**ABSTRACT**

A recommender system, a subclass of information filtering systems, aims to predict the rating or the preference a user might give to an item. Based on this prediction, it would suggest relevant items to the users, making their experience more personalized and enjoyable. For example, in the case of Netflix, it would suggest movies; in the case of kindle, it would be books. In this project, we worked with MovieLens dataset, then proceeded for building two types of recommender systems: content and item based collaborative system. The former system recommends relevant movies to the users by using content of the previously searched items that the user likes, while the later one deals with finding similar movies using SVD model. Here, content refers to the attribute of the movie, that is genre. At the end of the work, evaluation parameters have been incorporated in the systems, achieving 0.65 in average MAP,0.865 in average NDCG, while a score of 0.6410 and 0.4968 has been seen respectively in RMSE and MAE.

**WHAT OTHERS HAVE DONE**

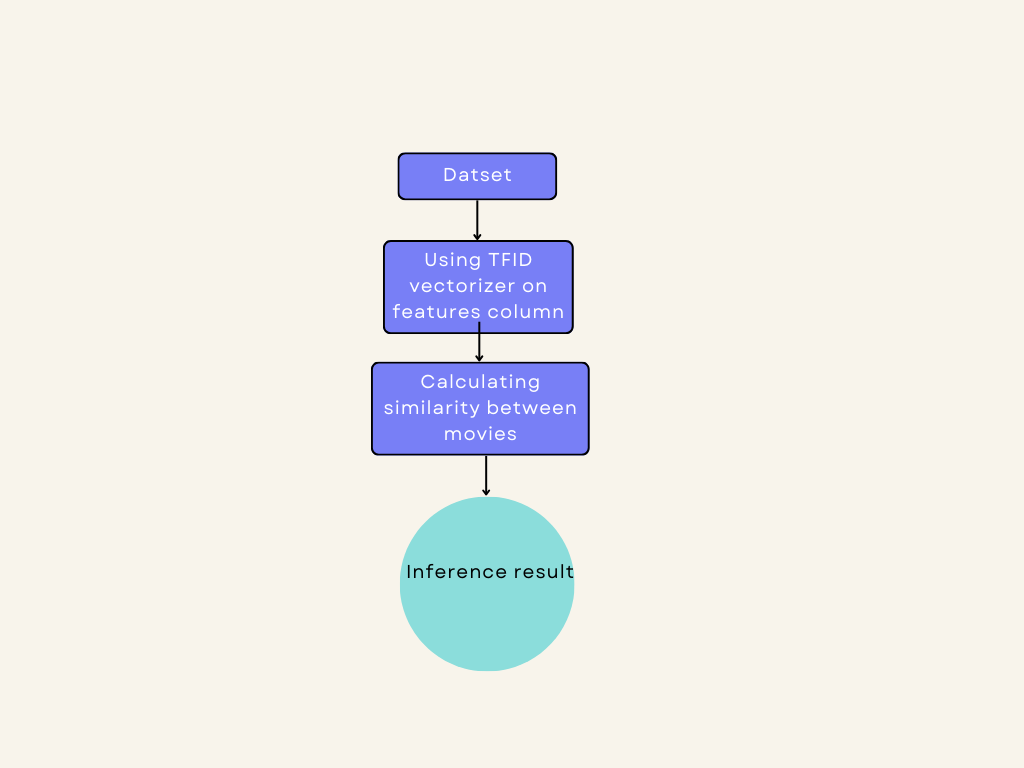
1. The user/item specific information is grouped to form a cluster using K-means clustering
2. Bayesian belief nets CF (collaborative filtering) was implemented
3. Some have used MDP based CF for improving predictions
4. Matrix factorizations on CF has been performed

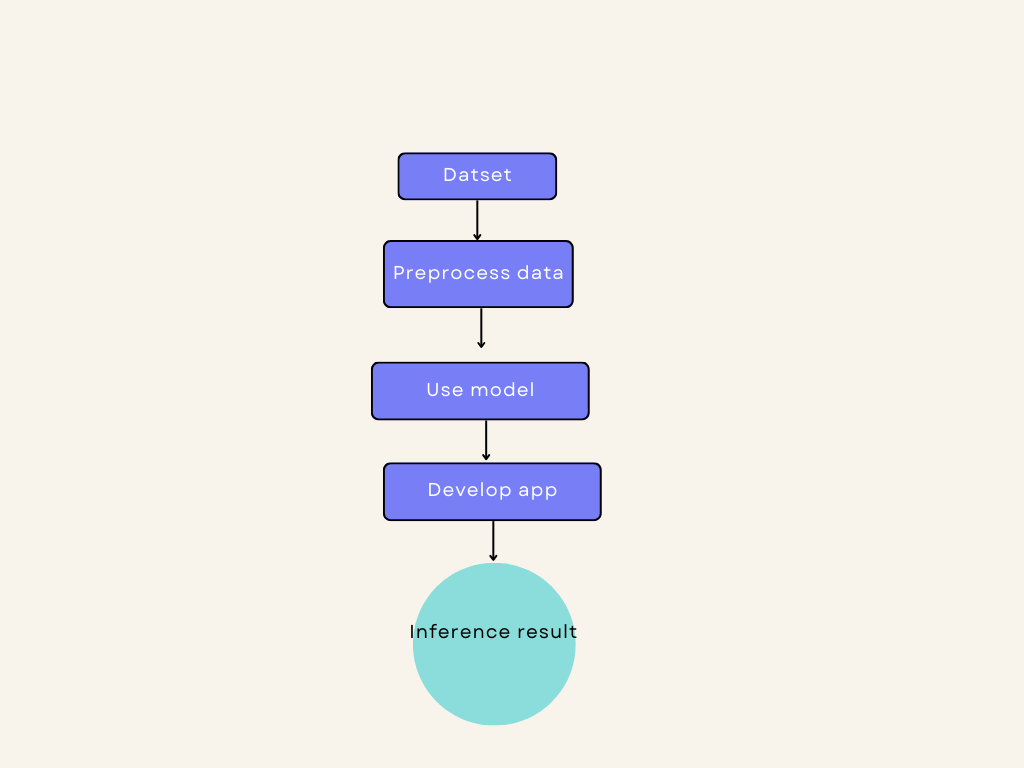
**LIMITATIONS**

1. With an increase in the number of users and movies, the computational time of the system increases.
2. Model building is expensive in case of Bayesian belief nets
3. There is trade-off between prediction performance and scalability
4. Lose useful information since only user id and movie id is used to create the embeddings

**METHODOLOGY**

i)Flowchart:

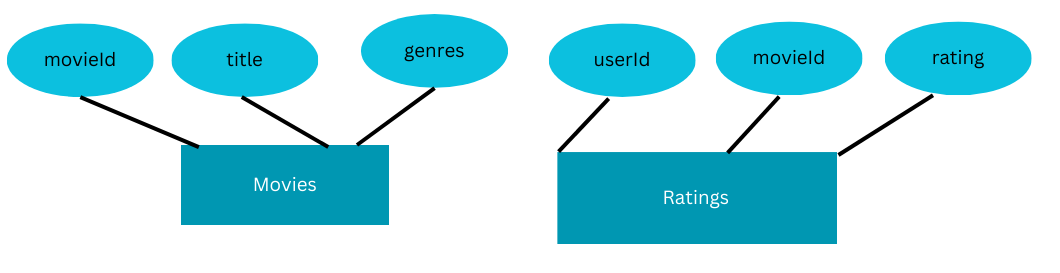


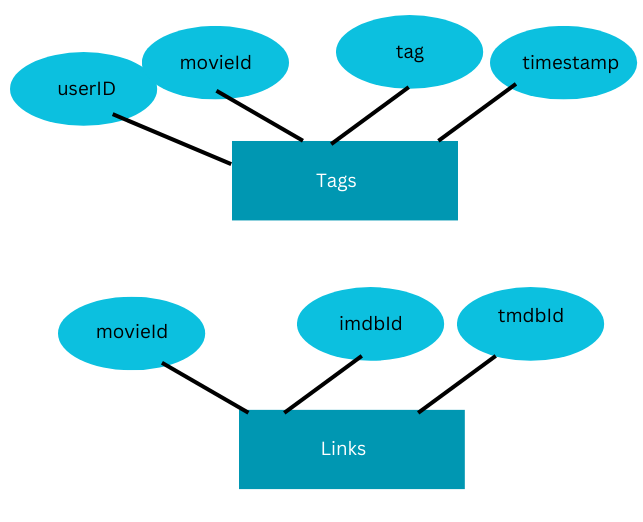


ii)Dataset:

We used MovieLens dataset(ml latest small). It has 3 tables: movies, ratings, links and tags. For our project, we used only 2 tables: movies and ratings. The movies csv file has user id, movie id and genre while the ratings file contains user id, movie id, ratings and genre. All of this information is fundamental when we will apply filtering. For tags csv file, it contains the tags that best describes the movie. As for the links csv file, it contains movie id, imdb id and tmdb id.

iii) At first, the movies and ratings file have been merged using their common attribute, that is movie id. This is done to get the overall information of the movies. After this, EDA was performed on the dataset. We checked the presence of Nan values in any column and if found, the rows were dropped.The datasets are represented using ERD.

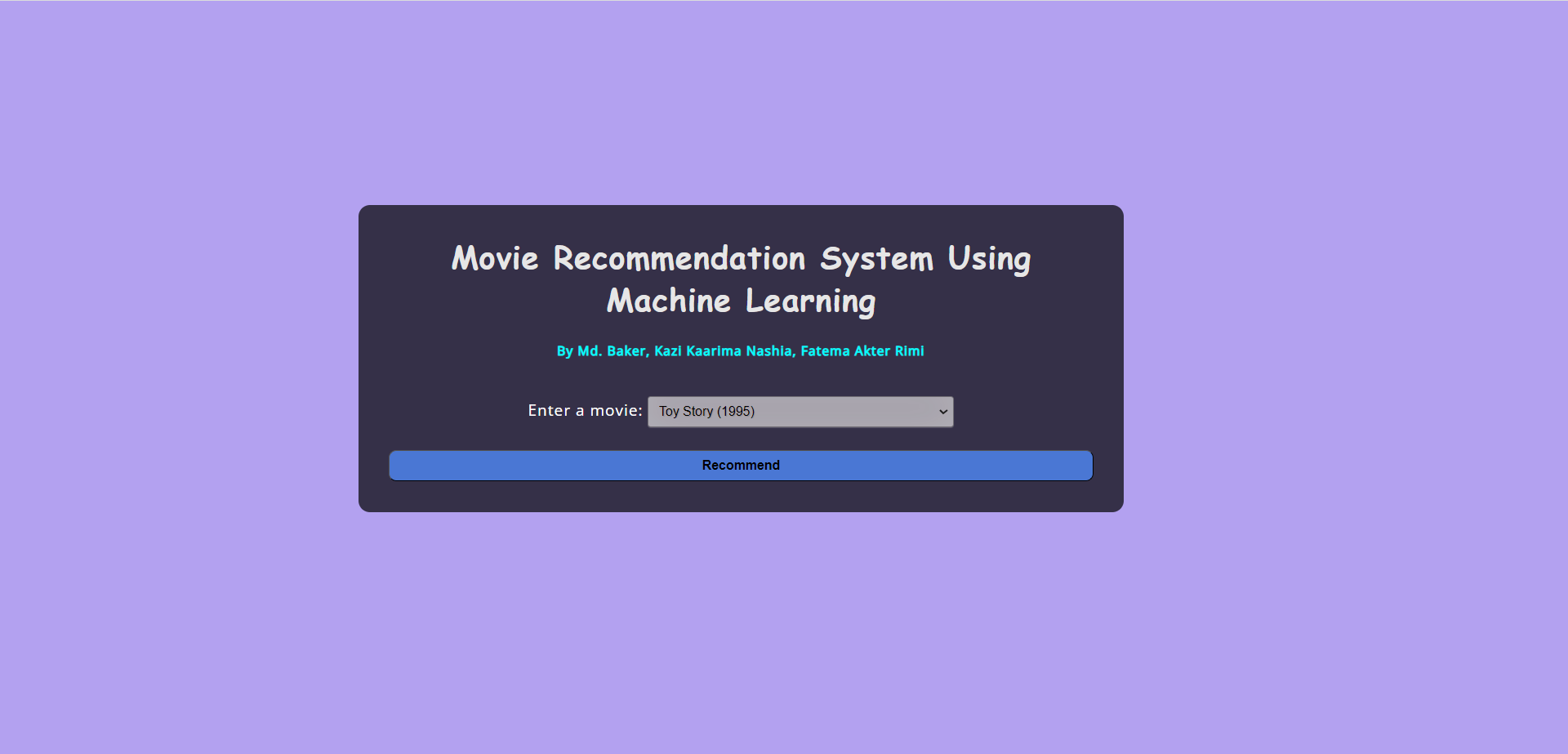




iv) Models and Programming languages: Initially, we worked with various models such as Decision Tree, Random Forest, Global K, followed by content filtering and collaborative filtering. In the end, we selected collaborative filtering as its accuracy was best compared with the other models.

For implementation, Python, being the backbone of machine learning, was used along with Flask for developing the web application, accompanied by HTML and CSS.

v) App:



**RESULT**

Results were evaluated using RMSE,MAE,NDCG

RMSE=

MAE=