Movie Recommendation System Using Collaborative And Content-Based Filtering

Saurabh Bhalla[1], Baibhav Kumar[2], Rajat Tiwari[3], Dr. P.A Jadhav[4]

[1,2,3]Students, [4]Professor
Department of Information Technology
BV(DU)COE, Pune

***Abstract*: A recommendation system uses various algorithms for giving the most preferable and relevant items to users. The system checks the past behavior of a person and gives similar results that might be likely preferable to users. Suppose a new user visits an e-commerce site, it doesn't have any past history of the user, so in this scenario how will not show any recommendation to a new user? In this case, the site can recommend the best-selling product, I e. the product which is high in demand. Another conceivable arrangement could be to prescribe the items which would carry the greatest benefit to the business. Three main approaches are used for our recommender systems. One is Content-based i.e they offer generalized recommendations to each and every user, based on movie popularity and/or genre. The System recommends the same movies to users with similar content features. Since each user is different, this approach is considered to be too simple. The basic idea behind this system is that movies that are more popular and critically acclaimed will have a higher probability of being liked by the average audience. Second is Knowledge-based filtering, where we try to profile the user's interests using information collected, and recommend items based on that profile. The other is collaborative filtering, where we try to group similar users together and use information about the group to make recommendations to the user.**

***Keywords*: Python Language, Content database, DJango, Bootstrap, Numpy, Panda, Scipy**

Introduction:
A recommendation system is a type of knowledge filtering system which attempts to predict the preferences of a user and make a suggestion based on their choices. There are many types of applications for recommended systems. These have become very popular over the last few years and are now utilized in most online platforms that we use. The content of such stages fluctuates from motion pictures, music, books, and recordings, to companions and stories via web-based networking media stages, to items on web-based business sites, to individuals on professional and matrimonial sites, to indexed lists returned on Google. Often, these systems collect information about users' choices and use this information to improve their suggestions in the future.
For example, if Amazon observes that a large number of customers who buy the latest Apple MacBook also buy a TYPE-C-to USB Adapter, they can recommend the Adapter to a new user who has just added a MacBook to his cart. Due to the advances in recommender systems, users constantly get good recommendations. Suppose a music streaming app is not able to find the music that the user likes, then the user will simply stop using it. This has led to high stress for tech companies on improving their recommendation systems. However, the problem is bigger than it seems. Every user has different preferences and likes. They must explore new domains to discover more about the user, while still making the most of what is already known about the user. Three principle approaches are utilized for our recommender frameworks. One is Demographic Filtering i.e They offer summed up proposals to each client, because of motion picture prevalence or potential classification. The System prescribes similar motion pictures to clients with comparative segment highlights. Since every client is diverse, this methodology is viewed as excessively straightforward. The essential thought behind this framework is that films that are progressively mainstream and widely praised will have a higher likelihood of being enjoyed by the normal crowd. Second is content-based separating, where we attempt to profile the client's intrigues utilizing data gathered and prescribe things dependent on that profile**.**

LITERATURE SURVEY-

A recommendation system provides suggestions to the users through a filtering process that is based on user preferences and browsing history. The information about the user is taken as an input. The information is taken from the input that is in the form of browsing data. This information reflects the prior usage of the product as well as the assigned ratings. A recommendation system is a platform that provides its users with various content based on their preferences and linkings. A recommendation system takes the information about the user as an input. The recommendation system is an implementation of the machine learning algorithms.

Python Language- Python is a high-level general-purpose programming language which is used for creating programs with mathematical source codes with easy readability and faster usability, its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together.

Methodologies-

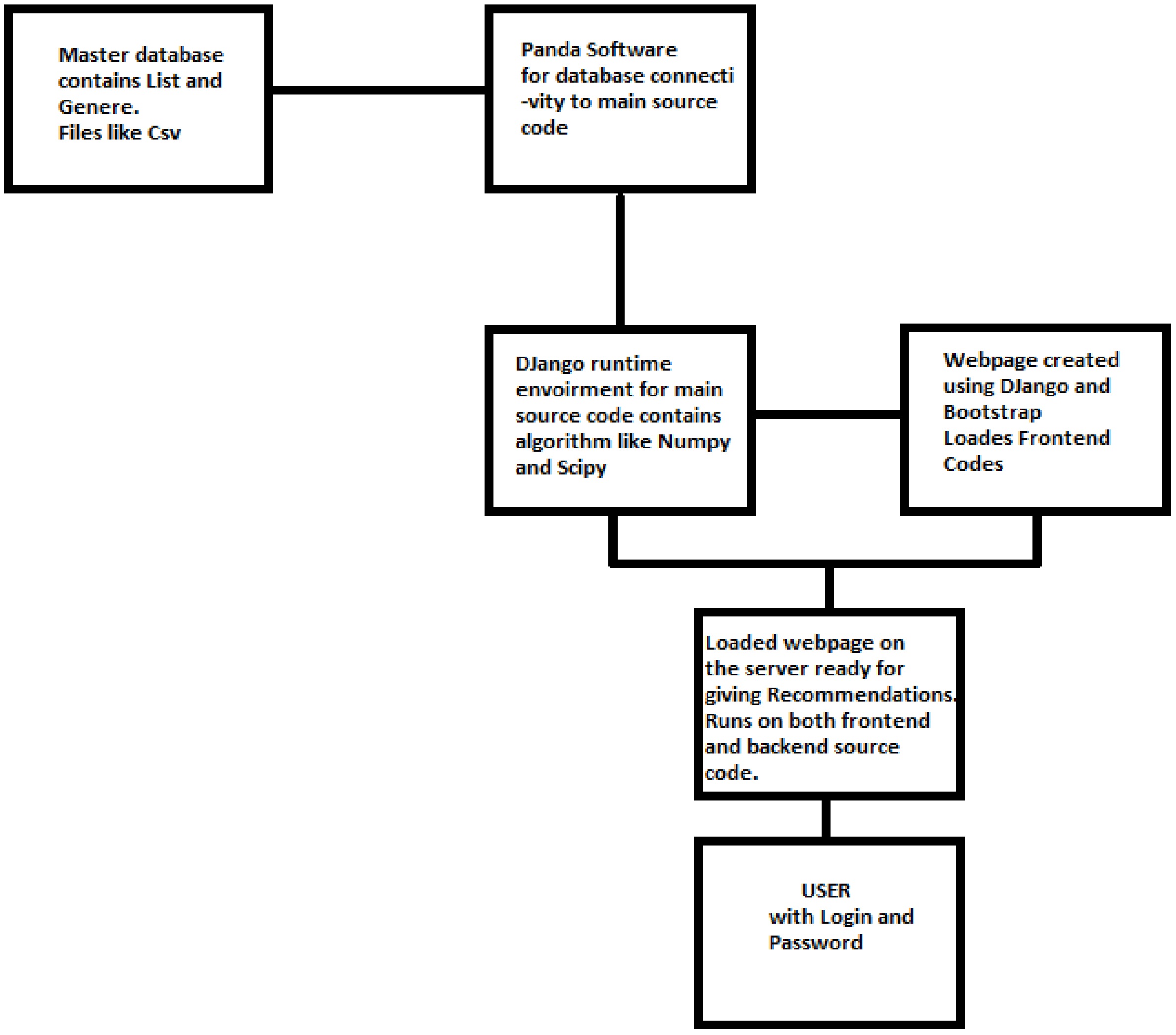
Existing system-
Content-based filtering uses item features to recommend other items similar to what the user likes, based on their previous actions or explicit feedback. To demonstrate content-based filtering, we should hand-engineer a few highlights for the Google Play store. The accompanying figure shows a component grid where each line speaks to an application and every section speaks to an element. Highlights could incorporate classes, (for example, Education, Casual, Health), the distributor of the application, and numerous others. To disentangle, accept this component framework is double: a non-zero worth method the application has that include. You also represent the user in the same feature space.

Some of the user-related features could be explicitly provided by the user. For example, a user selects "Entertainment apps" in their profile. Other features can be implicit, based on the apps they have previously installed. For example, the user installed another app published by Science R Us. The model ought to prescribe things significant to this client. To do as such, you should initially pick a closeness metric (for instance, spot item). At that point, you should set up the framework to score every competitor thing as indicated by this likeness metric. Note that the suggestions are explicit to this client, as the model didn't utilize any data about different clients.

Proposed system- With our proposed system we aim at building a system that gives more accurate results. Using Python language, we aim at creating a source code that is also compatible with our GUI. With the help Machine learning library like NumPy and scipy, we are making the system that can do mathematical dimensional array and matrices calculation on its own, and data analysis with the help of the panda software. In our project we use Python language as the main source code which will be implemented in an IDE development environment, Comand prompt will compile the source code and will download and install the required prerequisite in the system, which will then start the local server and will also provide the link to the website. The database which we are going to use contains Movie information and Ratings and as per that given information the system is going to give a recommendation based on the two algorithms:

For the frontend, we are going to use Django and Bootstrap for the creation of our GUI which is both user friendly and attention seeker. Panda is going to connect our GUI with the Backend Source code and Database, then after everything is done, our system is going to start giving recommendations to the users which will be the final phase of our system.

SYSTEM ARCHITECTURE-


Master database contains List and Genere. Files like Csv
Panda Software for database connecti -vity to main source code
DJango runtime envoirment for main source code contains algorithm like Numpy and Scipy
Webpage created using DJango and Bootstrap Loades Frontend Codes
Loaded webpage on the server ready for giving Recommendations. Runs on both frontend and backend source code.
USER with Login and Password

Architecture Overview-

1.) Master database: Our system will automatically restore data from database which are saved in 'CSV' format, the database contains list Movies, their genres, rating, and user's interaction time.
2.) Python language Source code: Containing content-based filtering and collaborative filtering algorithms with essential libraries like 'Numpy', 'Scipy'.
3.) Django: Acting as a platform where source code is compiled and creating a path between webpage and data. Master database containing the data.
4.) Webpage: Act as a GUI interface for the user to get recommended suggestions. It is created with the DJango framework and bootstrap designing for frontend and Python language as backend.

Collaborative Filtering- In our implementation, we are going to use 2 types filtering method:
1.User-based Collaborative filtering- Firstly, we will have to predict the rating that user 3 will give to item 4. In user-based CF, we will find say k=3 users who are most similar to user 3. Commonly used similarity measures are cosine, Pearson, Euclidean, etc. We will use cosine similarity here which is defined as below

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2}\sqrt{\sum\limits_{i=1}^{n} B_i^2}}$$

"Cosine Similarity"

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

"Pearson Correlation"

In the NearestNeighbors method can be used to search for k nearest neighbors based on various similarity metrics. Find similar users function uses this method to return the similarities and indices of k-nearest neighbors for an active user. The function predict_userbased further predicts the rating that user 3 will give to item 4, using a user-based CF approach. Predictions are computed as a weighted average of deviations from neighbor's mean and adding it to an active user's mean rating. Deviations are used to adjust for the user associated biases. User biases occur as certain users may tend to always give high or low ratings to all items.

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u \in K}\left(r_{u,i} - \bar{r}_u\right) \times w_{a,u}}{\sum_{u \in K} w_{a,u}}$$

Item-based Collaborative filtering-In this approach, similarities between pairs of items are computed using a cosine similarity metric. The rating for target item I for an active user to being predicted by using a simple weighted average as:

$$p_{a,i} = \frac{\sum_{j \in K} r_{a,j} w_{i,j}}{\sum_{j \in K} |w_{i,j}|}$$

where K is the neighborhood of most similar items rated by an active user a, and w(i,j) is the similarity between items i and j the function find similar items uses the NearestNeighbors method employing cosine similarity to find k items similar to item *i*. The function predict_itembased further predicts the rating that user 3 will give to item 4, using an item-based CF approach.

Selection of similarity matrix-

Some helpful clues while selecting matrix are-

1) Use Pearson when your data is subject to different rating scales of the user.
2) Use Cosine, if data is spare.
3) Use Euclidean when data is not spare and the magnitude of the attribute value is significant
4) Use adjusted Cosine for item-based approach to adjust for user bias

Objectives:

1.) To get the most accurate recommended suggestions.
2.) To create a GUI with a user-friendly interface.
3.) To apply both content-based and collaboration algorithm.

Conclusion-

Recommendation Systems are the most well-known kind of Machine learning applications that are utilized in all areas. They are an improvement over the customary arrangement calculations as they can take numerous classes of information and give closeness positioning based calculations to give the client exact outcomes. These Recommendation systems have evolved and have incorporated many advanced Machine learning systems to give the clients the substance that they need.

References-

[1] Personalized Recommendation Combining User Interest and Social Circle Xueming Qian, Member, IEEE□ He Feng, Guoshuai Zhao, Tao Mei, Senior Member, IEEE

[2] Nilashi, M., Ibrahim, O., Bagherifard, K.: ‘A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques’, Expert Syst. Appl., 2018, **92**, pp. 507–520

[3] Tarun Bhatia: Recommendation System: Technology Review, ResearchGate 2018.

[4] Urszula Kuzeleswska: Clustering Algorithms in Hybrid Recommender System on MovieLens Data, SILGR 2014.

[5] Patrikainen, A., Manilla, H.: Subspace clustering of high-dimensional binary data - a probabilistic approach. In: Workshop on Clustering High Dimensional Data and its Applications, SIAM International Conference on Data Mining (2004).

[6] Zan Wang, Xue Yu, Nan Feng, Zhenzua Wang, An improved collaborative movie recommendation system using computational intelligence, Elsevier 2014.