Introduction-To build a movie recommendation engine that takes a single movie in input and returns a collection of movies that are similar to it..
There are Three ways to build a movie recommendation engine-

1. Popularity based
2. Content based
3. Collaborative filtering

In this Project we are making a movie recommendation system Using Content based filtering.

Prerequisites-

The making of a movie recommendation system requires the basics of Python,machine learning, statistics,and algebra,covered in theory part of the project.

Theory-

Movie recommendation engine-

There are Three ways to build a movie recommendation engine-
1.Popularity based-In this we keep a record of the number of views on a particular movie and then recommend a user the list of most watched movies sorted by views.
2.Content based-In this type of engine we take a movie from the user as input then analyse this movie based on this content(storyline ,genre,cast etc),then we give a list of movies in sorted order which are similar to its content
3.Collaborative filtering- This algorithm at first tries to find similar users based on their activities and preferences (for example, both the users watch the same type of movies or movies directed by the same director). Now, between these users(say, A and B) if user A has seen a movie that user B has not seen yet, then that movie gets recommended to user B and vice-versa. In other words, the recommendations get filtered based on the collaboration between similar user's preferences (thus, the name "Collaborative Filtering"). One typical application of this algorithm can be seen in the Amazon e-commerce platform, where you get to see the "Customers who viewed this item also viewed" and "Customers who bought this item also bought" list.

Steps needed to make a Movie recommendation system-

1.Data extraction-We have taken aur data from trusted sources like IMDB which has the list of movies,along with its features like the director,cast genre etc.

2.Data cleaning-Is done by pandas library.All we had to do is replace all the null or NA values,it is done by fillna() method from pandas library.

3.Building the machine learning model-It has done using python on google colab platform with the help of machine learning libraries like numpy pandas and sklearn and the algorithm discussed below.

Machine learning-

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed.

Some machine learning methods-

1.Supervised machine learning- we can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.
Unsupervised machine learning-

2.Unsupervise learning- When the information we are using is neither labeled,not classified unsupervised learning is used.It may be used for information like clustering,anomaly detection,association mining.

Here in our project we have used supervised machine learning coz we have a dataset of movies, and we have trained our data accordingly to get the desired outcome.

Algorithm for movie recommendation-

1.Read csv file
- Using pandas library in python

2.Make a data frame that contains the set of all the movie info.
- Using read_csv() method
- Attributes
  - Column->Features

2.Select features-
- Select the features form the column
  - In this project we have used features like
    - Cast
    - Genre
    - Director
    - Budget
    - Popularity
    - Production Companies
    - Production countries
    - Release date
    - Crew
    - Revenue
    - And a keyword for identification

3.Make a function to combine all the features in one big string.

4.Apply this function in the data frame

5.Go over all the features and fill all the nan values by any empty string
- It is done so that our program may not give an error when we combine all the features of a film in one string

6.Create a column in Data frame which combines all features

4.Create a count matrix from new combined column
- Count vectoriser class,fit_transform method is used
- Instead of text we have data frame of combined features

5.Compute cosine similarity based on count matrix
- It is done by using `cosine_similarity() method from sklearn library`

5.Get index of this movie from its title
- We enumerate the row of the movie index,it gives list of tuples
- enumerate() method is used to do so
- The first value of the tuple contains the index of the movie while the second contains its similarity values.

6.Now sort list of tuple using the second value of the tuple.

6.Get sorted list of similar movies.

7.Print first 50 similar movies.

Explanation of the algorithm-

- Here we have read the csv file and converted features of a movie in a string,then each feature of the movie acts as the basis of the vector space of our data frame.
- And a count matrix is being made
- Now we have a collection of movies in form of vector now we have to calculate the similarity between each vector
- The similarity between every vector is calculated using the cosine similarity method and we get a symmetric matrix whose ith row and jth column determines the degree upto which movie i is similar to j.
- Now we have trained the dataset. If we get a movie as input we check the ith row and sort the movies on the basis of similarity and output the similar movies.

Cosine Similarity method-

Cosine similarity method is used to determine how similar two vectors are.In this we measure the cosine of the angle between two vectors projected in a multidimensional space.

Mathematically, if ‘a’ and ‘b’ are two vectors, the cosine equation gives the angle between the two.

$$\vec{a} \cdot \vec{b} = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$$

$$\cos\theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\|\|\vec{b}\|}$$

$$\|\vec{a}\| = \sqrt{a_1^2 + a_2^2 + a_3^2 + \cdots + a_n^2}$$

$$\|\vec{b}\| = \sqrt{b_1^2 + b_2^2 + b_3^2 + \cdots + b_n^2}$$

$$\textit{Consider following vectors}$$

$$a:[1,1,0]$$

$$b:[1,0,1]$$

$$\textit{Norm of vector } a \textit{ is} = \|\vec{a}\| = \sqrt{1^2+1^2+0^2} = \sqrt{2}$$

$$\textit{Norm of vector } b \textit{ is} = \|\vec{b}\| = \sqrt{1^2+2^2+1^2} = \sqrt{2}$$

$$\vec{a}\cdot\vec{b} = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + a_3 b_3 = 1\times1+1\times0+0\times1 = 1+0+0 = 1$$

$$\cos\theta = \frac{1}{\sqrt{2}\times\sqrt{2}} = \frac{1}{2} = 0.5$$

$$\theta = \cos^{-1} 0.5 = 60^{\circ}$$

To compute the cosine similarity, you need the word count of the words in each document. We use the countVectorise from scikit-learn.

Example-

Suppose we have to compare the similarity between the two texts-
1.java python java
2.python python java
First we have to build a count matrix for every line which has every unique word as the basis of both the vectors,then for a given text the ith row of the matrix we count the no. of times each basis has appeared and place it in the matrix.
Here the basis are java and python
So the count matrix contains[[2,1],[1,2]],now we find the cosine distance between the two vectors and place it in the cosine matrix.
So cosine matrix looks like [[1,0.8],[0.8,1]]
The value of the ith row and jth column represents the extent of similarity between ith vector and jth vector.

Why cosine Distance over euclidean distance is used-

- The Cosine similarity can still be calculated even though n is unknown.
- The magnitude difference between 20 and 1000 does not significantly change that these vectors are well aligned (cosine = 0.97).
- As the dimensional space becomes large, this still works well, which is why it is used for lots of text similarity applications.
- Without adjustment, the output is always between 0 and 1.

Description of library functions used-

1.From pandas-it is used to extract the dataset from a given source

- read.csv() function is used to load the dataset in a dataframe
- fill.na() method,is used to fill all the nan values from the dataframe by its entered parameter

2.From Scikit-learn-

- CountVectorise() method is used to convert a text document to a vector of terms and token counts
- cosine_similarity() method this function takes a matrix as input and returns the cosine similarity matrix

3.enumerate() method adds a counter to an iterable and returns it in a form of enumerate object. This enumerate object can then be used directly in for loops or be converted into a list of tuples using list() method.

4. sorted() method function returns a sorted list of the specified iterable object.

References-

- https://scikit-learn.org/stable/
- https://www.python.org/
- https://pandas.pydata.org/
- https://www.imdb.com/
- https://www.geeksforgeeks.org/
- https://www.tutorialspoint.com/index.htm
- https://colab.research.google.com/
- https://github.com/