# IMPROVE PERFORMANCE OF MOVIE RECOMMENDATION SYSTEMS USING MULTI-CRITERIA AND SUPERVISED LEARNING

A PROJECT REPORT

*Submitted by*


CB.EN.U4CSE16303 A.S.S.S. RAM CHANDU

CB.EN.U4CSE16304 ADVAITH M.S.

CB.EN.U4CSE16323 GOPIKRISHNA K.S.

CB.EN.U4CSE16448 PAVITHRAN S.


*in partial fulfillment for the award of the degree of*

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING



AMRITA SCHOOL OF ENGINEERING, COIMBATORE

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE 641 112


OCTOBER 2019

**AMRITA VISHWA VIDYAPEETHAM**

**AMRITA SCHOOL OF ENGINEERING, COIMBATORE, 641112**


## BONAFIDE CERTIFICATE

This is to certify that the project report entitled " **Improve performance of Movie Recommendation Systems using multi-criteria and Supervised Learning**" submitted by A.S.S.S. RAM CHANDU (CB.EN.U4CSE16303), ADVAITH M.S. (CB.EN.U4CSE16304), GOPIKRISHNA K.S. (CB.EN.U4CSE16323) and PAVITHRAN S. (CB.EN.U4CSE16448) in partial fulfillment of the requirements for the award of the Degree **Bachelor of Technology in Computer Science and Engineering** is a bonafide record of the work carried out under our guidance and supervision at Department of Computer Science and Engineering, Amrita School of Engineering, Coimbatore

| PROJECT GUIDE | CHAIRPERSON |
|---|---|
| **Dr. C Selvi, Vidhya S.(Co-guide)** | **Dr. (Col) P.N. Kumar** |
| Assistant Professor | Professor |
| Dept. of Computer Science and Engg. | Dept. of Computer Science and Engg. |

This project report was evaluated by us on :. . . . . .

INTERNAL EXAMINER EXTERNAL EXAMINER

## Acknowledgment

We express our gratitude to our beloved Satguru Sri Mata Amritanandamayi Devi for providing a bright academic climate at this university, which has made this entire task appreciable. This acknowledgement is intended to be a thanks giving measure to all those people involved directly or indirectly with our project.

We would like to thank our Vice Chancellor Dr. Venkat Rangan. P and Dr. Sasangan Ramanathan Dean Engineering of Amrita Vishwa Vidyapeetham for providing us the necessary infrastructure required for completion of the project.

We express our thanks to Dr.(Col.P.N.Kumar), Chairperson of Department of Computer Science Engineering, Dr.C.Shunmuga Velayutham and Dr. G. Jeyakumar, Vice Chairpersons of the Department of Computer Science and Engineering for their valuable help and support during our study. We express our gratitude to our guide, Dr. C Selvi, Vidhya S.(Co-guide) , for the guidance, support and supervision. We feel extremely grateful to Mr. Prashant R. Nair, Dr. Senthil Kumar M., Mr. A. Baskar, Ms. Dhanya M. Dhanalakshmy and Ms. P. Subathra for their feedback and encouragement which helped us to complete the project. We also thank the staff of the Department of Computer Science Engineering for their support. We would like to extend our sincere thanks to our family and friends for helping and motivating us during the course of the project.

**Abstract**

Recommendation systems is a subclass of information filtering system that seeks to predict the "rating" or "preference" a user would give to an item. Having a good recommendation system is an indispensable need for majority of companies in various scenarios as it helps businesses to target their customers with appropriate products thus leading to efficient management of resources and increased profit. The aim of the project is to improve the performance of recommendation systems by transforming collaborative filtering into supervised learning. The impact of adding timestamp of the rating and user demographics like gender, age, zip code to different techniques like random forest, artificial neural network is observed.

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Background

Recommender systems is a subclass of information filtering system that seeks to predict the "rating" or "preference" a user would give to an item. They are primarily used in commercial applications. Nowadays, users are provided with too many choices making it almost impossible for them to get what they actually need. Thus comes the need for personalised recommendation systems thereby benefiting all the stakeholders. The proposed solution is to compare the existing models that transform collaborative filtering into supervised learning and improve the performance by adding more features.

## 1.2 Problem Statement

To improve the performance of recommendation systems by transforming collaborative filtering into techniques of supervised learning, and assess impact of adding more features to input.

## 1.3 Specific Objectives

The dataset used is the Movie Lens 100k dataset which comprises of 1, 00, 000 ratings given by 943 users to 1682 movies. After implementing the base paper,a model will be proposed which takes into consideration multi-criteria such as the users age, gender, occupation, zipcode rather than looking into only the ratings given by the user. Advanced supervised learning techniques will be then used to improve the accuracy of recommendation systems that use collaborative filtering.

## 1.4 Findings

In the base paper , a methodology was proposed to transform the Collaborative Filtering scheme into a Supervised Learning scheme through the construction of an input feature space based on the latent variables extracted from the rating matrix. The proposed method also didn't consider the features associated with users such as demographic data and user opinions and sentiments.
The proposed method also didn't use timestamp as a feature.

# Chapter 2

# LiteratureSurvey

## Transforming collaborative filtering to Supervised Learning[1]

The input is a n*m two dimensional matrix having n users and m columns and r[i, j] represents the rating given by user i for the product j. The matrix is sparse as in very few ratings are known. The aim is to find the unknown ratings by extrapolating from the filled ones. The input is only the matrix and has to be converted to its factors(input and output) to apply traditional supervised learning methods. Dimensionality reduction techniques to transform sparse vector into fixed space of features. New input feature space where user-item pairs are represented by vectors of features, labeled with their corresponding ratings. Steps involved include preprocessing, latent variables extraction, and regression/ classification. The parameters are number of latent variables, method for extracting those variables, strategy to deal with unfilled positions of the rating matrix, supervised learning algorithm.

## Toward the Next Generation of Recommendation Systems: A Survey of the State-of-the-Art and Possible Extensions[2]

Recommendation systems are mainly of three types.

- Content-based recommendations: The user will be recommended items similar to the ones the user preferred in the past

  Methods

  - TF-IDF : Important keywords are extracted and corresponding vectors are constructed
  - Naive Bayesian Classifier : The probability that it belongs to each class given the features is calculated and class with maximum probability is selected.

  Limitations

  - Overspecialisation : user is limited to being recommended items that are similar to those already rated
  - New User Problem

- Collaborative recommendations: The user will be recommended items that people with similar tastes and preferences liked in the past.

  Limitations

  - New User Problem
  - New Item Problem
  - Sparsity : can be overcome using Demographic Filtering, Transitive Associations, Singular Value Decomposition

- Hybrid approaches: These methods combine collaborative and content-based methods.

The following are ways to extend capabilities of a recommendation system

- Multidimensionality of Recommendations : vacation package should consider time, with whom, traveling conditions, restrictions, movie should consider where and how, with whom, when and other contextual information.

- Comprehensive understanding of users and items : advanced profiling of users and items based on keywords, demographic information
- Multcriteria Ratings : Optimise the important criterion and convert the other criteria to constraints.

## Matrix Factorization Techniques for Recommendation Systems[3]

The input is a latent factor space of dimensionality 'f' mapping both users and items ,such that user-item interactions are modelled as inner products in that space. To rectify missing values a system is modelled that minimizes the regularized squared error on the set of known ratings.User bias and Item bias with the global average rating is added to the matrix factorization model.To avoid the cold start Problem Boolean implicit feedback is considered by having a set of items for which the user has given implicit feedback .Another information source that is including is of user attributes which describe the gender ,age group and other attributes.Item bias ,user bias and user preferences can change over time , therefore these features are modeled as a function of time , so that they can change dynamically.Item characteristics are always static in nature so don't have to be modeled as a function of time.All ratings do not deserve the same weight or confidence as many of them might be temporary due to factors such as advertising etc. So a confidence value is assigned to the cost function and while giving the confidence value,various factors such as if its a recurring event are considered.we can infer that the more complex factor models, whose descriptions involve more distinct sets of parameters, are more accurate, and that temporal components are particularly important to model as there are significant temporal effects in the data.This model delivers accuracy superior to classical nearest-neighbor techniques and offer a compact memory-efficient model that systems can learn relatively easily.The techniques are very useful as models can integrate naturally many crucial aspects of the data, such as multiple forms of

feedback, temporal dynamics, and confidence levels.

## Collaborative filtering and deep learning based recommendation system for cold start items[4]

Collaborative filtering (CF) is the most popular approaches used for recommender systems, but it suffers from complete cold start (CCS) problem where no rating record are available and incomplete cold start (ICS)problem where only a small number of rating records are available for some new items or users in the system. In this paper, we propose two recommendation models to solve the CCS and ICS problems for new items, which are based on a framework of tightly coupled CF approach and deep learning neural network. A specific deep neural network SADE is used to extract the content features of the items. The state of the art CF model, timeSVD++, which models and utilizes temporal dynamics of user preferences and item features, is modified to take the content features into prediction of ratings for cold start items. Extensive experiments on a large Netflix rating dataset of movies are performed, which show that our proposed recommendation models largely outperform the baseline models for rating prediction of cold start items

## Pessimists and optimists: Improving collaborative filtering through sentiment analysis[5]

This work presents a novel application of Sentiment Analysis in Recommendation Systems by categorizing users according to the average polarity of their comments. These categories are used as attributes in Collaborative Filtering algorithms. To test this solution a new corpus of opinions on movies obtained from the Internet Movie Database (IMDb) has been generated, so both ratings and comments are available. The experiments stress the informative value of comments. By applying Sentiment Analysis approaches some Collaborative Filtering algorithms can be improved in rat-

ing prediction tasks. The results indicate that we obtain a more reliable prediction considering only the opinion text , than when apply similarities over the entire user community and sentiment analysis can be advantageous to recommender systems. This will also take into account the distances between the opinion texts and the sentiments expressed in these opinions, integrating polarity values from comments as factors in inter-vector distances.

| Citation | Author Details | Methodology | Dataset | Advantages/ Disadvantages |
|---|---|---|---|---|
| [1] | Filipe Braida, Carlos E Mello, Marden B Pasinato, Geraldo Zimbrao | Latent variable extraction andmapping to new space, classification and regression | Movie Lens 100K | Better accuracy than traditional content based and collaborative based techniques but no multi criteria ratings |
| [2] | Gediminas Adomavicius, Member, IEEE, and Alexander Tuzhilin, Member, IEEE | Multi dimensional recommender system and multi criteria user rating | NA | Limited Content Analysis, Over Specialisation, New-User, New-Item Problem, Sparsity |
| [3] | Yehuda Koren ,Robert Bell ,Chris Volinsky | Matrix factorization | Netflix Prize | Accuracy superior to classical nearest-neighbor techniques, Temporal Dynamics, Compact memory-efficient model |
| [4] | Miguel A.García-Cumbreras, Arturo Montejo-Ráez, Manuel,C. Díaz-Galiano | Sentimental Analysis | IMDB dataset | Useful in Building Context.Improves Collaborative Filtering |
| [5] | Jian Wei, Jianhua He, Kai Chen, Yi Zhou, Zuoyin Tang | Collaborative filtering, SVD++ | Netflix Movies | Extraction of item content features and prediction of unknown ratings. |

Table 2.1: Summary of Related Papers

# Chapter 3

# Proposed System

## 3.1 System Architecture

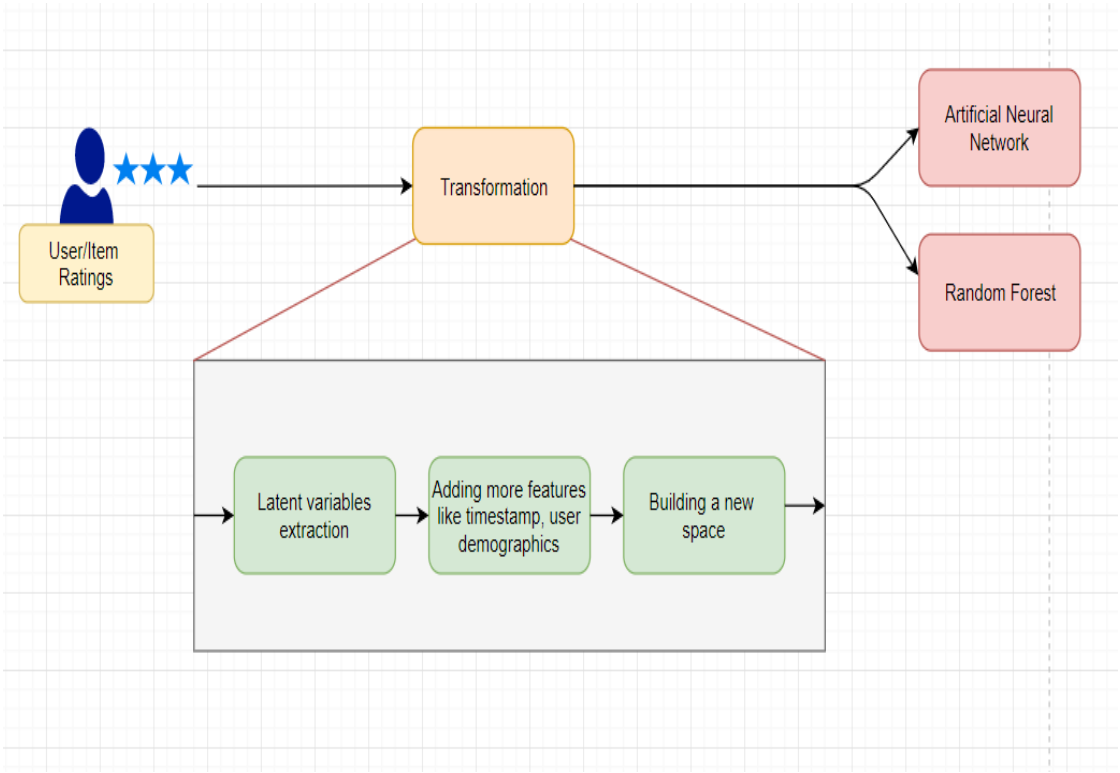


Figure 3.1: Process flow diagram

The Movie Lens dataset is a table having columns as userid, movieid, rating ,age , gender , occupation,zip code. So first we consider only the columns userid, movieid

and rating which are preprocessed by reshaping in such a way that the rows represent users and the columns represent movies and each cell represents the rating. Missing values were replaced with zeroes and the values were mean centered. Latent variables are then extracted from this dataframe and concatenated with the other columns i.e, age, gender, occupation, zip code which are preprocessed beforehand by normalizing age, zipcode while gender and occupation are converted to numerical values.This forms the input for the different supervised learning algorithms that would be performed. The supervised learning algorithms executed include random forest and artificial neural network.

## 3.2 System Specification

Jupyter notebook environment

Numpy - Version 1.16.4

Pandas - Version 0.24.2

Scipy - Version 1.3.0

Keras - Version 2.2.4

## 3.3 Methodology

- Singular Value Decomposition : In linear algebra, the singular value decomposition (SVD) is a factorization of a real or complex matrix. It is the generalization of the eigen decomposition of a positive semi definite normal matrix (for example, a symmetric matrix with positive eigenvalues) to any (m x n) matrix via an extension of the polar decomposition. It has many useful applications in signal processing and statistics. In the project, it is used for dimensionality reduction to transform the sparse vector into fixed space of features and to extract the latent variables.

- Random forests: It is an ensemble learning method for classification, regression

and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean prediction of the individual trees.It has the power to handle a large data set with higher dimensionality and also if there are more trees, it won't allow overfitting trees in the model.

- Artificial neural network (ANN): It is a computational model based on the structure and functions of biological neural networks.ANNs have three layers that are interconnected. The first layer consists of input neurons. Those neurons send data on to the hidden layer, which in turn sends the output neurons to the third layer. It has cost-effective and ideal methods for arriving at solutions while defining computing functions or distributions and can be used for supervised learning to extrapolate the unknown ratings .

## 3.4 Implementation

The data is loaded and reshaped followed by extraction of the top eight latent variables. These features of users and items are concatenated along with age, gender, occupation, zip code . Age, zip code are scaled while gender and occupation are converted to numerical values. This forms the input where comparison of different supervised algorithms is performed.

# Chapter 4

# Results and Discussion

It is found that adding features like timestamp and user demographics like gender, age, zip code reduces the error in techniques namely random forests, and artificial neural network as clear from the screenshots below.

```
[88] from sklearn.ensemble import RandomForestClassifier

     RFmodel = RandomForestClassifier(n_estimators=100,
                                      bootstrap = True,
                                      max_features = 'sqrt')

     RFmodel.fit(X_train, y1_train)

     rf_predictions = RFmodel.predict(X_test)

     from sklearn import metrics
     print("MAE:",metrics.mean_absolute_error(y1_test, rf_predictions))
     print("MSE:",metrics.mean_squared_error(y1_test, rf_predictions))
```

```
MAE: 0.9710498409331919
MSE: 1.7277836691410393
```

Figure 4.1: Random Forest before adding multi criteria

```
from sklearn.ensemble import RandomForestClassifier

RFmodel = RandomForestClassifier(n_estimators=100,
                                 bootstrap = True,
                                 max_features = 'sqrt')
```

```
RFmodel.fit(X_train, y_train)
rf_predictions=RFmodel.predict(X_test)
```

```
from sklearn import metrics

print("Accuracy:",metrics.accuracy_score(y_test, rf_predictions))
print("MAE:",metrics.mean_absolute_error(y_test, rf_predictions))
print("MSE:",metrics.mean_squared_error(y_test, rf_predictions))
```

```
Accuracy: 0.32767762460233296
MAE: 0.9667020148462354
MSE: 1.7217391304347827
```

Figure 4.2: Random Forest after adding multi criteria

```
[ ]  from keras.layers import Dense, Dropout
     from keras.optimizers import RMSprop
```

```
[ ]  model1 = Sequential()
     model1.add(Dense(32, activation='relu', input_shape=(16,)))
     model1.add(Dropout(0.2))
     model1.add(Dense(64, activation='relu'))
     model1.add(Dropout(0.2))
     model1.add(Dense(128, activation='relu'))
     model1.add(Dropout(0.2))
     model1.add(Dense(64, activation='relu'))
     model1.add(Dropout(0.2))
     model1.add(Dense(32, activation='relu'))
     model1.add(Dropout(0.2))
     model1.add(Dense(6, activation='softmax'))

     model1.summary()

     model1.compile(loss='categorical_crossentropy',
                    optimizer='rmsprop',
                    metrics=['accuracy', 'mae', 'mse'])

     model1.fit(X_train, y_categorical_train, epochs = 20)
```

```
[ ]  accuracy = model1.evaluate(X_test, y_test)
```

```
     9430/9430 [==============================] - 0s 26us/step
```

```
[86] print("MAE:", accuracy[1])
     print("MSE:", accuracy[2])
```

```
     MAE: 0.35344644754587784
     MSE: 0.24151230579911714
```

Figure 4.3: Neural Network before adding multi criteria

```
model1 = Sequential()
model1.add(Dense(64, activation='relu', input_shape=(42,)))
model1.add(Dropout(0.2))
model1.add(Dense(128, activation='relu'))
model1.add(Dropout(0.2))
model1.add(Dense(256, activation='relu'))
model1.add(Dropout(0.2))
model1.add(Dense(256, activation='relu'))
model1.add(Dropout(0.2))
model1.add(Dense(128, activation='relu'))
model1.add(Dropout(0.2))
model1.add(Dense(64, activation='relu'))
model1.add(Dropout(0.2))
model1.add(Dense(42, activation='relu'))
model1.add(Dropout(0.2))
model1.add(Dense(6, activation='softmax'))

model1.summary()

model1.compile(loss='categorical_crossentropy',
               optimizer='rmsprop',
               metrics=['mae', 'mse'])

model1.fit(X_train, y_categorical_train, epochs = 20)
```

```
accuracy = model1.evaluate(X_test, y_categorical_test)
```

```
print("MAE:",accuracy[1])
print("MSE:",accuracy[2])
```

```
MAE: 0.2425119848607833
MSE: 0.13018517331600443
```

Figure 4.4: Neural Network after adding multi criteria

# Chapter 5

# Conclusion and Future Work

In this work, we proposed a methodology where we have added multi-criteria and timestamp as a feature to an existing model which is an input feature space based on the latent variables extracted from the rating matrix. In order to evaluate we have compared several supervised learning algorithms. In this analysis, Artificial Neural Networks and Random Forest were considered for comparison. The results has proven that our method reduces the prediction error compared to the previous model.

To further improve accuracy, reviews of the users for the movies could be used as a feature in the input. Also, multi criteria ratings could be used which means to say the user should evaluate the movie on various parameters like story, visuals, etc. along with an overall rating to provide a more personalised recommendation.

# Chapter 6

# Bibliography


[1] Filipe Braida, Carlos E Mello, Marden B Pasinato, Geraldo Zimbrao **Transforming collaborative filtering into supervised learning** Elsevier Expert Systems with Applications Volume 42, Issue 10 (2015),4733 - 4742

[2] Gediminas Adomavicius ,Alexander Tuzhilin **Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions** IEEE Transactions on Knowledge and Data Engineering Volume 17, Issue 6 (2005),734-749

[3] Yehuda Koren ,Robert Bell ,Chris Volinsky **Matrix Factorization Techniques for Recommender Systems** IEEE Computer Volume 42, Issue 8 (2009),30 -37

[4] Miguel A.Garc ia-Cumbreras,Arturo Montejo-Raez,Manuel C.D iaz-Galiano. **Pessimists and optimists: Improving collaborative filtering through sentiment analysis** Elsevier Expert Systems with Applications Volume 40,Issue 17(2013),6758-6765

[5] JianWei ,JianhuaHe, KaiChen, YiZhou, ZuoyinTang **Collaborative filtering and deep learning based recommendation system for cold start items** Elsevier Expert Systems with Applications Volume 69,(2017),29-39