National College of Ireland

Top-N Nearest Neighbourhood based Movie Recommendation System using different Recommendation Techniques

MSc Research Project
Data Analytics

Muhammad Imran Shaikh
Student ID: x17119308

School of Computing
National College of Ireland

Supervisor: Dr. Muhammad Iqbal

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Muhammad Imran Shaikh
Student ID:	x17119308
Programme:	Data Analytics
Year:	2020
Module:	MSc Research Project
Supervisor:	Dr. Muhammad Iqbal
Submission Due Date:	28/09/2020
Project Title:	Top-N Nearest Neighbourhood based Movie Recommendation System using different Recommendation Techniques
Word Count:	5862
Page Count:	22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	27th September 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Top-N Nearest Neighbourhood based Movie Recommendation System using different Recommendation Techniques

Muhammad Imran Shaikh
x17119308

Abstract

Background: Recommendations engines are extremely common and utilized by many tech giants like Facebook, Google, IMDB, Netflix, financial services, and many other companies. The task is to build a Top-N Nearest Neighborhood-based movie recommender system using the recommendation techniques such as Content-based filtering, Collaborative filtering, and Matrix factorization.
Objective: This research project aims to build a recommender system that will recommend movies to the user just not only by predicting rating but also on the similarity basis of similar users and their interested items. The recommender engine will find out the nearest neighbors around that specific user by matching the similarity of items and ratings for the items given by those nearest neighbors.MovieLens dataset(ml-latest-small) is considered for this research which contains movie,ratings CSV files. Five models have been used i.e. KNNBaseline, KNNWithMeans(UB and IB), SVD, and SVDpp.Two cross-validation techniques have been utilized K-Fold CV and LOO(Leave One Out)CV to attain the best accuracy value from multiple machine learning models. Two accuracy measures have been considered which are RMSE and MAE.
Results: The best RMSE accuracy score result is achieved with the SVDpp algorithm by implementing it on the matrix factorization recommendation technique, the final accuracy mean of the model was almost 88% RMSE with 66% MAE score from K-Fold cross validator while 91% RMSE score 69% from LOOCV cross validator.

1 Introduction

1.1 Overview

Recommender systems are considered to be as one of the core applications of machine learning for increasing the commercial business. This section interrogates the overview of the use of the Recommendation system particularly on movies dataset along with different techniques of filtering techniques such as Content-based filtering, User and Item-based collaborative filtering, and Matrix Factorization. Many industrial applications have implemented the recommender system approach in sectors like e-commerce, social media, movies, etc. Big tech giants like Facebook, YouTube, IMDB, Instagram are facilitating their company business growth by improving the recommendation system algorithm to

give the user the best recommendation results. The Recommendation system mainly comprises of two basic types Content-based Filtering and Collaborative filtering (Kane; 2018). The main aim of this research is targeted towards the domain of the education sector. It is a small-scale project which has utilized different Recommendation system techniques by testing various models' accuracy to present the best possible results to the user.

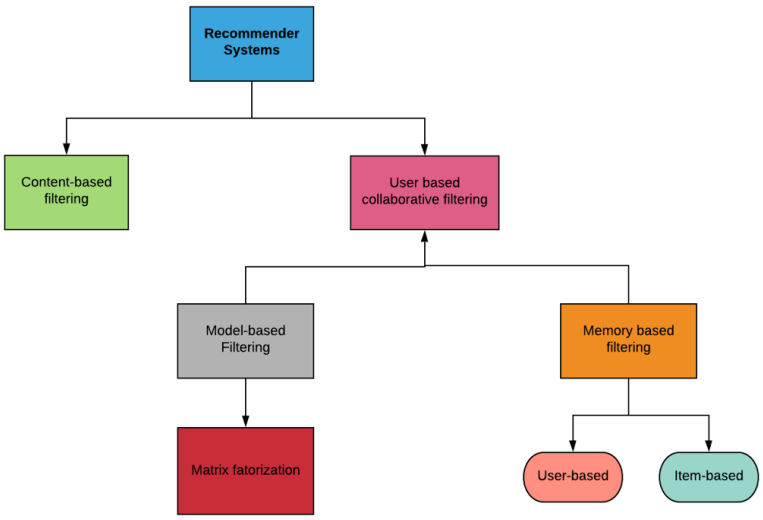

Figure 1: Basic Recommendation System

There are two basic techniques of Recommendation filtering, Content-based filtering, and User-based collaborating filtering in Figure 1. The selection for the Content-based filtering, also known as the cognitive filtering, It is dependent on the profile of the user and items' content. On other hand, the User-based collaborative filtering evaluates the items by users that have similar favorites rating or feedback as it recommends the items based on the group of people that have similar or close preferences (Sirikayon et al.; 2018). Matrix factorization is the most used Model-based collaborative filtering technique applied to discovers each user and item interaction and rating prediction based on inner latent factors product. It is too native on each latent factor.

1.2 Motivation

The motivation of this research is to understand the behavior of the recommendation system by implementing various techniques and machine learning models. Recommendations engines are extremely common and utilized by many tech giants like Facebook, Google, IMDB, Netflix, financial services, and many other companies. Netflix is one of the biggest industries which has implemented a recommendation system and improved it constantly for their business revenue generation. Moreover, (Li et al.; 2020) experimented that the improvement in the model-based collaborative a.k.a Matrix factorization which utilizes SVD(Singular Value Decomposition)algorithm to attain better RMSE score of 0.85 and present a new winner of Progress prize in recommendation algorithm competition.(Richter; 2020)shows the global expansion in Netflix subscribers growth results in

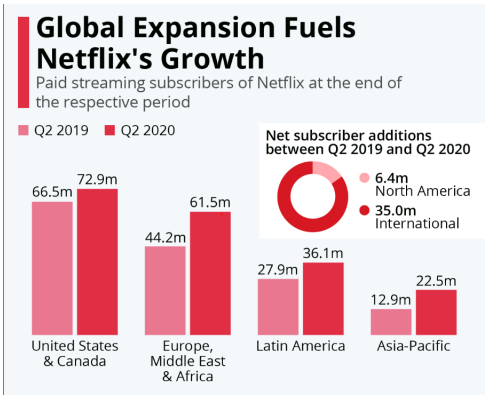

Figure 2: Netflix's Global Subscribers Growth Q2 2019 - Q2 2020 (Richter; 2020)

subscriber additions of 6.4 million in North America and around 35.0 million in the rest of the world between Q2 of 2019 and Q2 2020 in Figure 2

1.3 Objective

This research aims to build a recommender system that will recommend movies to the user just not only by predicting rating but also on the similarity basis. The recommender engine will find out the nearest neighbors around that specific user by matching the similarity of items and ratings for the items given by those nearest neighbors. Four models have been used i.e. KNNBaseline, KNNWithMeans, SVD(Singular Value Decomposition), and SVDpp(Singular Value Decomposition plus plus) with different similarities metrics like Pearson baseline and Cosine similarity. Two cross-validation techniques have been utilized K-Fold CV and LOOCV(Leave one out CV) to attain the best accuracy value from multiple machine learning models. Two accuracy measures have been considered which are RMSE and MAE. The research project objective includes the following steps :

- Data collection from Movielens dataset and to perform Data Analysis on it.
- Creation of a separate movie class which contains both movies and rating dataset in CSV format to utilize it as a final data to test and train different recommendation machine, learning models.
- Present top 10 nearest neighborhood-based movies result from recommendation techniques like Content-based filtering, User-Item based collaborative filtering, and Matrix Factorization along by implementing different machine learning algorithms.
- Find out the best parameters for different machine learning models by Models Hyper-parameter tuning to get the optimal results.
- Finally, in model evaluation two cross-validation techniques have been utilized i.e. K-Fold CV and LOO(Leave One Out)CV to attain the best accuracy values from multiple machine learning models.

The rest of the sections are arranged as follows: Section 2 discusses previous related research work. Section 3 describes the methodology used for this project, Section 4

underline proposed recommendation engine design specification, Section 5 demonstrates practical implementation outcomes, Section 6 represent models evaluation results and discussion. Section 7 discusses Future work and the conclusion.

1.4 Research Question

Q : How can we recommend Top-N good recommended movies according to the user's interest by using the nearest neighborhood-based recommendation system techniques and machine learning models?

2 Related Work

2.1 A Research on the Recommendation Engine Model

(Zhao; 2019) explains that there are four important stages in the process of recommendation, that work in a continuous loop. The recommendation process starts with the collection and organization of information, the data history is attained, that is used to predict results based on rankings for recommendations, which is then looped back to data collection.(Kane; 2018) defines there are two basic techniques of filtering information, Content-based filtering, and User-based collaborating filtering. The selection for the Content-based filtering also known as the cognitive filtering, It is dependent on the profile of the user and items' content. The product representation is based on the factors that differentiate the contents and are used in making recommendations for e-commerce if the user would purchase or like the item or not. The strength of each factor is demonstrated through the creation of the vectors for each item, which is then paired with the user with the similar strength and interest for that factor using the user's data history.(Çano and Morisio; 2017) discuss that, the Hybrid filtering solution reduces the difficulties faced in collaborative filtering and content-based filtering through its capability to adjust between the filtering techniques. For instance, the collaborative filtering uses the user rating matrix system whereas, the content-based filtering system has a choice between the user profile and items content, so the hybrid filtering technique can switch to content-based recommendation when there is no user rating.

2.2 A Research Study on the Content Based Filtering

By gathering the personal information of movie item description and user based on the history of the user which he/she used to like in the past. (Malik and Bansal; n.d.) experiment that with the help of surprise library KNN algorithm has been utilized by setting its different parameters and cosine similarity metric to get the best possible result. The two recommendation techniques i.e. collaborative filtering and content-based filtering have their limitation and problems like cold start so, to gain more efficient results Hybrid filtering approach comes for rescue.(Ahuja et al.; 2019) finds that K-mean Clustering and K-Nearest neighbor algorithms have been utilized in content and collaborative based filtering of recommendation system. Movies are clustered in a way that is similar to each other with other clusters along with KNN to obtain the best optimal results. The best RMSE value achieved is 1.081.(Ariff et al.; 2018) compares Content-based filtering and collaborative filtering which shows that both have their pros and cons. Content-based filtering focuses on the contentment of the object like for movie genres are considered as

a user liking parameter on the other hand with collaborative filtering the user rating similarities matters in collaboration with the item description. The evaluation performance of collaborative filtering was more optimum as compare content-based filtering because the movies that have been watched by the user is not recommended to the user.(Nilesh et al.; 2019) attempts the series of vector to check the similarity is attained by using the Cosine similarity matrix with content-based filtering that recommends recipe to the user. Content-based filtering is the main extraction feature as it recommends the contents that are matched with the corresponding profile of ingredients.(Rajarajeswari et al.; 2019) applies similarities metrics like Pearson baseline and Cosine similarity on the attributes of the movies to compare donates similarity results among the two movies on TMDB ratings. Nearest neighborhood-based algorithm (K-Nearest Neighbour) is utilized to get an output of 30 movies recommendation.

2.3 A Research Study on the User and Item (Memory) Based Collaborative Filtering

(Isinkaye et al.; 2015) discusses about different techniques evaluation and limitations. In content-based filtering, the Accuracy result is observed to be less as in contrast to collaborative filtering whereas the collaborative filtering technique suffers from a cold start problem and requires large datasets for users and items. The performance evaluation gauge is divided into 3 ways offline, Online and User study.(Pal et al.; 2017) explains that improve the sparsity level different MAE (Mean Absolute Error) values of collaborative filtering techniques like Hybrid CF, Pure CF and SVD (Singular Value Decomposition) has been considered. Among all the techniques the Hybrid CF approach gives better MAE value to improve sparsity b/w 1 to 2% depending on dataset.(Afoudi et al.; 2018) describe the advantage of Collaborative filtering which is further divided into two sections Memory-based collaborative filtering and Model-based collaborative filtering. The advantage of memory-based filtering is its easy implementation by just calculating the similarity score and it is scalability for the large dataset.(Venil et al.; 2019)presents the most representative approach is nearest neighborhood collaborative filtering where different models of KNN has been applied to gain result from Item based, User-based, and Ensemble-based collaborative filtering technique along with different similarity metrics. By testing the performance of the recommender model Ensemble-based KNN shows better RMSE (1.07) and MAE (0.85) as compared to the rest of the techniques.(Dong et al.; 2016) experiments to provide a personalized recommendation to a news system, the recommendation algorithms prefer user-based collaborative and content-based filtering. The user-based collaborative filtering measures the average user ratings by creating a user rating matrix, Nearest neighbor selection for user similarity calculation. There is a deviation in the experimental results for each algorithm, CF accuracy of 83.12% with Sd 18.10%, Item-based filtering produce accuracy of 69.19% with Sd of 21.52%, and the Hybrid approach shows 89.85% with Sd of 7.60%. This illustrates that the hybrid approach has more impact on producing efficient and accurate results for news recommender system. (Thukral and Ramesh; 2018) spells out that Similarity measures like Cosine, Euclidean Manhattan, Jaccard, Ensemble, and Pearson have been verified to find out MAE and RMSE errors. In final comparison Ensemble and Jaccard similarity provides good recommendation quality with implicit ratings while Pearson and Cosine Similarity metrics are considered to be best similarities measures for explicit ratings.

2.4 A Research Study on Matrix Factorization

(Sirikayon et al.; 2018) explains to improve the performance of the students, An efficient library book recommender engine has been designed utilizing the collaborative filtering recommendation system. There were several techniques used for experiments, including similarity calculation, prediction, and recommendation. The Matrix factorization technique is essential to resolve the constriction of the rating matrix. The rating matrix is composed of book borrowing records with time stamps. While the similarity and distance calculation utilizes techniques like Pearson correlation, Cosine similarity, and Euclidean distance.(Bodhankar et al.; 2019) discuss some of the challenges faced by the recommender system include cold start issue, adaptability, sparsity, protection, and overspecialization. The personalized recommendation system uses fundamental matrix factorization, the Circle Con model, and Context MF Model, to design a well-organized and effective hotel recommender system. The recommendation system based on a social network along with user interest build up with an architecture comprises of User rating, User personal interest, PCC Similarity, Estimating Interpersonal Influence, Top N proposed things.(Koren et al.; 2009) Matrix factorization is the most used Model-based collaborative filtering technique applied to discovers each user and item interaction and rating prediction based on inner latent factors product. It is too native on each latent factor. It works on PCA (Principal Component Analysis) phenomena where dimensionality reduction models like SVD, SVDpp, and Weighted SVD are utilized with different epochs and RMSE scores to produce better recommendations. The best RMSE score depends on different epochs. Baseline Methods (SVD, SVD++) Outperformed by Weighted-SVD models.(Fathan et al.; 2018) shows that the combined space dimensionality latent factor is created by matrix factorization models during the mapping of users and items interaction as an inner product. A well-developed technique to extract information by identifying semantic factors is known as SVD (Singular Value Decomposition), SVD which is model collaborative filtering model factors the matrix of user and item in the dataset. (Liu et al.; 2013) finds that in Bayesian probabilistic matrix factorization one common problem faced by matrix factorization models is sparsity in the user-item data matrix and with inaccurate imputation it many cause overfittings in the final recommendation results. The results by testing different models RMSE scores states that improvement can be seen in the accuracy of the matrix factorization models by model's factor dimensionality increment.

3 Methodology

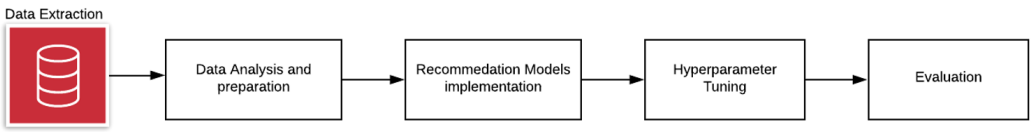

Figure 3: CRISP-DM Methodology Flow

The focus of this research project is to work on Top-N nearest neighbourhood based movie recommendation system by utilizing different Recommendation techniques.So for this research project CRISP-DM(Cross Industry Standard Process for Data mining) is utilized as methodology approach as shown in Figure 3

3.1 Business Approach

Recommendations engines are extremely common and utilized by many tech giants like Facebook, Google, IMDB, Netflix, financial services, and many other companies. Netflix is one of the biggest industries which has implemented the recommendation system and improved it constantly for their business revenue generation.(Kane; 2018) explains that there are various potential benefits that industries can facilitate. Improve the system by continuously monitoring users' preferences Like Netflix and other subscription-based platforms are availing. YouTube, Facebook, Instagram, and similar social apps take advantage but engaging customers to recommend them the stuff they like. Moreover, many other sectors are taking the edge in the industry by improving there recommender systems Such as Music, games, Web search, Apps, and others.

3.2 Data Extraction

The dataset (ml-latest-small)[1] consist of almost 100k ratings by different users with over 1200 movie tags from 9125 movies. 'ml' stands for movie lens.(Harper and Konstan; 2015) states that each selected user had at least rated 20 movies. There are 4 files included in this dataset named 'movies.csv', 'ratings.csv', 'tags.csv', 'links.csv'.This dataset intends to work on the movie recommendation service. The dataset is collected and generated by GroupLens Research Group which is the Department of Computer Science and Engineering at the University of Minnesota. The dataset is publicly available on grouplens website for machine learning and recommendation system application purpose.

Attributes	**Description**	**Domain**
MovieId	Identification of Movie	9000+ movies ids
Title	Title of movies	9126 different movie titles
Genres	Movies categories	18 genres overall in pipe-separated list

Table 1: Movie table Features Description

The Table 1 shows description about MoviesId , Movietitle and Genres. The file is in csv format. There are 18 genres overall.

[1] `https://grouplens.org/datasets/movielens/latest/`

Attributes	Description	Domain
UserId	User Id who have rated movies	650+ users
MovieId	Id for identification of Movie	9126 different movie titles
Ratings	Ratings given to the movies by different users	Ratings range (0.5 - 5.0)

Table 2: Ratings table Features Description

The above Table 2 show the information about the ratings of the different movies given my multiple users. All selected users had at least rated 20 movies.

3.3 Data Preparation and Analysis

Data Analysis and preparation on the MovieLens dataset involved steps of data cleaning by checking the null values, merging of data tables i.e. movies and ratings on movies id as an inner join, Describing Columns, Removal of unnecessary features, Genre column that describes movie genre in the movie table are separated with multiple pipes in a single cell for each movie, A separate function (count_word) is created to remove those pipes and separate genre from every single cell. The function not just only splits genres from genre column, it also counts how many numbers of times each genre appears. For initial data visualization, the merged dataset and count_word function helped to create a word cloud and histogram that illustrates the most popular movie genre. Furthermore, with ratings count for each movie title Top 25 movies with the highest ratings are plotted.

3.4 Models Descriptions

The following are the machine learning model which are used in our recommendation engine. Fortunately, the Surprise library which is an official python recommendation library holds these built-in algorithms.

3.4.1 KNNBaseLine

KNNBaseLine algorithm are based on KNN supervised classification algorithm, It is one of the best content and collaborative based filtering algorithm by taking into account baseline rating. For best prediction use the pearson_baseline similarity measure.

3.4.2 KNNWithMeans (User and Item-Based)

KNNWithMeans algorithm is based on KNN supervised classification algorithm, It is a basic collaborative based filtering algorithm by taking into account the mean ratings of each user. The switching in the user_based parameter of sim_options will result in User or Item-based collaborative filtering. When (user_based= True), It is used for User-based collaborative filtering) while user_based= False (for Item-based collaborative filtering) .For best prediction use cosine and msd(Mean Squared Difference) and similarity measure.

3.4.3 SVD

SVD (Singular Value Decomposition) is one of the popular matrix factorization algorithms which also wins Netflix prize. It works the same as PCA (Principal Component analysis). It is used to find finding latent factors that can be extracted from data.

3.4.4 SVDpp

SVDpp (Singular Value Decomposition plus plus) which is an extended version of SVD and one of the import matrix factorization algorithm. It works mostly on users' implicit and explicit ratings for the item.

3.5 Similarity Metrics

The following two similarity measures are considered for computing similarities between the user and the item table.

3.5.1 Pearson_BaseLine

A similarity metric that is utilized in our KNNBaseLine Algorithm, computes correlation constants with Pearson correlation among all user or item pairs with baseline method.

3.5.2 Cosine

A similarity metric that is utilized in our KNNWithMeans Algorithm for collaborative filtering, calculates one common user or item pairs at a time by using cosine similarity.

3.6 Cross Validators

Two cross validators are implemented to evaluate our recommendation system models.

3.6.1 K-fold CV

A validation technique that is responsible for splitting training data in no of K-folds and test and validate predictive models. In our recommendation system, we use each fold to train the recommendation system independently and then measure the accuracy against the test set.

3.6.2 LOOCV

LOO(Leave One Out) CV is a tricky cross-validation technique helps to compute Top-N recommendation for each user in our training data by removing one of the user's item intentionally from user training data, so when our recommender system test that item it is left out. In this way, we can measure or test our recommendation system able to test the top-N list for the user that is left out from our training data.

3.7 Accuracy Measures

The following are the models' accuracy measures used in model evaluation for testing accuracy scores.

3.7.1 RMSE

RMSE (Root Mean Square Error) is an accuracy measure that computes the square difference between the values of the predicted model and actual model values. It then takes the average of these values and measures the root mean square of it. The lower score of RMSE is better for good accuracy.

3.7.2 MAE

MAE (Mean Absolute Error) is an accuracy measure to calculates average magnitude values of each prediction errors in test data. Prediction errors define the difference between actual and predicted values. A lower score of MAE shows good results and accuracy.

4 Design Specification

The purpose of this section is to present the proposed Architecture of our recommendation engine. The basic Recommendation system consists of two basic recommendation techniques, Content-based filtering, and User-based collaborating filtering in Figure 1. With the help of these designs specification, the implementation of our Recommendation engine techniques were possible. The Implementation section is discussed in the next section.

4.1 Proposed Content based Filtering Technique

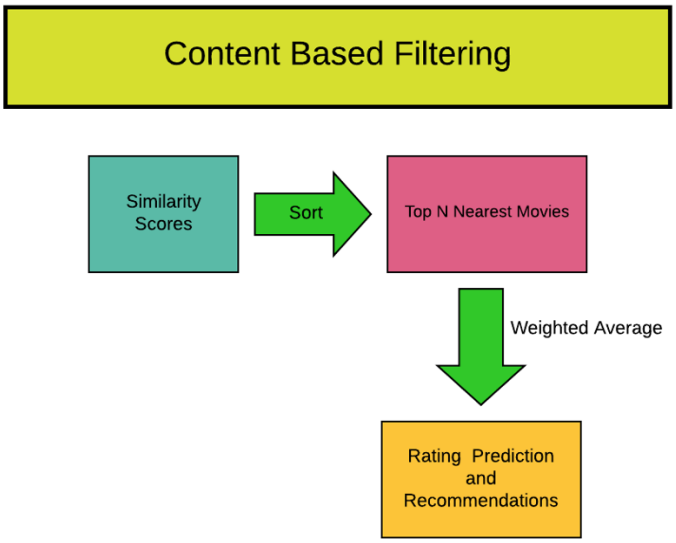

Figure 4: Content based Filtering Architecture

In the above-illustrated Figure 4, the proposed architecture of Content-based filtering can be seen. Content-based filtering-based recommender system focuses on attributes of items i.e. in our case Genres and years of movies by finding similarities between those item attributes which each user liked to provide a recommendation at the end. The process starts measuring content-based similarity between attributes of the movies and the user rating then our Nearest Neighbourhood algorithm will sort Top N (N could be any value of K) nearest neighbors to the movies and by finding those neighbors with the highest similarity score and turn them into actual prediction by computing the weighted average.

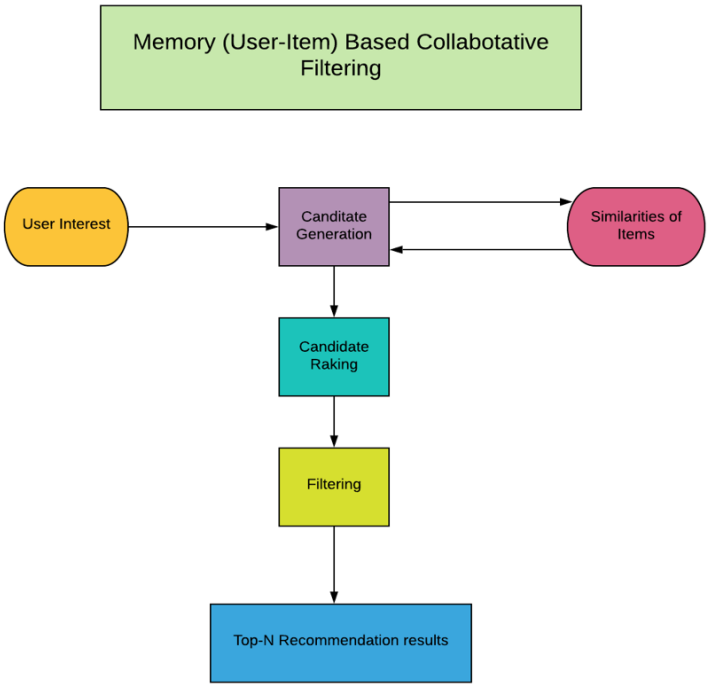

Figure 5: Memory(User-Item)based Collaborative Filtering Architecture

4.2 Proposed Memory based Filtering Technique

In Figure 5, there is design architecture of Memory-based collaborative filtering. Memory-based collaborative filtering further contains two more types.

- User-based Collaborative Filtering
- Item-based Collaborative Filtering

Both collaborative techniques are based on figure architecture, The step involved in User-based collaborative filtering are User-Item rated matrix, User Similarity Matrix(lookup Similar User), Candidate Generation(by pooling together all of the stuff those similar users rated), Candidate Ranking(by looking at how the similar users rated them) and Filter out already watched items by the specific users and recommend it Top-10 nearest neighbors movies recommendation which he or she has not watched yet. Item-based collaborative filtering is the same as user-based by just replacing the User Similarity Matrix(lookup Similar User) step with Item Similarity Matrix(lookup Similar Item).

4.3 Proposed Matrix Factorization Technique

$$R = U \sum M^T \tag{1}$$

Matrix factorization which is also known as the Model-based collaborative filtering technique is one of the useful techniques in recommendation systems. The above equation (1) explains that R is our Matrix factorization matrix which is equal to dot product ($\sum$) of two matrices i.e. U and M^T.Here U and M^T are typical User and Movie matrix which

are created by implementing SVD and SVDpp algorithms on the original User-Movie matrix. So, by taking the dot product of the typical User matrix and typical Movies matrix and get the rating for every combination of users and items, latent features are then gathered with that dot product in our matrix factorization matrix through which prediction of a rating for a specific user and item is possible to recommend Top-N movies as the final result.

5 Implementation

Figure 6 below illustrates the implementation flow of our recommendation engine. The steps involved in this flow are Data Collection, Data Analysis, and pre-processing. Recommendation Techniques with machine learning Algorithms, Models Hyperparameter Tuning, and Evaluation. All coding is done in a python programming language by using Jupyter Notebook as an IDE.

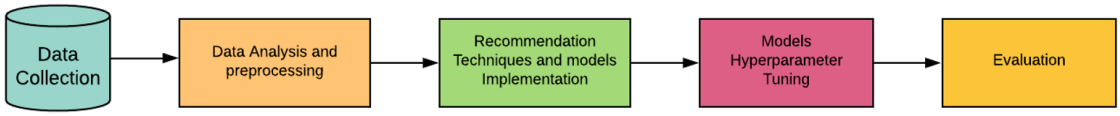

Figure 6: Implementation Flow of the Research

5.1 Data Collection:

Data is collected from the MovieLens website which contains 4 CSV files in a folder, but the project utilizes only two of them i.e. 'movies.csv' and 'ratings.csv' as per requirement. Files opened with the help of the python pandas library.

5.2 Data Analysis and Data Pre-processing

Data Analysis and pre-processing on the MovieLens dataset involved steps of data cleaning by checking the null values, merging of data tables i.e. movies and ratings on movies id as an inner join, Describing Columns, Removal of unnecessary features, Genre column that describes movie genre in the movie table are separated with multiple pipes in a single cell for each movie, A separate function (count_word) is created to remove those pipes and separate genre from every single cell. The function not just only splits genres from genre column, it also counts how many numbers of times each genre appears. For initial data visualization, the merged dataset and count_word function helped to create a word cloud and histogram that illustrates the most popular movie genre. Furthermore, with ratings count for each movie title Top 25 movies with the highest ratings are plotted.

5.3 Implementation of Recommendation System Techniques

Three recommendation system techniques have been implemented in our recommendation engine Nearest Neighbourhood Content-based filtering technique, Nearest Neighbourhood

User – Item Collaborative filtering technique, and Matrix factorization. Mostly Surprise library is used for the implementation of Recommendation system-based machine learning models. Surprise library is an official Recommender system library that contains machine learning algorithms like KNNBaseLine, KNNWithMeans, KNNBasic, SVD, SVDpp.

5.3.1 MovieLens Dataset Class

A separate MovieLens dataset class is created to utilize it in our proposed recommendation techniques. The MovieLens class loads both movies and rating datasets and to define relative functions or methods. A method named "loadMovieLensLatestSmall(self)" is responsible for building and returning a rating dataset that contains four columns i.e. user, item, rating, and time stamp. This function also converts movie id to its title and vice versa. Moreover, one more function" getMovieName" is designed to fetch movie name directly while just pushing the movie id in its parameter.

5.3.2 Nearest Neighbourhood Content Based Filtering Technique

Content-based filtering-based recommender system focus on attributes of items i.e. in our case Genres and years of movies by finding similarities between those item attributes to provide a recommendation. Recommendation action becomes more accurate if the user provides more input. In our Content-based recommendation engine, we are finding the nearest neighbors of the specific movie and recommending Top – 10 movies as a final result. These recommended movies are based on similar genre and similar years to the movies which each user liked. For similarity measure between the content similarities, the recommendation engine uses the pearson_baseline similarity metric for computing similarity along with the KNNBaseLine machine learning Algorithm for resulting top-10 nearest neighbors for that specific movie.

5.3.3 Nearest Neighbourhood Collaborative Filtering Technique

By using this recommendation technique our recommendation engine produces recommendations based on the attitude or Knowledge of the user toward the items. If two users have rated the same item which is common in both so, the system predicts that both have similar tastes of interest. There are two types of collaborative filtering.

5.3.4 Memory Based Collaborative Filtering (User-Item based Collaborative Filtering)

Memory-based collaborative filtering is also divided into two parts i.e. Item-based collaborative filtering and Item-based Collaborative filtering. In our recommendation system, we are using both Memory-based collaborative filtering. In our User and Item-based collaborative filtering implementation, we start by loading up data from MovieLens class, Train, and test that loaded data. Initiate our KNNWithMeans model along with the cosine similarity measure that computes similarities. Starting the process by building a lookup table of the user to all of the items they have rated and by finding those items and user system builds a 2D matrix of similarity scores between every pair of users. After that our engine is ready to provide a recommendation to any user, but the target is top lookup top similar users to that users so next, generate recommendation candidates by

pooling together all of the stuff those similar users rated. We then score all of those candidates by looking at how the similar users rated them, how similar the user rating them was to the user. Finally, by filtering out the already watched item we recommend Top-10 nearest neighbors' movies results for those specific users. In our item-based collaborative filtering, practically the phenomena are the same as user-based the difference is that the engine flips the matrix where items are rows and users are columns so that our similarity matrix will measure the similarity between item based on the user that rated those items and compute similarity scores between every possible item pair based on the users' item has in common who rated them.

5.3.5 Model Based Collaborative Filtering (Matrix Factorization)

One of the most popular techniques in model-based collaborative filtering is Matrix Factorization. Matrix factorization is the most used Model-based collaborative filtering technique applied to discovers each user and item interaction and rating prediction based on inner latent factors product. It is too native on each latent factor. In our implementation of matrix factorization, A User-Movie matrix is created and with Using SVD(Singular Value Decomposition) and SVDpp(Singular Value Decomposition plus plus) Algorithm that works on PCA(Principal Component Analysis) phenomena by finding those latent factors and extract them from our Movie and User Matrix data.SVD matrix factorization algorithm distills those latent features into the N-Space dimension and produces a typical User matrix and typical Movie matrix. In our assumption typical User matrix consist users as row and genres as the latent features find by SVD algorithm while typical movie matrix consist movies as row and genres as the latent features, the gathering of latent factors happens by transposing original User-Movie matrix where Movies are rows and Users are columns and matrix is based on ratings given by users to those movie titles. Training data is described in terms of smaller matrices that the factors of rating we want to predict. So, in general, SVD and SVDpp construct our Matrix factorization by taking the dot product of the typical User matrix and typical Movies matrix and get the rating for every combination of users and items. By gathering these latent factors, prediction of a rating for a specific user and item is possible by taking the dot product of the associated row in a typical user matrix for the user, and the associated column for the item and recommending Top-N movies accordingly.

5.4 Models Hyper-parameters Tuning

Hyperparameter tuning is a process of searching the best parameters for the models to obtain the best optimal results. For the hyperparameter tuning of our recommendation engine models, we have considered the Grid Search CV technique to find the best parameters for the models.

- For Content based filtering technique we are using KNNBaseLine Algorithm, The best parameters find by Grid Search CV for KNNBaseLine algo are ('k': 30, 'bsl_options': 'method': 'sgd', 'n_epochs': 5, 'learning rate': 0.00005, 'sim_options': 'name': 'pearson_baseline', 'user_based': False) and best RMSE value is 0.90.
- For User based Collaborative filtering technique we are using KNNWithMeans Algorithm, the best parameters find by Grid Search CV for KNNWithMeans algo are ('k': 50, 'n_epochs': 1, 'learning rate': 0.00001, 'sim_options': 'name': 'cosine', 'min_support': 1, 'user_based': True) best RMSE value is 0.92.

- For Item based Collaborative filtering technique we are using KNNWithMeans Algorithm, the best parameters find by Grid Search CV for KNNWithMeans algo are ('k': 30, 'n_epochs': 1, 'learning rate': 0.00005, 'sim_options': 'name': 'cosine', 'min_support': 1, 'user_based': False) best RMSE value is 0.91.

- For Matrix Factorization technique we have utilized two Algorithms i.e. SVD and SVDpp, the best parameters search by Grid Search CV for SVD and SVDpp algorithms are 'n_factors': 40, 'n_epochs': 40, 'lr_all': 0.008, 'reg_all': 0.1 with best RMSE score of 0.88(SVD) and 0.87(SVDpp).

With the help of these Models Hyperparameter tuning result, the Evaluation of the model are validated which is discussed in the next section.

6 Evaluation

For the evaluation purpose, the accuracy values of different machine learning models are validated with two cross-validation techniques i.e. K-fold Cross Validator and LOO(Leave One Out) cross validator. Two accuracy measures i.e. RMSE(ROOT Mean Square Error) and MAE(Mean Absolute Error) are utilized for evaluation purposes. Five Machine learning algorithms (KNNBaseline, KNNWithMean(Ub), KNNWithMean(Ib), SVD, SVDpp) accuracies have been validated with 3 folds from each validator.

Models	**RMSE**	**MAE**
KNNBaseLine(CB)	89%	67%
KNNWithMean(Ub)	92%	71.2%
KNNWithMean(Ib)	93%	71%
SVD(MF)	90%	69%
SVDpp(MF)	88%	66%

Table 3: K-Fold Cross validation Mean Accuracy Results (Lower is better)

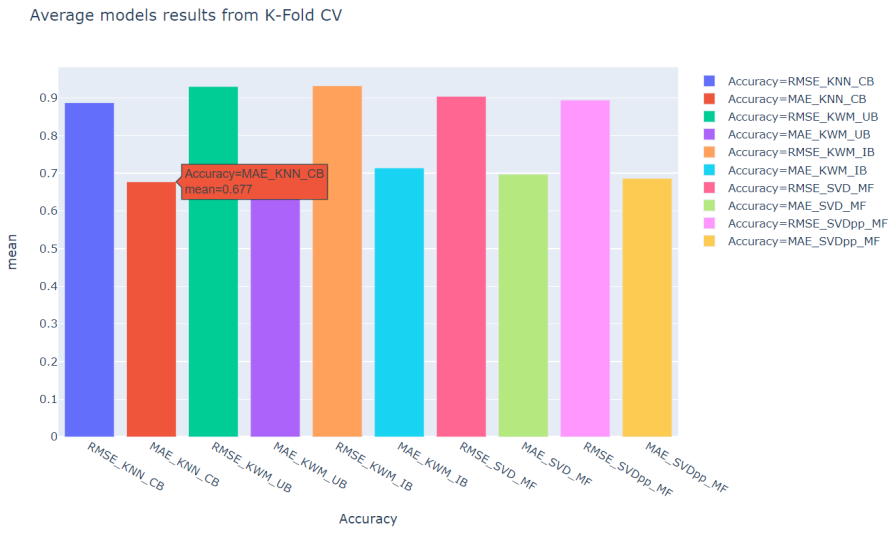

Figure 7: K-Fold Cross validation Mean Accuracy Results (Lower is better)

Figure 7 represents K-fold cross-validation mean accuracy results of models used in our recommendation engine. The lower value of RMSE and MAE are considered to be good accuracy model. From Table 3 it can be seen that The best RMSE and MAE scores are attained by SVDpp used in Matrix factorization technique i.e. 88%(RMSE) and 66%(MAE) followed by KNNBaseline SVD, KNNWithMean(Ib) and KNNWithMean(Ub).

Models	RMSE	MAE
KNNBaseLine(CB)	91%	69%
KNNWithMean(Ub)	94%	71%
KNNWithMean(Ib)	93%	70%
SVD(MF)	92%	69.6%
SVDpp(MF)	91%	69%

Table 4: LOO(Leave One Out) Cross validation Mean Accuracy Results (Lower is better)

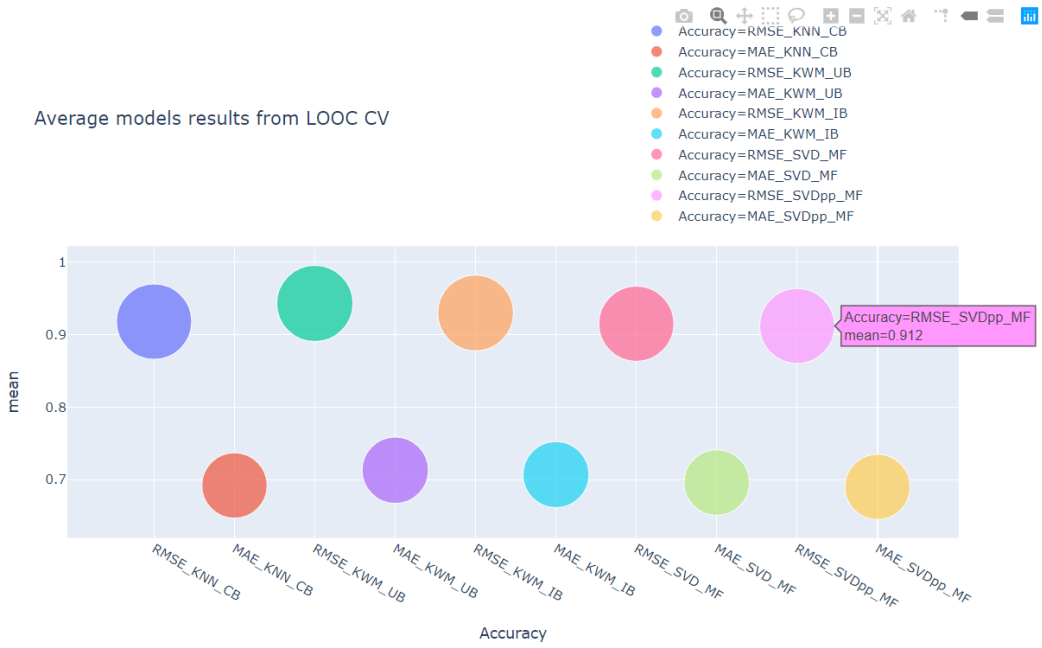

Figure 8: LOO(Leave One Out) Cross validation Mean Accuracy Results (Lower is better)

Furthermore, Figure 8 illustrates Leave one out cross-validation mean accuracy results of models. From Table 4 it can be seen that The best RMSE and MAE scores are obtained by SVDpp used in Matrix factorization technique and KNNBaseline Content-based filtering technique i.e. 91%(RMSE) and 69%(MAE) followed by SVD, KNNWithMean(Ib) and KNNWithMean(Ub).

6.1 Experiment with KNN BaseLine Algorithm by using K-Fold and LOOCV

Table 5 describes 3 cross-validation results for the KNNBaseLine algorithm with is used in our (Content-based filtering recommendation technique) by using two Cross validator

Folds	K-Fold(RMSE)	K-Fold(MAE)	LOOCV(RMSE)	LOOCV(MAE)
Fold 1	0.88	0.67	0.93	0.70
Fold 2	0.88	0.67	0.90	0.67
Fold 3	0.88	0.67	0.91	0.69

Table 5: CV Results(Lower is better) for KNNBaseLine Algorithm

name K-Fold and LOOCV along with two accuracy measures RMSE and MAE, The best RMSE score achieved from K-Fold CV fold is 0.88 which is almost equal in each fold and best MAE is 0.67 which is also similar with each fold in Figure 9. In Figure 10 Leave one out Cross-Validation,the good RMSE and MAE score is achieved Fold 2 i.e. 0.90 and 0.67.

6.2 Experiment with KNNWithMeans Algorithm for User Based Collaborative Filtering by using K-Fold and LOOCV

Folds	K-Fold(RMSE)	K-Fold(MAE)	LOOCV(RMSE)	LOOCV(MAE)
Fold 1	0.92	0.70	0.95	0.72
Fold 2	0.928	0.71	0.95	0.71
Fold 3	0.93	0.71	0.92	0.70

Table 6: CV Results(Lower is better) for KNNWithMeans(UB)

Table 6 shows 3 cross-validation results for KNNWithMeans(UB) algorithm with is used in our (User-based Collaborative filtering recommendation technique) by using two Cross validator name K-Fold and LOOCV along with two accuracy measures RMSE and MAE, The best RMSE and MAE score achieved from K-Fold CV fold is 0.92 and 0.70 in fold 1 followed by fold 2 and fold 3 in Figure 9. Whereas Leave one out Cross-Validation shows good RMSE and MAE score in Fold 3 i.e. 0.92 and MAE value of 0.70 shown in Figure 10.

6.3 Experiment with KNNWithMeans Algorithm for Item Based Collaborative Filtering by using K-Fold and LOOCV

Folds	K-Fold(RMSE)	K-Fold(MAE)	LOOCV(RMSE)	LOOCV(MAE)
Fold 1	0.92	0.71	0.94	0.71
Fold 2	0.93	0.71	0.92	0.70
Fold 3	0.93	0.71	0.92	0.70

Table 7: CV Results(Lower is better) for KNNWithMeans(IB)

Table 7 defines 3 cross-validation results for KNNWithMeans(IB) algorithm with is used in our (Item-based Collaborative filtering recommendation technique) by using two

Cross validator name K-Fold and LOOCV along with two accuracy measures RMSE and MAE, The best RMSE and MAE score achieved from K-Fold CV fold is 0.92 and 0.71 in fold 1 followed by fold 2 and fold 3 in Figure 9. Whereas Leave one out Cross-Validation presents better RMSE and MAE score in Fold 3 and 2 i.e. 0.92 and MAE value of 0.70 in Figure 10.

6.4 Experiment with SVD(Singular Value Decomposition) Algorithm for Matrix Factorization by using K-Fold and LOOCV

Folds	K-Fold(RMSE)	K-Fold(MAE)	LOOCV(RMSE)	LOOCV(MAE)
Fold 1	0.90	0.69	0.90	0.70
Fold 2	0.90	0.69	0.93	0.70
Fold 3	0.90	0.70	0.90	0.68

Table 8: CV Results(Lower is better) for SVD Matrix Factorization

Table 8 explains 3 cross-validation results for SVD(Singular Value Decomposition) algorithm with is used in our (Matrix Factorization recommendation technique) by using two Cross validator name K-Fold and LOOCV along with two accuracy measures RMSE and MAE, The best RMSE and MAE score achieved from K-Fold CV fold is 0.90 and 0.69 in fold 1 and 2 followed by fold 1 Figure 9. On the other hand, Leave one out Cross-Validation show some variations in results that cause better RMSE score in Fold 1 and 3 i.e. 0.90 and MAE value of 0.68 in fold 3 in Figure 10.

6.5 Experiment with SVDpp(Singular Value Decomposition plus plus) Algorithm for Matrix Factorization by using K-Fold and LOOCV

Folds	K-Fold(RMSE)	K-Fold(MAE)	LOOCV(RMSE)	LOOCV(MAE)
Fold 1	0.88	0.67	0.91	0.70
Fold 2	0.89	0.68	0.92	0.69
Fold 3	0.88	0.67	0.89	0.68

Table 9: CV Results(Lower is better) for SVDpp Matrix Factorization

Table 9 represent 3 cross-validation results for SVD(Singular Value Decomposition plus plus) algorithm with is used in our (Matrix Factorization recommendation technique) by using two Cross-Validators named K-Fold and LOOCV along with two accuracy measures RMSE and MAE, The best RMSE and MAE score achieved from K-Fold CV fold is 0.88 and 0.66 almost in fold 1 and 3 in Figure 9. In contrast to that, Leave One Out Cross-Validation depicts some variations in results that cause better RMSE and MAE in Fold 3 i.e. 0.89 and 0.68 followed by the good score in fold 1 and 2 in Figure 10.

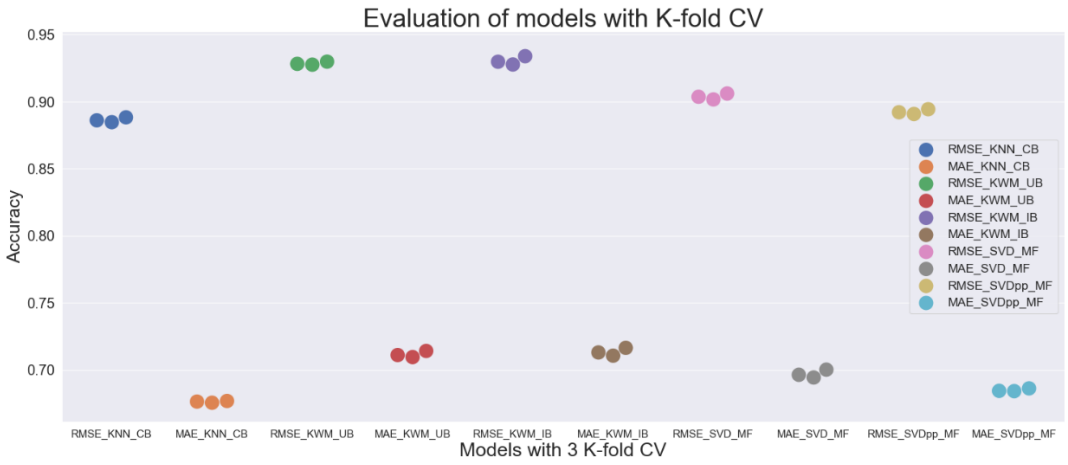

Figure 9: Evaluation of models K-Fold CV

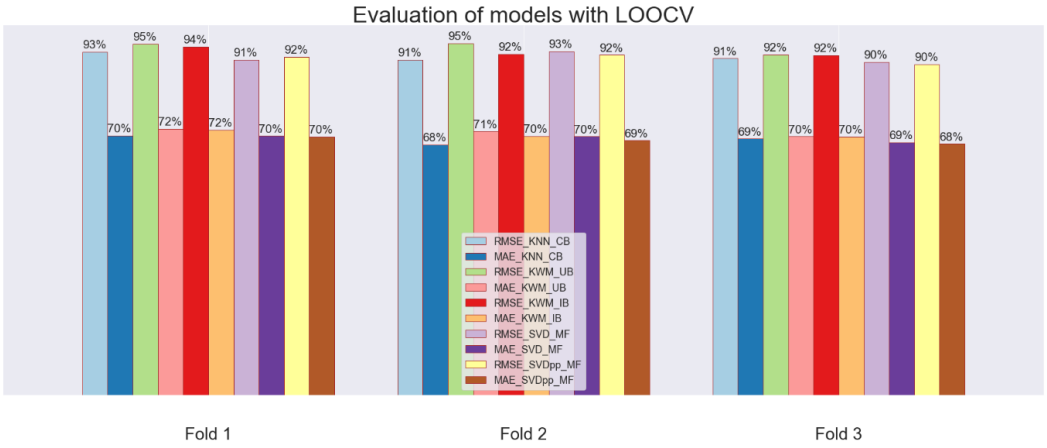

Figure 10: Evaluation of models with LOOCV

6.6 Discussion

After Experimenting the findings from different machine learning models of our recommendation engine with 2 cross-validation techniques, the discussion concludes that both cross validators have presented the same model with optimal accuracy i.e. SVDpp (utilized in Matrix factorization technique) with 88%(RMSE) and 66%(MAE) from K-Fold CV and 91%(RMSE) and 69%(MAE) from LOOCV followed by KNNBaseLine, SVD, KNNWithMeans(Ib) and KNNWithMeans(Ub) but K-fold cv folds results are 2% more accurate in comparison to LOOCV. Moreover, the top-10 recommendation results of movies for the specific user were more accurate on basis of prediction and similarity scores from SVDpp as compare to the rest of the models.

(Li et al.; 2020) research movie rating prediction algorithms find the best RMSE value of 0.85 and 0.64 MAE value by using the SVD algorithm for a group of ratings from the MovieLens dataset. In contrast to that, our recommendation system presents the best RMSE score of (0.88) and MAE(0.67) which is a bit low as compared to (Li, Zhao,

Wang and Yu, 2020) results, the reason might be the use of large data, more deep data prepossessing to remove users with more aggressive movies rating and more parameters tuning with more folds of cross-validation.

(Ahuja et al.; 2019) explains K-mean Clustering and K-Nearest neighbor algorithms have been utilized in content and collaborative based filtering of recommendation systems. Movies are clustered in a way that is similar to each other with other clusters along with KNN to obtain the best optimal results. The best RMSE value achieved is 1.081 whereas our recommendation system produces the best collaborative filtering RMSE score of 0.92

(Venil et al.; 2019) compares results for user-based collaborative filtering, content-based filtering, and Ensemble-based collaborative filtering technique for the Nearest neighborhood-based movie recommendation system on the MovieLens dataset. By testing the performance of the recommender model, Ensemble-based KNN shows better RMSE (1.07) and MAE (0.85) as compared to the rest of the techniques however our recommendation engine gives the best RMSE(0.92) and MAE(0.70) while using KNNWithMeans for Item-based collaborative filtering with cosine similarity and best RMSE(0.93) and MAE(0.70) score while using KNNWithMeans for Item-based collaborative filtering with cosine similarity metric.RMSE score for content-based filtering gets 0.89 with the KNNBaseLine Algorithm that uses baseline as similarity metrics.

7 Conclusion and Future Work

This research project focuses on building a TOP-N Nearest Neighbourhood Based Movie Recommender System by implementing different recommendation techniques. From Data Analysis to Model Evaluation, the research contains multiple sections like literature reviews, Design Specifications, Methodology, Implementation, and Recommendation machine learning models evaluation.

So, it is concluded that the accuracy values of different machine learning models are validated with two cross-validation techniques i.e. K-fold Cross Validator and LOO(Leave One Out) cross validator. Two accuracy measures i.e. RMSE(ROOT Mean Square Error) and MAE(Mean Absolute Error) are utilized for evaluation purposes. From the K-Fold cross validator mean accuracy results (Lower Values are better), the best RMSE and MAE scores are attained by SVDpp used in Matrix factorization technique i.e. 88%(RMSE) and 66%(MAE) followed by KNNBaseline SVD, KNNWithMean(Ib) and KNNWithMean(Ub). While by implementing LOO(Leave One Out)CV, the best RMSE, and MAE scores are obtained by SVDpp used in Matrix factorization technique and KNNBaseline Content-based filtering technique i.e. 91%(RMSE) and 69%(MAE) followed by SVD, KNNWithMean(Ib) and KNNWithMean(Ub) but it is found that K-fold CV folds results are 2% more accurate in comparison to LOOCV.

In term of issues with the real-world challenge of recommendation system, there are problems of data sparsity, Cold start, StopList, Random Exploration, Filter bubble, Trust and outliers, etc.

For Future improvement in this recommendation engine, Implementation of deep learning model(Restricted Boltzmann Machine and Auto Encoders), Hybrid Filtering technique and increment no of iteration on different recommendation machine learning models can improve the RMSE and MAE score for better accuracy results and final recommendation. Further by scaling up the project on the cloud services like AWS SageMake can open chances to utilize big data.

Acknowledgement

Foremost, I am so thankful and praise to the Almighty Allah, The Prophet Muhammad (S.A.W), and Ahle-Bait(A.S) for blessing me much more than I deserve and giving me the strength, ability to learn, understand, and implement. Secondly, my sincere and profound gratitude goes towards my parents(Arif and Rehana), my wife(Kiran Imran), my brothers(Adnan, Irfan, and Salman), and my Maternal Uncle(Nadeem Khan) who always stood by me throughout my career journey and critical times. Their immense support and motivation never let me gave up, but instead encouraged me to achieve this milestone. I would also like to express personal appreciation for my research project Professor Dr. Muhammad Iqbal for his assistance, support, guidance, and feedback in the complex areas of the research project sections. Finally, I would like to say that, the entire Master's journey required a lot of intense hard work and dedication towards the study. The 13 weeks of the final thesis project were exceptionally challenging and thrilling but with proper consultancy and support from my mentors, family and friends make it possible.

References

Afoudi, Y., Lazaar, M. and Al Achhab, M. (2018). Collaborative filtering recommender system, *International Conference on Advanced Intelligent Systems for Sustainable Development*, Springer, pp. 332–345.

Ahuja, R., Solanki, A. and Nayyar, A. (2019). Movie recommender system using k-means clustering and k-nearest neighbor, *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, IEEE, pp. 263–268.

Ariff, N. M., Bakar, M. A. A. and Rahim, N. F. (2018). Comparison between content-based and collaborative filtering recommendation system for movie suggestions, *AIP Conference Proceedings*, Vol. 2013, AIP Publishing LLC, p. 020057.

Bodhankar, P. A., Nasare, R. K. and Yenurkar, G. (2019). Designing a sales prediction model in tourism industry and hotel recommendation based on hybrid recommendation, *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, IEEE, pp. 1224–1228.

Çano, E. and Morisio, M. (2017). Hybrid recommender systems: A systematic literature review, *Intelligent Data Analysis* **21**(6): 1487–1524.

Dong, Y., Liu, S. and Chai, J. (2016). Research of hybrid collaborative filtering algorithm based on news recommendation, *2016 9th international congress on image and signal processing, biomedical engineering and informatics (CISP-BMEI)*, IEEE, pp. 898–902.

Fathan, G., Adji, T. B. and Ferdiana, R. (2018). Impact of matrix factorization and regularization hyperparameter on a recommender system for movies, *Proceeding of the Electrical Engineering Computer Science and Informatics* **5**(1): 113–116.

Harper, F. M. and Konstan, J. A. (2015). The movielens datasets: History and context, *Acm transactions on interactive intelligent systems (tiis)* **5**(4): 1–19.

Isinkaye, F., Folajimi, Y. and Ojokoh, B. (2015). Recommendation systems: Principles, methods and evaluation, *Egyptian Informatics Journal* **16**(3): 261–273.

Kane, F. (2018). *Building Recommender Systems with Machine Learning and AI: Help people discover new products and content with deep learning, neural networks, and machine learning recommendations.*, Independently published.

Koren, Y., Bell, R. and Volinsky, C. (2009). Matrix factorization techniques for recommender systems, *Computer* **42**(8): 30–37.

Li, X., Zhao, H., Wang, Z. and Yu, Z. (2020). Research on movie rating prediction algorithms, *2020 5th IEEE International Conference on Big Data Analytics (ICBDA)*, IEEE, pp. 121–125.

Liu, J., Wu, C. and Liu, W. (2013). Bayesian probabilistic matrix factorization with social relations and item contents for recommendation, *Decision Support Systems* **55**(3): 838–850.

Malik, S. and Bansal, M. (n.d.). Recommendation system: Techniques and issues.

Nilesh, N., Kumari, M., Hazarika, P. and Raman, V. (2019). Recommendation of indian cuisine recipes based on ingredients, *2019 IEEE 35th International Conference on Data Engineering Workshops (ICDEW)*, IEEE, pp. 96–99.

Pal, A., Parhi, P. and Aggarwal, M. (2017). An improved content based collaborative filtering algorithm for movie recommendations, *2017 tenth international conference on contemporary computing (IC3)*, IEEE, pp. 1–3.

Rajarajeswari, S., Naik, S., Srikant, S., Prakash, M. S. and Uday, P. (2019). Movie recommendation system, *Emerging Research in Computing, Information, Communication and Applications*, Springer, pp. 329–340.

Richter, F. (2020). Infographic: Global expansion fuels netflix's growth.
URL: *https://www.statista.com/chart/20345/netflix-subscriber-growth-by-region/*

Sirikayon, C., Thusaranon, P. and Pongtawevirat, P. (2018). A collaborative filtering based library book recommendation system, *2018 5th International Conference on Business and Industrial Research (ICBIR)*, IEEE, pp. 106–109.

Thukral, R. and Ramesh, D. (2018). Ensemble similarity based collaborative filtering feedback: a recommender system scenario, *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, IEEE, pp. 2398–2402.

Venil, P., Vinodhini, G. and Suban, R. (2019). Performance evaluation of ensemble based collaborative filtering recommender system, *2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN)*, pp. 1–5.

Zhao, X. (2019). A study on e-commerce recommender system based on big data, *2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, IEEE, pp. 222–226.