Movie Recommendation System using Machine Learning

MSc Research Project
Data Analytics

Metehan Bereketoglu
Student ID: x22161872

School of Computing
National College of Ireland

Supervisor: Musfira Jilani

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Metehan Bereketoglu
Student ID:	x22161872
Programme:	Data Analytics
Year:	2023
Module:	MSc Research Project
Supervisor:	Musfira Jilani
Submission Due Date:	14/12/2023
Project Title:	Movie Recommendation System using Machine Learning
Word Count:	6730
Page Count:	20

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:	Metehan Bereketoglu
Date:	31st January 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	X
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	X
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.	X

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):	

Movie Recommendation System using Machine Learning

Metehan Bereketoglu
x22161872

14th December 2023

Abstract

Based on the user's past actions and preferences, recommendation systems use a variety of algorithms to provide suggestions. They lessen the amount of time users spend searching and provide suggestions for relevant products, which greatly improves their experience and happiness with online interfaces. Likewise, a movie recommendation system may help individuals discover movies they would like quickly—sometimes even without them having to search. Sites like YouTube, Netflix, etc., that have amassed substantial user data often use this approach to play back user-generated content. Using this information, they may train recommendation systems to provide users with movie suggestions based on their interests. Since users can quickly locate the best-rated films in their interest area, this not only enables them to boost viewing length and revenue without advertising, but it also increases user happiness and royalties.

1 Introduction

1.1 Research brief

Movie recommender systems is found to be an intelligent algorithm that provides recommendations for films to watch for users based on their watching preferences and past viewing activity. It seems these systems provide customized suggestions by analyzing data, including viewing histories, ratings, and reviews from users. This study is going to develop a movie recommender system using machine learning for effective user experience and better understanding about machine learning Furtado and Singh (2020). It will completely change how people find and watch movies by making it easier for consumers to browse through enormous movie libraries. It is going to practically implement machine learning in the development of movie recommendations leading to an expanded approach of artificial intelligence. It will also use movie lens data set for development of movie recommendation systems and enhance effective representation. It will also show the practical use of machine learning in movie recommendation systems by applying it and observing hyperparameter tuning as well.

1.2 Background of the research

Machine learning generally offers various services that enhance user experience and ease the working operations with cutting-edge technology to assist consumers in finding the best solution as desired. It seems people are highly confused in choosing suitable movies to watch in their busy schedule and implementation of machine learning to develop movie recommendation systems may resolve their issue and save their time with precise movies they desire. In addition to saving time spent perusing movie listings, this recommendation engine may provide more tailored results to help viewers avoid feeling overloaded with choices. It seems machine learning and its various applications can effectively be utilized to create a movie recommendation system today, including everything from user preference analysis and data collection to model training and application deployment. It can be observed with every facet of machine learning-based movie selection for the users as per their desire categorized byu movie category and ratings. On the other hand, the movie recommender system has grown more and more crucial for both individual consumers and production firms as the internet streaming market grows Surendran and Kumar (2020). Moreover, these algorithms are used to provide users with the best selections in an effort to improve their movie-watching experience. Apart from that, two main categories may be used to categorize recommendation system filtering strategies: collaborative filtering and content-based filtering. The study on using machine learning in movie recommendation systems would increase knowledge about algorithms and its application in daily life activities for people for better experience with technology.

1.3 Research aim

The study aims to develop an effective movie recommendation system using machine learning and understand its significance in the enhancement of user experience. It also aims to understand the machine-learning algorithms significance and requirement in the efficient working of recommendation engines for movie suggestions.

1.4 Research objectives

- To learn the effectiveness of different machine learning algorithms and their usage.
- To analyze the effectiveness of machine learning in development of recommendation engines to help customers in their decision making.
- To evaluate how machine-learning algorithms have been used to make an efficient recommendation engine.
- To develop the best machine learning model that provides the highest accuracy to the recommendation engine.
- To discuss problems with the last developed recommendation engine with the help of machine learning to predict movies.
- To enhance efficiency of machine learning in the recommendation engines system leading to better user experience

1.5 Research questions

1. What are different machine learning algorithms and their usage?

2. What is the significance and use of machine learning in development of recommendation engines to help customers in their decision making?

3. Why is machine learning required for the development of recommendation engines?

4. How can machine-learning algorithms be used to make an efficient recommendation engine?

5. How to develop the best machine learning model that provides the highest accuracy to the recommendation engine?

6. What are the problems with the last developed recommendation engine with the help of machine learning to predict movies?

7. How to enhance efficiency of machine learning in the recommendation engines system leading to better user experience?

1.6 Problem statement

The recommendation system is highly demanded in the recent advanced technological era of digital platforms and services. People are so highly dependent on machine learning that they require its assistance in the decision making for movies, which also influences the idea of development of movie recommendation systems using machine learning. However, due to poor implication of algorithms all existing movie recommendation systems have various errors and bugs that reduce the efficiency of the system and user experience. The movie recommendation system aims to suggest a movie to the user as per their desire Furtado and Singh (2022). It seems that giving consumers of internet service providers appropriate material selected from a library of things that are both relevant and irrelevant. The primary advantages of recommendation systems are increased income and client happiness. The movie recommendation system is a significant and very effective mechanism. However, movie recommendation systems also have concerns with scalability and low suggestion quality as a result of the limitations with the pure collaborative method. However, this study focuses on developing movie recommendation system that would provide best movie recommendation based on user demand and rating of movie using machine learning.

1.7 Research rationale

The design of the movie recommendation system is a sophisticated procedure that makes use of a number of different algorithms to propose films to consumers according to their tastes. The architecture gathers information about user behavior such as past movie choices and ratings and uses it to generate a customized list of recommendations. The algorithm utilised in the movie recommendation system forms the basis of this system. However, recommendation systems frequently utilise collaborative filtering algorithms, that examine a user's profile and behaviour in conjunction with other users' to provide recommendations. The use of hybrid recommendation systems, which integrate many algorithms to provide more accurate and customized recommendations, is growing in popularity. Generally speaking, a movie recommender system's architecture is well thought out to provide users a smooth, entertaining movie experience Keshava and Naik (2020). On the other hand, this recommendation system would help to evaluate the machine learning practice on data by the approach involving processing, best algorithm selection as well as hyperparameters tuning. In order to guarantee the validity and efficacy of the algorithms utilised by the system, industry experts strive to get certification by passing a Machine Learning certification test. With this certification, they may be considered as experts in the area of movie recommendation systems and as proof of their competency in the newest machine learning technologies and methodologies. This study would improve the application of machine learning in the recommendation system and influence various OTT platforms to use it as their marketing strategy. The study would also help the user to get the best recommendation for movies as per their categories.

1.8 Research significance

The study is focusing on the use of machine learning in the development of effective movie recommendations that would influence better understanding of algorithms and enhanced user experience. In order to model a user's tastes and interests, recommendation systems utilize machine learning algorithms to evaluate vast volumes of user data, including ratings, search queries, and previous purchases Keshava and Naik (2020). Afterwards, individualized suggestions based on the requirements and tastes of every user are produced using this approach. On the other hand, recommendation systems use a variety of algorithms to forecast user preferences based on past actions and choices. It seems that by cutting down on search time and recommending the most relevant products, they contribute to improving user pleasure and experience, particularly in online interfaces Wu and Bhandary (2018). In a similar vein, a recommendation engine for films assists users in quickly and sometimes even without having to search for the films they would find interesting. This approach is mostly used by online movie players, such as YouTube, Netflix, and others, that have gathered sufficient data on customer use history. However, with the use of this data, they may train recommendation algorithms to propose the most relevant films to their patrons. Apart from that because users can quickly locate the highest-rated films from their desired domain, this not only allows them to boost viewing duration and their revenue as a consequence without any promotion, but it also increases customer happiness as well as royalties. Therefore, the study helps to expand the application of machine learning and influence people to get better movie recommendations by recommendation system.

1.9 Summary

This section of the introduction covers a number of topics, including background of machine learning, recommendation system, problem statement, research significance and others. The aim, goals, and research questions pertaining to movie recommendation systems using machine learning have also been discussed. The primary rationale for the research is elucidated in detail to enable readers to comprehend its aim and significance about the influence of machine learning and its use in the development of movie recommendation systems. It also helps to understand the overview of the study with proper research questions, objective and problem statements regarding machine learning and development of movie recommendation systems.

2 Related Work

2.1 Introduction

Recommender systems are computer programs that provide suggestions or recommendations to users for goods or, in the case of e-commerce, products. These systems discover the relevant or desired things or products by using a suitable algorithm and the inputs of the consumers' preferences or interests. The use of machine learning in the development of the movie recommendation system would help the user to get the best recommendation for movies as per desire Furtado and Singh (2020). The goal of the movie recommendation system is to provide recommendations for movies based on user preferences. Providing acceptable content to internet service providers' customers seems to involve choosing items from a library that have content that is both relevant and unrelated. This portion is going to describe the use of NMF algorithm, slopeone algorithm, KNN with means algorithm, KNN baseline model, SVD algorithm and Hyper-parameter tuning. The literature gap is also discussed to fill in order to obtain desired outcome.

2.2 NMF algorithm

NMF makes use of techniques from linear algebra and multivariate analysis. The method repeatedly modifies A and B's values until their product becomes closer to X. This technique ensures that the weights and base are both non-negative while maintaining the original data's structure. Once the approximation errors converge or reach a predetermined number of iterations, NMF terminates. It has to be initialized using a seed so that the iterations may refer to the beginning point Sharifi and Nasiri (2014). This is because there is no global minimization procedure and the processing space is very dimensional. Therefore, obtaining meaningful outcomes may depend on proper initialization. When doing dimensionality reduction preprocessing for tasks like regression, clustering, classification, and other NMF is used.

In fact, it may be used in any circumstance when there are no negative members in the input data matrix. This approach for unsupervised learning reduces the dimensionality of data into spaces of fewer dimensions. Applications like text mining, picture analysis, and recommendation systems are where this approach is most often utilised. The most recent feature extraction technique, NMF, is helpful when there are a lot of ambiguous and poorly predictable characteristics. It may result in significant themes, subjects, and patterns.

2.3 SlopeOne algorithm

Slope One is a linear regression-based item-based collaborative filtering recommendation system. It can provide good recommendations in real time and is adaptable to data sparsity. The Slope One method can be easily implemented and extended. Numerous online recommendation systems, like the Discover MP3 and Hitflip DVD recommendation systems, employ it. Apart from that it seems one popular method in recommendation systems is collaborative filtering. One major aspect influencing collaborative filtering's prediction accuracy is data sparsity. The Slope One approach solves the data sparsity issue using a basic linear regression model. However, the K-nearest-neighbor approach may maximise the quality of evaluations provided by prediction participants when combined with user similarities. A novel collaborative filtering method that combines Slope One with unknown neighbours is described, and it is based on the Slope One algorithm. Moreover, depending on how similar one user is to other users, a different number of neighbours are dynamically chosen for each user. Second, based on assessments from nearby users, average variances between pairs of relevant objects are produced Wang and Li (2016). Finally, the linear regression model predicts the object evaluations. On the other hand, tests conducted on the MovieLens dataset demonstrate that the suggested method outperforms Slope One in terms of suggestion quality and resilience to data sparsity. In terms of prediction accuracy, it also performs better than a few other collaborative filtering methods.

The recommendation system enhanced a method for collaborative filtering that combines Slope One with unknown neighbours. Initially, we determine the number of neighbours for each user based on their similarity in order to enhance the accuracy of the rankings provided by users who take part in prediction. Second, we use the linear regression algorithm to determine the divergence between items using the ratings of the chosen neighbours Song and Wu (2020). Thirdly, it is determined the ratings based on user ratings for specific items. Finally, based on the ranking order, we choose the top k goods to suggest to the intended user. Apart from that, tests conducted on the MovieLens dataset indicate that the enhanced algorithm may contribute to a rise in recommendation accuracy. Furthermore, it is more resilient to data sparsity and leads to effective development of movie recommendation systems. By sifting through content that could fit the user's interests or preferences, recommender systems help alleviate the difficulties associated with information overload. By removing extraneous material from searches for needed information, these systems help users effectively solve problems.

2.4 K-Means clustering

According to Sinaga and Yang (2020), A popular unsupervised machine-learning technique for grouping data into distinct, non-overlapping groups or clusters is K-Means clustering. Its main goal is to cluster data points so that, compared to points in other clusters, points within the same cluster are more similar to one another. The technique operates by allocating data points to one of K clusters iteratively based on feature similarity and minimizing within-cluster variation. Here's a streamlined procedure:

Initialization: Select K randomly or deliberately chosen initial cluster centroids from the dataset.

Assignment: Determine the distance (typically in terms of Euclidean distance) between each data point and each centroid. Then, assign each data point to the closest centroid, creating K clusters.

Update centroids: Recalculate the centroids of the K clusters by taking the mean of all data points assigned to each cluster.

Reassignment: Using the new centroids, carry out the assignment step again, assigning each data point to the closest centroid.

Convergence: Until centroids no longer significantly change or a predetermined number of iterations is reached, repeat the update and reassignment procedures.

K-Means aims to minimize the sum of squared distances between data points and cluster centroids. It can, however, converge to local minima and be subject to outliers, depending on the starting centroids selected. Large datasets and spherically shaped clusters with comparable densities are ideal conditions for this method to function successfully Sinaga and Yang (2020). It's employed in a variety of applications, including customer segmentation, image compression, and document clustering. Choosing the appropriate number of clusters (K) remains difficult and frequently requires domain expertise, visualization, or approaches such as the elbow method or silhouette score to determine the best clustering solution.

2.5 KNN baseline model

According to Keshava and Naik (2020), A simple yet effective supervised machine-learning technique that may be used for regression and classification tasks is the K-Nearest Neighbors (KNN) algorithm. As a baseline model, KNN makes predictions by comparing input data points. KNN assumes that related entities are nearby. Based on a distance metric (often Euclidean distance), it chooses the K closest (nearest) data points for a new data point inside the training set.

Here's a quick breakdown:

Classification: The new data point in a classification task is given the modal (most popular) class label among its K closest neighbors. The new point is classified as Class A, for instance, if K=5 and three of its neighbors are in Class A and two are in Class B.

Regression: To forecast a continuous value for the incoming data point, KNN computes the average, or weighted average, of the goal values of the K nearest neighbors.

Hyperparameters: K, which stands for the number of neighbors to consider, is the primary hyperparameter in KNN. Selecting the right K is important since a lower K could produce a noisy forecast that is susceptible to outliers, while a higher K could cause over-smoothing and ignore local patterns.

Baseline Model: The simplicity and ease of implementation of KNN make it an ideal starting point or baseline model. For increasingly complicated algorithms, it aids in creating a performance standard.

KNN can be computationally inexpensive during prediction but possibly expensive during query time, particularly with large datasets Andersson and Tran (2020). Due to its simplicity and lack of explicit training. Its speed can also be greatly impacted by feature selection and scaling because it thinks that all features are equally relevant.

2.6 SVD algorithm

A matrix factorization technique called singular value decomposition (SVD) extends the eigen structure of the square matrix to include any matrix. If one is not familiar with eigenvectors or eigen decomposition. Apart from that Simon funk presents the SVD algorithm for the Netflix Prize competition. It is observed that by minimising the sum-squared distance, it may determine the Low-rank approximations. Using supplied rankings, SVD creates the matrix Rˆ = UT V and minimises the sum-squared distance. This approach is comparable to Probabilistic Matrix Factorization since it does not require baselines. It has been observed that the user along with item factors are randomly initialised based on a normal distribution, and the baselines are set to 0. During training, these parameters' means and standard deviations will be computed to fine-tune the system. However, the regularization term and the learning rate may be controlled by the model and initialized at the beginning of training Rajarajeswari and Uday (2019). It seems that singular value decomposition (SVD) helps to develop recommended systems as well. It is observed that the recommender system design for digital libraries is built on the content-based approach with effective use of SVD algorithm. The scourge of dimensionality, a typical issue in machine learning where algorithm performance declines as the number of distinct characteristics in the data rises, is lessened by SVD by lowering the size of the data. It is useful to find the matrix's rank, measure how sensitive a linear system is to numerical inaccuracy, or get the best lower-rank approximation possible for the matrix. The cosine of the angle that separates two vectors in an inner product space is used to calculate the cosine similarity, which is a measure of similarity.

2.7 Hyper-parameter tuning work in Machine Learning

Data scientists may optimize model performance via hyperparameter adjustment. This is a fundamental step in the machine learning process, and the right selection of hyperparameter values is critical to the outcome. Moreover, consider the scenario when one utilising the model's learning rate as a hyperparameter. Apart from that each model contains a number of parameters, and it's important to start the model with certain values when building an instance of it. The majority of time parameters are set to their default settings, which have produced the best accuracy across the majority of test datasets. However, these defaults may not function well when the data changes, which makes hyperparameter adjustment even more crucial. However, in this instance, the Grid Search Hyperparameter tweaking is done by cross validation Pouransari and Ghili (2014). Moreover, it seems that using this procedure, all defined parameters and values are obtained, and the model is run for every conceivable combination of parameters and values. Additionally, k-fold cross validation is done internally by this approach. Furthermore, after choosing new parameter values, the model is run three times with various data splits to assess evaluation metrics and provide the best model for these parameters. This is known as the "three-fold same" cross validation setting. Then, using different parameter values, compares metrics with the best model that was previously stored. It is observed that if new values result in any improvements, the grid search updates the best model. On the other hand, the final result is the parameters of the top two models with the lowest RMSE and MAE after all parameters have been checked. It is observed that model performance, functionality, and structure are all directly controlled by hyperparameters. Moreover, data scientists may optimize model performance via hyperparameter adjustment. Therefore, this is a fundamental step in the machine learning process, and the right selection of hyperparameter values is critical to the outcome.

Consider the scenario when utilising the model's learning rate as a hyperparameter. Moreover, an excessive value might cause the model to converge too rapidly and provide less-than-ideal outcomes. On the other hand, training takes too long and outcomes could not converge if the rate is too low. Accurate models and superior model performance are produced by carefully selecting and balancing the hyperparameters. However, manual or automatic hyperparameter tweaking is available. However, the advantage of manual tuning is that it helps to understand how hyperparameter weightings impact the model, even if it is slow and laborious. However, it would typically use one of the popular hyperparameter learning methods in most cases Pouransari and Ghili (2014). Apart from that, iterative hyperparameter tuning involves experimenting with various parameter and value combinations. Typically, it begins by identifying the objective variable, such as accuracy, which is wanted to enhance or reduce as the main measure. Furthermore, cross-validation approaches are a useful way to ensure that the model is not focused on just one subset of movie data.

2.8 Literature gap

The information on the machine learning and movie recommendation system are quite low in the existing literature. The information on the hyper-parameter tuning, SVD algorithm, KNN baseline model and other machine learning algorithms are quite complicated and scattered. It is useful for putting the model with the cosine similarity formula into practice and for suggesting a string that is similar to the user's selection by comparing the user's writing patterns to those of the other string that is mentioned. Furthermore in giving users movie suggestions the existing recommendation system seems to perform not well. It also seems the application of machine learning is poorly done in the recommendation system which is going to be done in this study that would assist users with best suggestions for movies.

3 Methodology

3.1 Introduction

Recommendation systems use various algorithms to predict what users prefer based on-their previous behavior and selection. They help to increase the user experience and satisfaction especially in online interfaces by reducing search time and suggesting the most related items.

Similarly, a movie recommendation system helps people to find the movies that may interested in shortest time even sometimes without searching. This system mostly utilizes for online movie players same as YouTube, Netflix etc.that have collected enough information about the past usage of users Furtado and Singh (2020). They can train recommendation algorithms with these data and suggest the most relevant movies to their customers.

This not only helps them to increase watching duration and their income as its result without any marketing but also increase satisfaction and royalty of users, because they can find the top-rated movies from their interested domain with minimum time for searching.

3.2 Dataset

Link: https://www.kaggle.com/datasets/prajitdatta/movielens-100k-dataset/data

The Movie Lens dataset is used for creating this recommendation system. It was collected by the Group Lens Research Project at the Minnesota University. This dataset is publicly available through Kaggle by this link:

It has 100000 records of users rating to 1682 movies that were collected from 943 users while each of them rated minimum 20 movies. In this site the other information such as user demographic information involve age, gender, occupation and zip code, movies characteristics include title, published year and genre are provided. The dataset cleaned by authors and the ratings of users without demographic information, invalid format of rating and also users with less than 20 ratings were deleted. The data has been collected from 19th of September 1997 to 22 of April 1998.

3.3 Data preprocessing

Removing Noise

In recommendation system, the goal is to suggest high rank movie to the audience, then the movie with a smaller number of ratings cannot help the system. If we taking them into account, it will increase the complexity of the model without any impact on the performance. Thus, in the preprocessing steps, all the movies that have ratings less than the median number of rating (27) drop from the dataset and 92323 records remain for training the system.

Splitting data into train and test

For training and testing the models the K-fold cross validation method is utilized. In this method the data is divided into K partitions and the model is trained K times while each time one of these partitions is used for testing and the remaining data is utilized for training the data. In this way, the part of data that have the best performance is chosen as final splitting and its model return as the most accurate model.

Here the 3-fold cross validation is selected which means the data is divided into 3 parts and model trains 3 times while one of these data parts is opted as test dataset. In this way each round the data division happens with around 67:33 rate.

Recommendation model

Consider to the overall architecture, to find the most suitable recommendation model, various algorithms are trained and evaluated, the best algorithm is selected and hyper parameter tuning is applied on it. The tuned model is the proposed model of this work. All these steps explained in details in following sections.

3.4 Train different algorithms

There are various algorithms that can be used to train movie rating dataset. They have different method to find the patterns and relation between data and also different data have different behavior and relation. Thus, the performance of algorithms can be different and to find the accurate algorithm for these datasets, they need to perform and choose based on the performance. Here, the Surprise library is selected for modeling recommendation system that implemented some of the algorithms and made them easy to use for training and predicting. The SVD, NMF, SlopeOne, KNNWithMeans and KNNBaselineare all the algorithms selected for this work.

3.5 SVD algorithm

The SVD algorithm is introduced by Simon Funk for the Netflix Prize challenge. It can find the Low-rank approximations by minimizing the sum-squared distance. SVD create the matrix Rˆ = UT V from given ranks and minimizes the sum-squared distance. In this algorithm the baselines are not used and make it similar to Probabilistic Matrix Factorization.

TheSVD is used the stochastic gradient descent method during training with the aim of following question to minimize the error:

The Baselines are initialized to 0 and the user and item factors are initialized randomly according to a normal distribution and will tune by calculating the mean and standard deviation of these parameters during training. The model can have control the learning rate and the regularization term and initialize them in start of training. The prediction performs by calculating following equation, if the user or items are unknown the bias of them and p (user) and q (item) factors are set to 0:

For this work, this algorithm is performed with all defaults value as the goal is to compare algorithm and select the best one them apply hyperparameter tuning for best algorithm.

3.6 NMF algorithm

It is a collaborative filtering method that works based on Matrix Factorization. This matrix is non-negative which means, does not have any negative values. NMF is used rui

= qiTpu equation for prediction which is similar to SVD prediction formula but here the user and item factors always remain positive Sharifi and Nasiri (2014).

It uses a stochastic gradient descent for optimization which is regularized with a specific number of strides that ensures to keep all the factors non-negative and also initiate them with positive values. During each strides of SGD the u , f and i which are user, factors and items respectively are updated through following formulas:

wherei and uare regularization parameters.

NMF is dependent on initial values hyper parameters which are initLow and initHigh and the user and item factorsare provided between these two values. Then these are important parameters while using this algorithm. The other important hyperparameter is bias and if it set to True, the bias also is added to the algorithm same as SVD.

Here, the parameters of the algorithm keep as default to have a fair comparison between all algorithms while selecting the best one.

3.7 SlopeOne algorithm

It is another collaborative filtering algorithm that is so simple, it uses the following equation for prediction where Ri(u) is a group of relevant items. the group of items j rated by u that has minimum one common user with i group Wang and Li (2016).

The deviation of (i,j) is defined as the meanof subtraction between the ratings of i and j.

There is no changes in default values of algorithm parameters and here the base algorithm is used to train data and the evaluation metric is calculated to have a criteria for comparing different algorithms and select the best of for these data.

3.8 K Means algorithm

The K-Means algorithm is one of the algorithms that inspired from K nearest neighborhood that is a supervised learning method and can be used for both classification and regression Sinaga and Yang (2020). It utilizes the similarity between items by considering the mean ratings of each user and groups them into a separate section.

The 3-fold cross validation method is applied while training this model, to feed the model with various data and find the best set of data for training the model that help to improve performance and reduce the error of prediction.

3.9 KNN baseline algorithm

KNNBaseline also inspired from KNN algorithm, it takes the benefit of similarity concept to predict the target value on future data that can be done by assuming the K nearest neighbors of each datapoint in training dataset.

Also, it is known as a distance-based method because calculates the distance of all points from new data and select the target value of the nearest datapoint as target of it. It utilizes the Cosine or the Pearson method to measure similarity. In addition, it has various way for computing the baseline estimation Keshava and Naik (2020).

Firstly, this model trained initiate with default values recommends by implementor of Surprise library, then as it has highest performance as compare to other models the hyper parameter tuning is performed.

3.10 Hyperparameter tuning of KNN baseline

Every model has various parameters and while creating an instance of it, it is vital to initiate model with specific values. Most of the time parameters set with default values which has achieved best accuracy on most of the test datasets. However, when the data changes these defaults may not work well and this increase the importance of Hyperparameters tuning Pouransari and Ghili (2014).

Here, The Grid Search Cross validation is used for hyper parameter tuning. This method gets all the define parameters and the values and run the model for all possible combination of values and parameters. This method also performs k-fold cross validation internally.

The cross validation set to 3-fold same which means after selecting new values for parameters, the model is run 3 times with different split of data and measure the evaluation metrics and return the best model for these parameters. Then compares metrices with best previous saved model (with other values for parameters) and if new values create any improvement, grid search updates best model. When all parameters verified, the parameters of best two models with lower RMSE and MAE are returned as its final outcome.

3.11 User interface

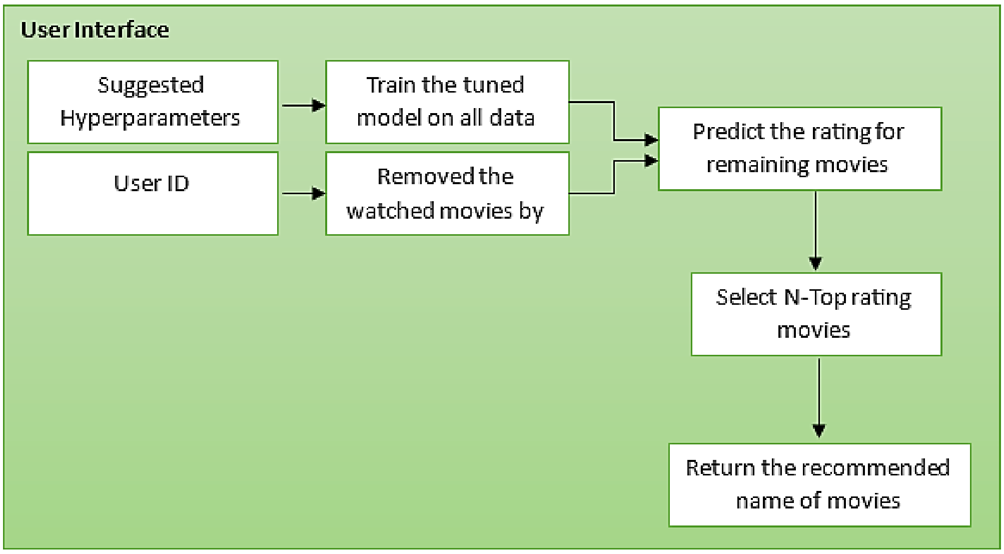

User Interface
Suggested Hyperparameters
User ID
Train the tuned model on all data
Removed the watched movies by
Predict the rating for remaining movies
Select N-Top rating movies
Return the recommended name of movies

Figure 1: User Interface

The model became ready after hyperparameter tuning and it is usable for using it to recommend new movies to the users. For this reason, a user interface designed that get the user identification number (userId) and the number of movies to recommend (n). when the user enter userid and n a function is run. The function trains the tuned model on whole dataset.

Then all the movies that were watched before by user are removed from unique list of movies, the trained model is used to predict interest rating of this user for remaining movies and a list of movie id and its predicted rate is created.

The predicted ratings are sorted and the top n movies areseparated to recommend to the user. As training dataset contains only the ID on movies, we mapped the movies id with movie characteristic dataset and extract name of the movies which will be the output of recommendation function.

3.12 Evaluation metrics

The recommendation model predicts the rating for each pair of user-movie which is a numerical and continues values and in this type of problems for evaluating the models the metrics such as RSME and MAE are used that are explained in following sections Steurer and Pfeifer (2021).

Root of Mean square Error (RMSE)

RSME measure the average distance between the predicted value and real target value on test data that is used for regression problems. It is a square root of values calculated by the mean square error formula.

RMSE = (Pi – Ri)2/ n

Where is the sum of values for all i, Pi is the predicted value for ith and Ri is the actual value for the ith user-movie rating in the dataset and n is total number

The less value of RMSE shows the more accurate model.

Mean Absolute Error (MAE) MAE is a metrics that compute the exact difference between predicted and actual values. It is another common method for measuring the performance of regression models and it should be near 0 to show the model perform well. It is calculated as the sum of absolute errors divided by total number of samples through following equation: MAE = —(Pi – Ri)— / n

3.13 Technology stack

For this project, the models are designed using the Python programming language in the Google Colab environment. The Surprise library is used for implementing and training the models, in addition other common libraries including pandas, NumPy, matplotlib, seaborn, scikit-learn are utilized for loading data and analysis them in chart format.

4 Design Specification

The proposed method is used to obtain an accurate and precise recommendation system that can facilitate finding the desirable movies for users with minimum effort and time, also increase the income of companies with least cost of marketing. The history data is used as inputs of recommendation algorithms and for better performance cleaning and removing noises is done as first step, then different algorithms are trained and the best one is chosen as final algorithm. After that to improve accuracy of selected method and tune it, the hyperparameter tuning method is utilized. Finally, the tuned model is used to recommend movies to the specific users. The overall architecture of the model is denoted in following-

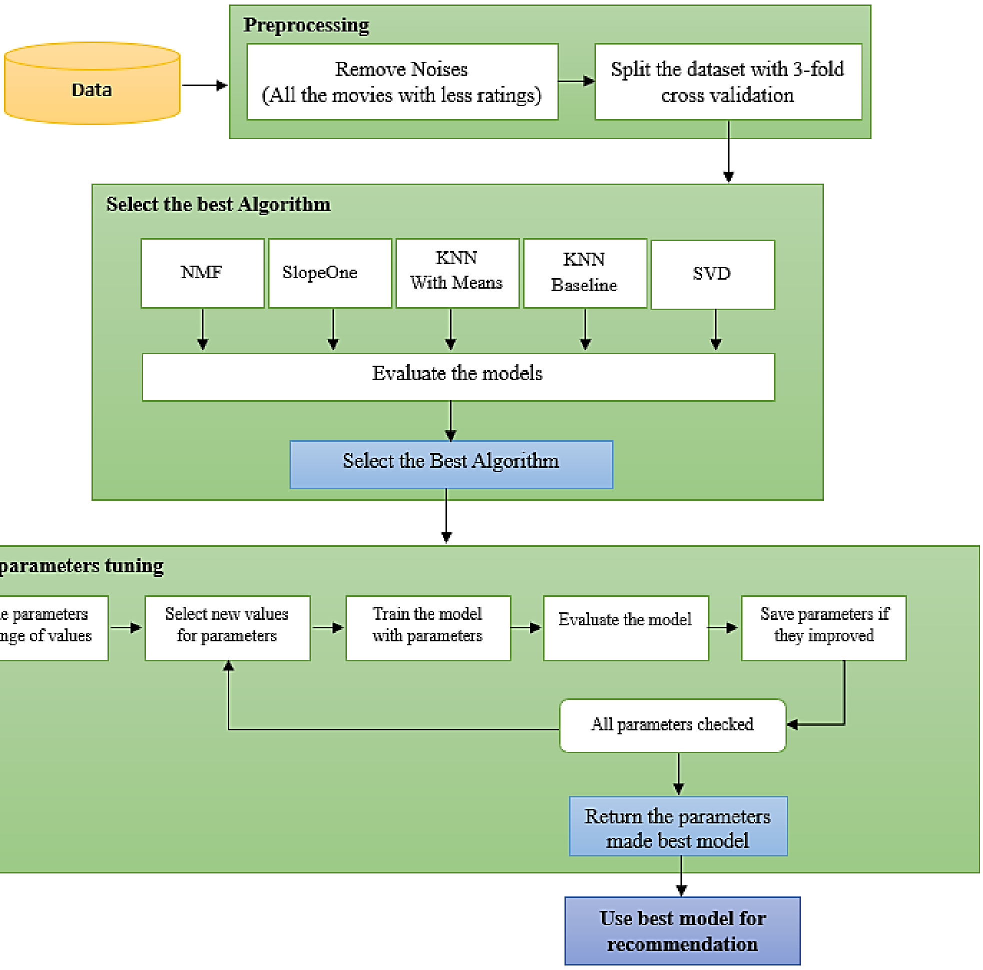

Data
Preprocessing
Remove Noises
(All the movies with less ratings)
Split the dataset with 3-fold cross validation
Select the best Algorithm
NMF
SlopeOne
KNN With Means
KNN Baseline
SVD
Evaluate the models
Select the Best Algorithm
Hyperparameters tuning
Define parameters and range of values
Select new values for parameters
Train the model with parameters
Evaluate the model
Save parameters if they improved
All parameters checked
Return the parameters made best model
Use best model for recommendation

Figure 2: Overall Architecture of the proposed model

5 Implementation

In this chapter the data analysis and performance metrics of implemented models is denoted. In three different section that are data analysis, result of training different models and last section performance of hyper parameter tuning.

5.1 Dataset characteristics

The dataset contains 100000 records that 943 users rating the total of 1682 movies ad each of user did not rate minimum 20 movies. The dataset includes 3 variables that are user id, movie id and rating f user for that movie. The second dataset has demographic information of 943 unique users that participating in rating movies and their rating accepted.

5.2 Analysis and distribution of data

Before creating suggesting model, some data analyzing was dome on data that will explain in coming section. The rating value is between [1:5] and graph of rating present, most of the rating values by users are between 3 and 4 and after them 5 has the highest frequency.

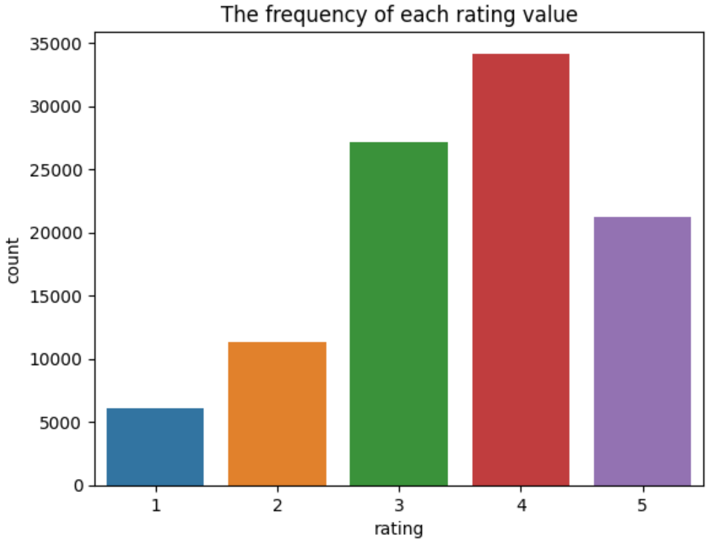


Figure 3: Frequency of rating values

As most of the users were between 15 to 55 years old. This shows the elder people are less attracted to watching movie.

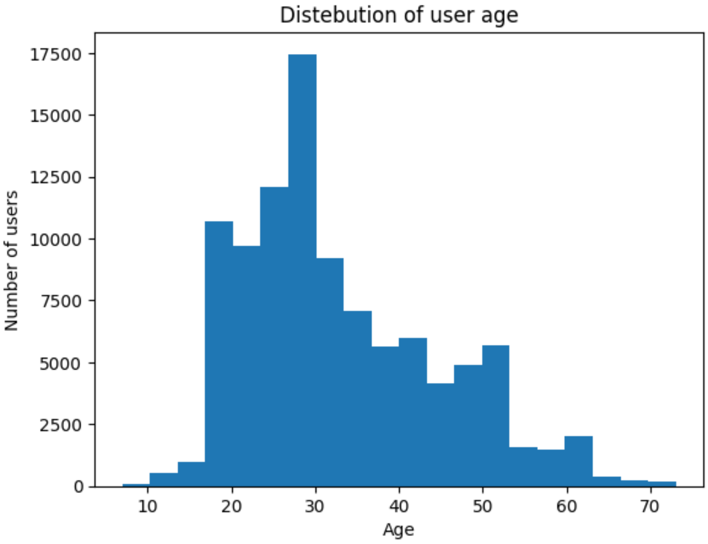


Figure 4: Distribution of users per age

The other factor is gender of user, based on following figure, mostly men were participated in this survey as compare to women. But the flow of rating has same behavior among men and women.

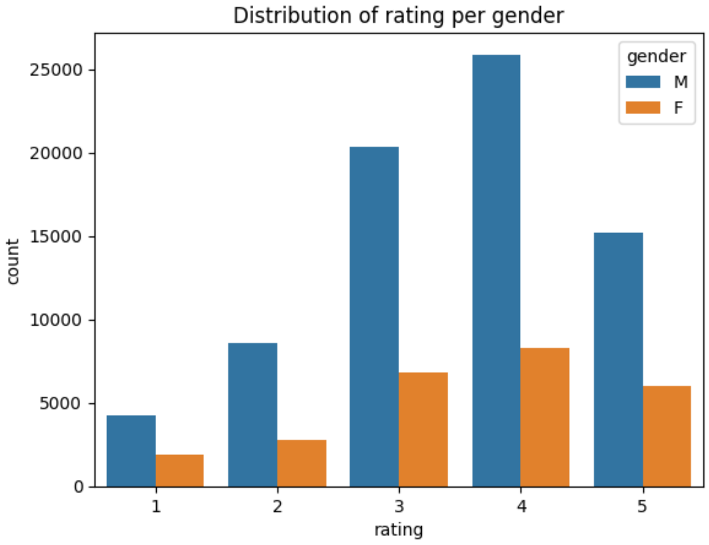


Figure 5: Distribution of rating per gender

The relation between occupation and rating is presented in following chart and shows, most of the customers are student because maybe they have more free time as compare to adult. After them the engineer, administrator, educator and programmer have higher amount of participating in this survey.

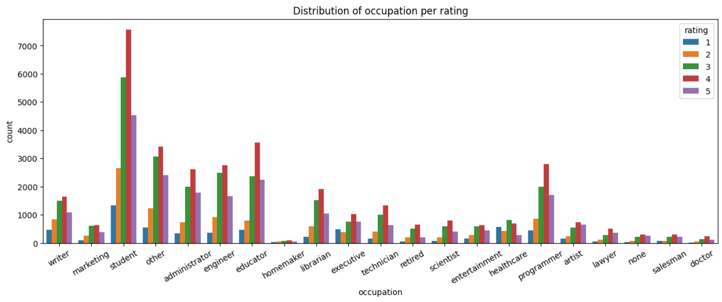


Figure 6: Distribution of rating per user occupation

After verifying number of movie rating, the result illustrates, minimum number of reviews is 1 and maximum number of reviews is 583 for movies, that tells the rating has a high variance. It can be observed from average number of reviews of 59.45 and median number of reviews of 27.0.

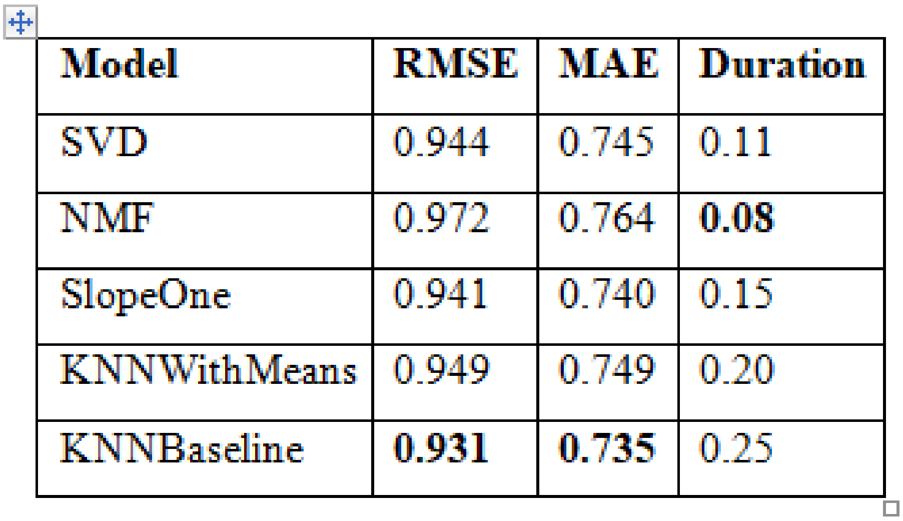
6 Evaluation

This section aims to offer a thorough examination of the research's conclusions and key findings, as well as the major findings associated with the research paper. The findings has been presented one by one in a tabluar format-

6.1 Performance of training data with different algorithms

After running all the models that are SVD, NMF, SlopeOne, KNNWithMeans and KNNBaseline the following performance is achieved that are donated in coming table. It is observed that the KNNBaseline with RMSE of 0.931 and MAE of 0.735 could achieved higher performance on test data. The training takes 0.25 minute which is longer as compared to all other models.

After using SlopeOne with RMSE of 0.941 and MAE of 0.74 has been obtained as less error rates.The training duration of the SlopeOne was 0.15 that is 0.1 minutes less than the KNNBaseline algorithm. Based on this achievement although, the KNNBaseline model took longer time for training but is could cultivate 0.01 RMSE and 0.005 MAE.Thus, it is selected as final algorithm that is sent to hyperparameter tuning.



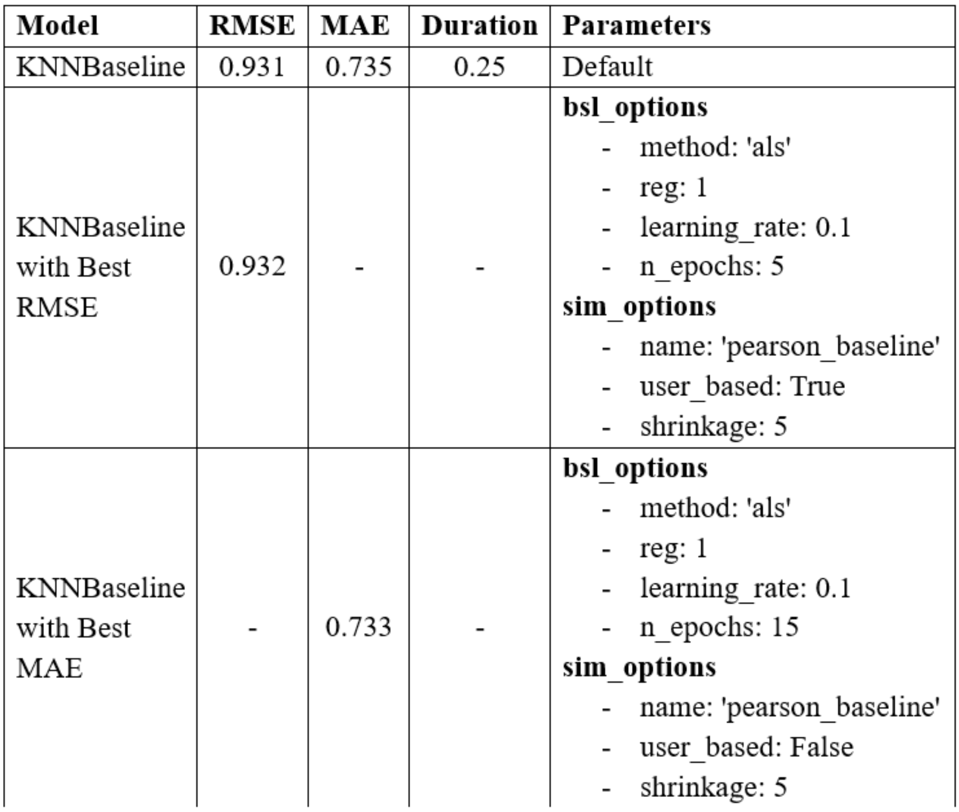
Model	RMSE	MAE	Duration
SVD	0.944	0.745	0.11
NMF	0.972	0.764	**0.08**
SlopeOne	0.941	0.740	0.15
KNNWithMeans	0.949	0.749	0.20
KNNBaseline	**0.931**	**0.735**	0.25

Figure 7: Performance results of 5 different models

6.2 Performance of training data with different algorithms

The best model was KNNBaseline that could obtain the best performance among all; thus, it is sent to hyperparameter tunings with GridSearchCV. It run by 3-flod cross validation and took 137 minutes to verify all the possible combination of parameters for KNNBaseline model. The parameters were suggested for both RMSE and MAE evaluation metrics separately.

It is observed that there is minor improvement on MAE after hyperparameter tuning that MAE of the model reduced from 0.735 to 0.733, and these parameters will use for predicting final recommendation system.



Model	RMSE	MAE	Duration	Parameters
KNNBaseline	0.931	0.735	0.25	Default
KNNBaseline with Best RMSE	0.932	-	-	**bsl_options** - method: 'als' - reg: 1 - learning_rate: 0.1 - n_epochs: 5 **sim_options** - name: 'pearson_baseline' - user_based: True - shrinkage: 5
KNNBaseline with Best MAE	-	0.733	-	**bsl_options** - method: 'als' - reg: 1 - learning_rate: 0.1 - n_epochs: 15 **sim_options** - name: 'pearson_baseline' - user_based: False - shrinkage: 5

Figure 8: Comparison performance of baseline model with tuned models

6.3 Sample running the recommendation system

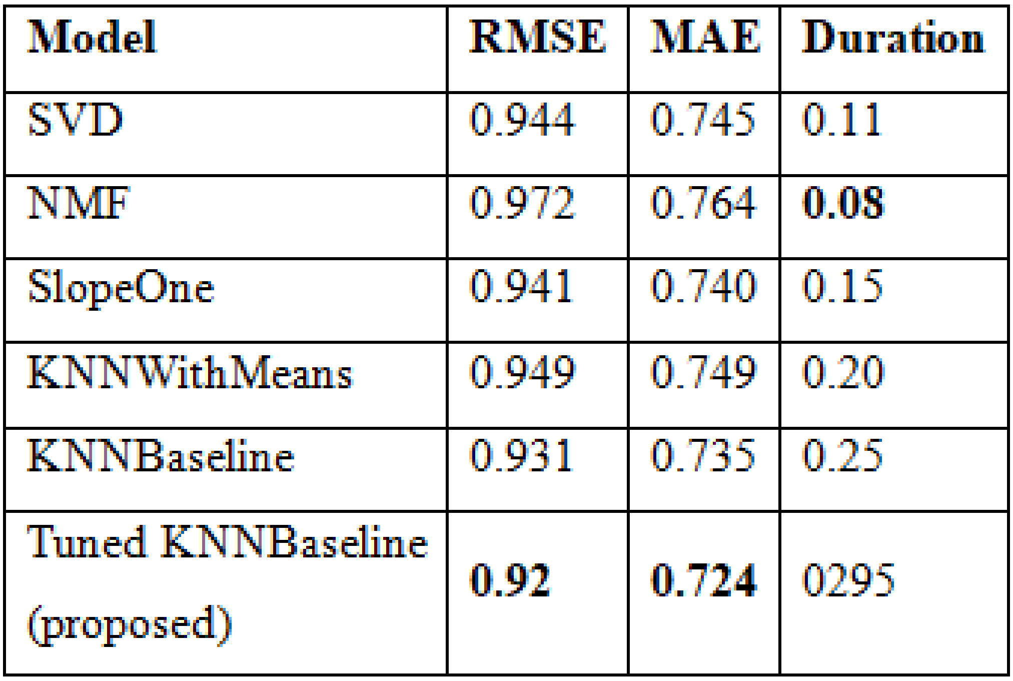
As explained in user interface section, the user should enter userId and n (number of movies to recommend), the model will exclude all the movies watched by this user and predict the rating for remaining movies, then the n top-rated movies will return as the recommended movie for user. The sample output is represented in following:

```
Please enter user id: 196
Please enter number on movie recommendation: 5
Recommended movies for user id 196 is:

- North by Northwest (1959)
- 12 Angry Men (1957)
- Citizen Kane (1941)
- Wallace and Gromit: The Best of Aardman Animation (1996)
- One Flew Over the Cuckoo's Nest (1975)
```

7 Conclusion and Future Work

The comparison table of all models denoted in following table:



Model	RMSE	MAE	Duration
SVD	0.944	0.745	0.11
NMF	0.972	0.764	**0.08**
SlopeOne	0.941	0.740	0.15
KNNWithMeans	0.949	0.749	0.20
KNNBaseline	0.931	0.735	0.25
Tuned KNNBaseline (proposed)	**0.92**	**0.724**	0295

Figure 9: Comparison of all models

In this work various models from different categories such as KNN base, Matrix Factorization base and slope One algorithms are selected to verify which group are more suitable for this specific data. In addition, cross validation is used to train models with diverse combination of data that help to model find the patterns more accurately. These two helps to achieve high performance with low cost which helps users to find their interested movies in the shortest time.

It is observed that the proposed model which is the tuned KNNBaseline model could achieve the lower error rate which is around 0.01 for RMSE and 0.01 for MAE. After that the best model among all is KNNBaseline with default values for parameters. On the other hand, the tuned model needed the longest time for training that is around 0.04 minute more than best model. The SlopeOne train in shortest time but its performance is not as high as other models and it has the highest rate of errors as compare to other modes. For the future work, there is some suggestion:

- There are some other algorithms in Surprise library than can be verified
- A hybrid model includes combination of content base filtering and collaborating filtering can try.

References

Andersson, M. and Tran, L. (2020). Predicting movie ratings using knn.

Furtado, F. and Singh, A. (2020). Movie recommendation system using machine learning., *International journal of research in industrial engineering,* **9**(1): pp.84–98.

Furtado, F. and Singh, A. (2022). Movie recommendation system modeling using machine learning., *International Journal of Mathematical, Engineering, Biological and Applied Computing* pp. pp.12–16.

Keshava, M.C., R. P. S. S. and Naik, B. (2020). Machine learning model for movie recommendation system., *International Journal of Engineering Research Technology (IJERT),* **9**(04): pp.2278–0181.

Pouransari, H. and Ghili, S. (2014). Deep learning for sentiment analysis of movie reviews., *CS224N Proj,* pp. pp.1–8.

Rajarajeswari, S., N. S. S. S. S. P. M. and Uday, P. (2019). Movie recommendation system., *In Emerging Research in Computing, Information, Communication and Applications: ERCICA 2018,* **1**: pp. 329–340.

Sharifi, Z., R. M. and Nasiri, M. (2014). A new algorithm for solving data sparsity problem based-on non negative matrix factorization in recommender systems., *In 2014 4th International Conference on Computer and Knowledge Engineering (ICCKE)* pp. pp. 56–61.

Sinaga, K. and Yang, M. (2020). Unsupervised k-means clustering algorithm., *IEEE access,* **8**: pp.80716–80727.

Song, Y. and Wu, S. (2020). Slope one recommendation algorithm based on user clustering and scoring preferences., *Procedia Computer Science,* **166**: pp.539–545.

Steurer, M., H. R. and Pfeifer, N. (2021). Metrics for evaluating the performance of machine learning based automated valuation models., *Journal of Property Research,* **38**(2): pp.99–129.

Surendran, A., Y. A. and Kumar, A. (2020). Movie recommendation system using machine learning algorithms., *Eng Technol.* .

Wang, P., Q. Q. S. Z. and Li, J. (2016). An recommendation algorithm based on weighted slope one algorithm and user-based collaborative filtering., *In 2016 Chinese Control and Decision Conference (CCDC)* pp. pp.2431–2434.

Wu, C.S.M., G. D. and Bhandary, U. (2018). Movie recommendation system using collaborative filtering., *In 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)* **9**(1): pp. 11–15.