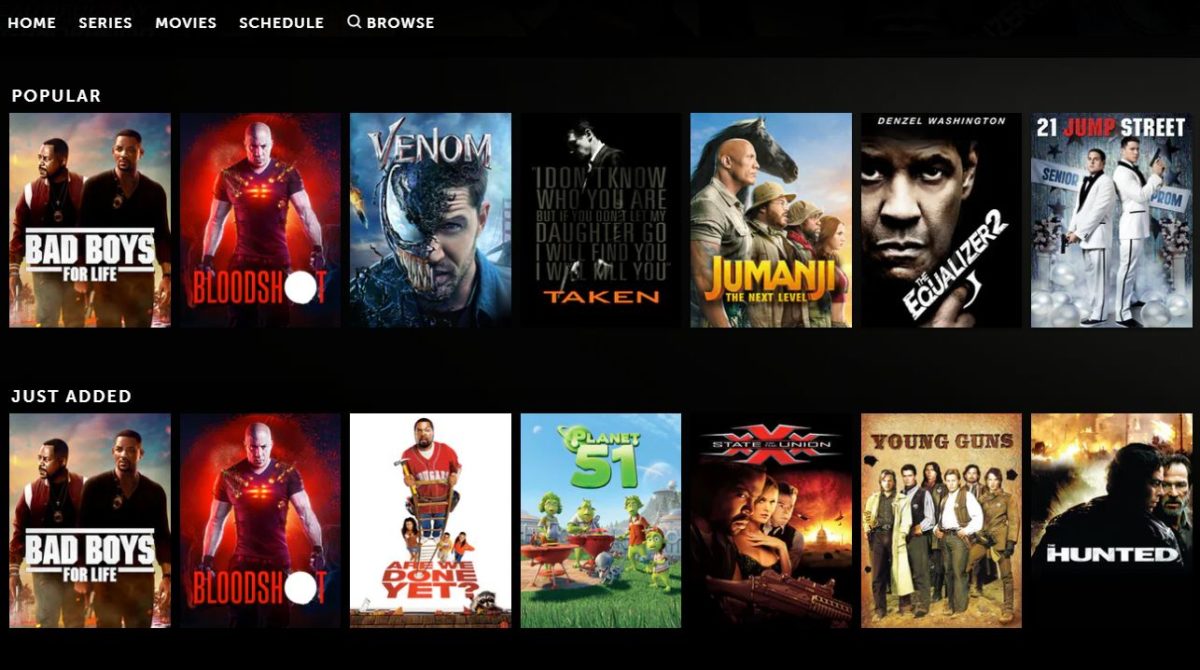
**Movie Recommendation System**



A **Movie Recommendation System**, or a movie recommender system, is an ML-based approach to filtering or predicting the users’ film preferences based on their past choices and behavior. It’s an advanced filtration mechanism that predicts the possible movie choices of the concerned user and their preferences towards a domain-specific item, aka movie.

We at Label Your Data have gathered the most up-to-date information about modern movie recommendation systems and how to build them using different ML solutions. We’ve also touched upon some of the most popular examples of these systems that help many movie fans today stay up to date with all the new releases as well as classics of the cinematography.

# **Types of Recommendation Strategies:**

* Content Based System – It focus on properties of the items.

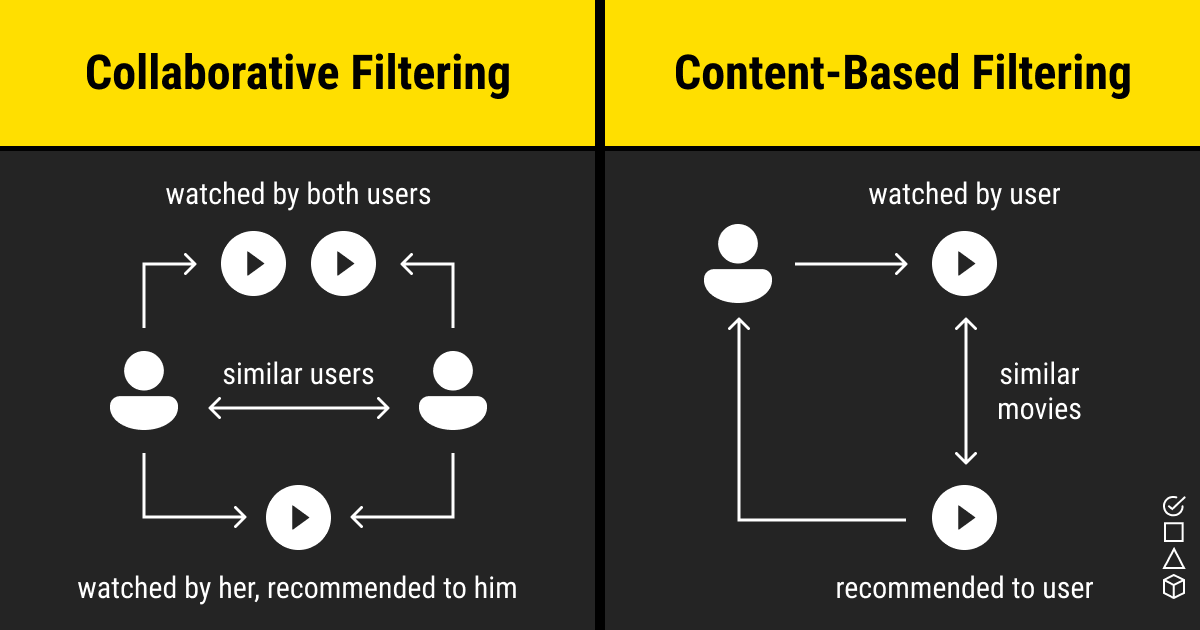
Ex:

Netflix , Amazon etc.

* Collaborative Filtering System – It focus on relationship between users and item

Ex:

FaceBook, Spotify etc.



INTRODUCTION:

Here my project deals with the recommendation of movies to the user based on the user input.

The main goal of this movie recommendation systems is to filter and predict only those movies that a corresponding user is most likely to want to watch.

Here I am explaining the steps involved to make a content based movie recommender system …

* IMPORTING THE DATASET…
* UNDERSTANDING THE DATA SET…
* DATA PREPROCESSING…
* CONVERTING TEXT DATA INTO FEATURE VECTORS…
* CHOOSING THE ALGORITHM TO PREDICT AND RECOMMEND …
* GETTING LIST OF RECOMMENDED MOVIES TO THE USER…

IMPORTING DATASET

Here I am extracting the movies csv file from the Kaggle website that contains list of movies along with casting, directors and others features.

The required libraries have been imported before importing this file

Our dataset contains 4803 rows and 24 columns

UNDERSTANDING THE DATASET

I am selecting the only few relevant features from the dataset

for recommendation as follows below:

* Genre
* Keywords
* Tagline
* Cast
* Director

Based upon this above features we find a similarity between them and build a recommender

DATA PREPROCESSING

Data preprocessing involves handling missing values and cleaning the dataset…

Using movies\_data.isnull().sum() we can find the null values and then we are able to clean them…



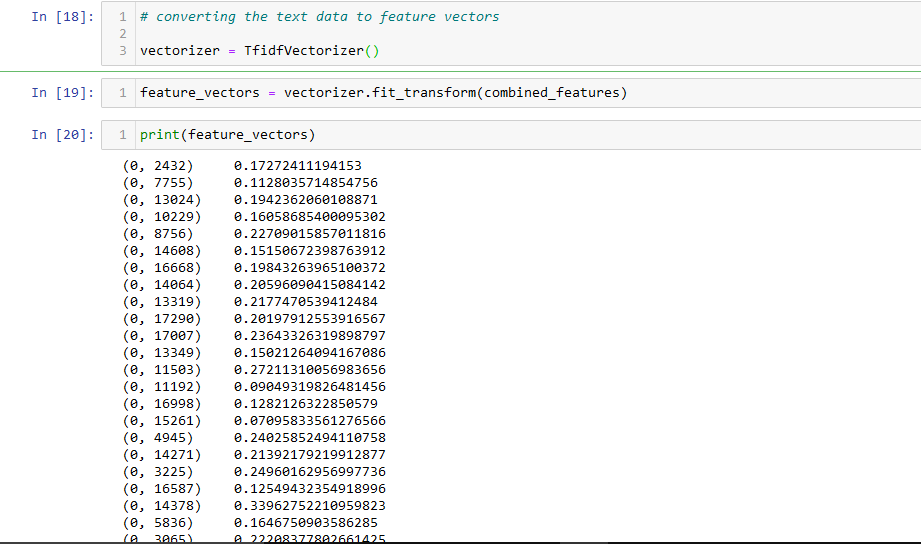
Here I am replacing the null values i.e.Nan with null strings for

the selected features to predict accurately recommender

CONVERTING TEXT DATA INTO FEATURE VECTORS

Since the selected features contains text data it is not possible to apply the algorithm

Then using TfidfVectorizer module the text data is converted into features vectors

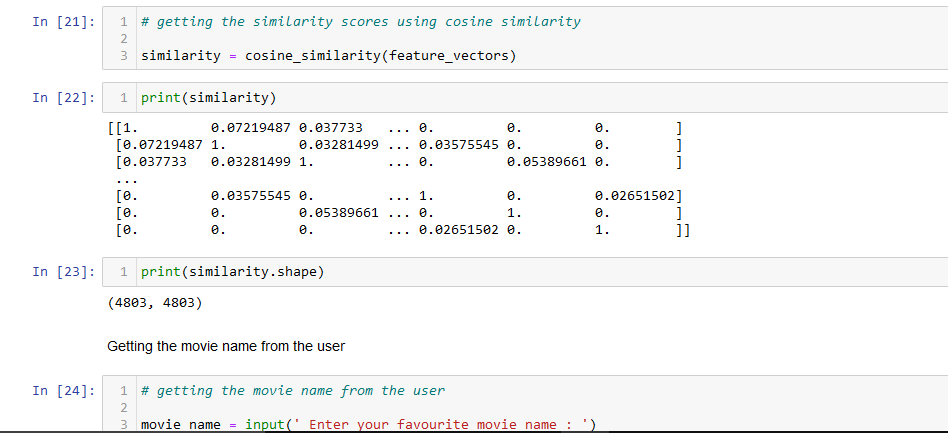


APPLYING COSINE SIMILARITY ALGORITHM

We will use the Cosine Similarity from Sklearn, as the metric to compute the similarity between two movies.

Cosine similarity is a metric used to measure how similar two items are. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space. The output value ranges from 0–1.

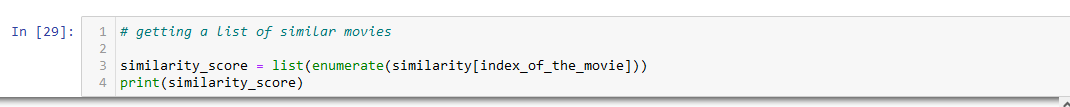
0 means no similarity, whereas 1 means that both the items are 100% similar.



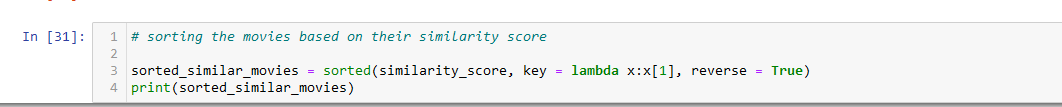
Next step is taking input from the user

After that create a list of all movies containing in the dataset now we are finding the close match for the movie name given by the user

Next we will generate a list of similar movies. We will use the movie\_index of the movie we have given as input movie\_user\_likes. The enumerate method will add a counter to the iterable list cosine\_sim and return it in a form of a list similar\_movies with the similarity score of each index.



Next step is to sort the movies in the list similar\_movies. We have used the parameter reverse=True since we want the list in the descending order, with the most similar item at the top.



GETTING LIST OF RECOMMENDED MOVIES TO THE USER

Now, here comes the last part of the project, which is to print the names of the movies similar to the one we have given as input to the system



SUMMARY

In this machine learning project, we build movie recommendation systems. We built a content-based recommendation engine that makes recommendations given the title of the movie as input.