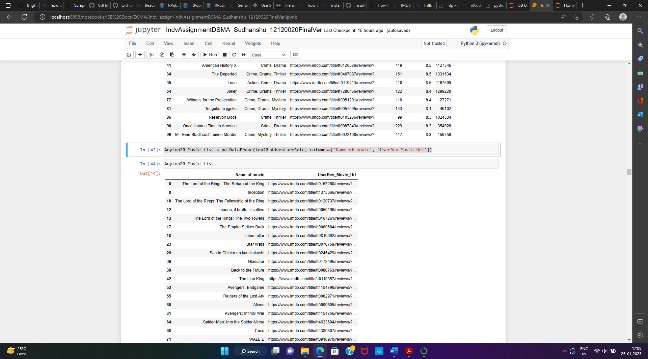
The Data Collection is based on Scraping Movie Details and Reviews using BeautifulSoup and Selenium Web Driver.

Web Scraping using BeautifulSoup is done for scraping List of Top 100 Movies from IMDB and their following details:-

* 1. Name of the movie
  2. Genre of the movie
  3. Link URL of the movie
  4. User Review Movie URL
  5. Year of release
  6. Watch time
  7. Movie rating
  8. Meta score
  9. Votes

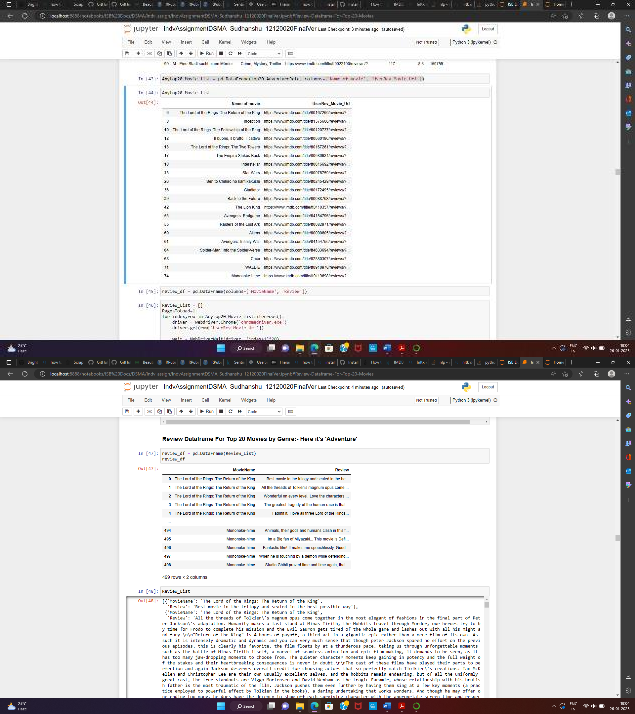
Next step is pre-processing the data scraped and performing some EDA, which helped finding out how many Genres are there in the Dataset, from there I selected **top20** movies of a particular Genre so that I can fulfil the requirement as mentioned. Now using **Selenium Web Driver** I scraped the Name of Movie and Reviews by iterating over the User Review Movie URL .

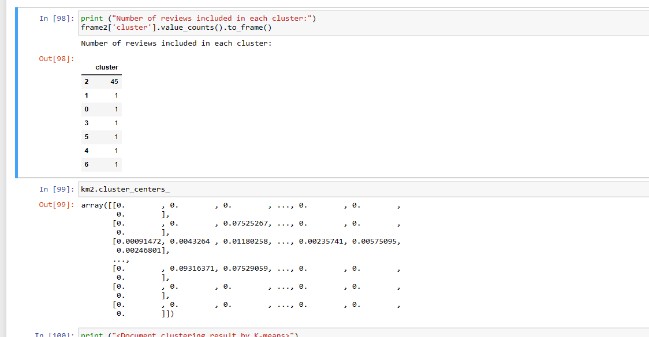
The Movie Genre Chosen :- Adventure movies sorted by Rating (Zoom pic to get clear view)



# Data Cleaning and exploration

Based on rating and genre I extracted the movie name and Review URL link for the respective movies and performed a Review extraction on the Movies.



After we got all the reviews for all the movies, proceeded towards selecting the reviews of a particular movie based on the user input then started cleaning the Review data by removing stopwords and other symbols and tags, the final “Cleaned” reviews I got for the further processes.

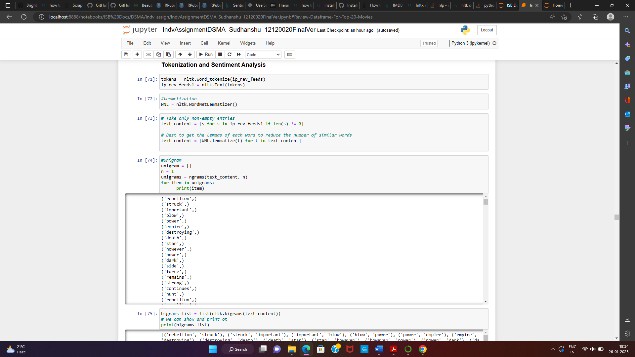
# KEYWORD EXTRACTION & ATTEMPT TO ACHEVE HIGH DIVERSITY SCORE

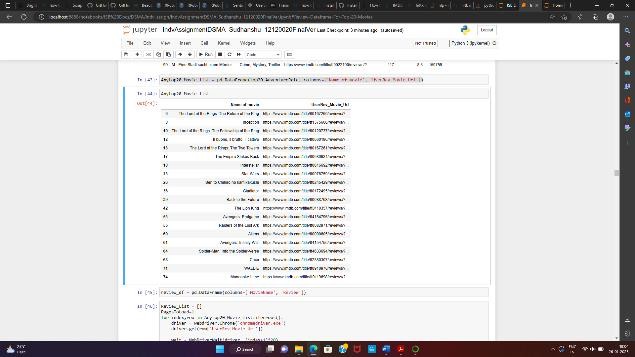
Used KeyBERT’s Maximal Marginal Relevance (MMR), for initial level Keyword Extraction with High Diversity Threshold, this method extracts the most relevant keywords that are the least similar to each other, it iteratively selects new candidates that are both similar to the document and not similar to the already selected keywords, one can choose a low/high-diversity threshold

# 2. SENTIMENTAL ANALYSIS

Sentimental analysis is the heart of this project. We are going to use python ML libraries to extract the features from the reviews collected. As a part of analysis, K means clustering was the first step performed which provided an information of how reviews were clustered. We are going to use Vader algorithm for extracting the feature. Vader algorithm is the best choice for this project since it is humanly automated so also contains some abbreviations and slang words which are recurring in the reviews/comments.

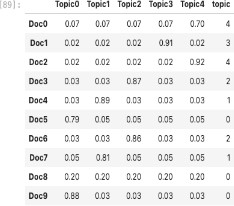
Extracted the feature and classified the emotions into 4 i.e. Positive, negative, neutral and compound. We will store the value for each review with the extracted features.



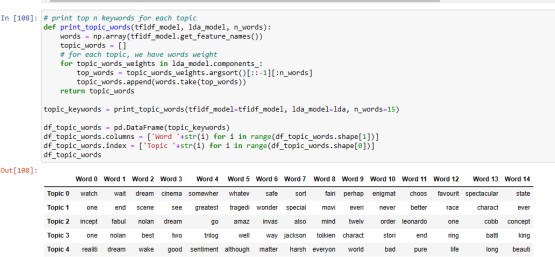


## 4. LDA

LDA, is used for classifying text in a document, and is basically a topic model. It finds certain natural groups of objects (topics), i.e; it builds a topic per document model and word per topic model, modeled as Dirichlet distribution. Here we used LDA for text classification on reviews.



The table below shows the words in the different themes that have been determined. Each row of the table represents a different topic, and each column represents a unique word from the corpus. Each cell (row) contains the likelihood that the word (column) belongs to the topic.



Vader Sentiment Analyzer is used for performing sentiment analysis and Vader score has been used to classify words as either positive, negative, or neutral sentiments. This will allow Theater owner to gain an insight of the audience sentiment towards a particular movie.

Sentiments count along with Sentiment analysis as calculated using Vader Lexicon has been displayed in the bar graph below.



The process of analysing the keywords or search phrases that bring visitors to a website via organic and paid search is known as keyword analysis. As a result, keyword research is the starting point and foundation of search marketing campaigns.

