

Software Design Document (SDD) Template

Software design is a process by which the software requirements are translated into a representation of software components, interfaces, and data necessary for the implementation phase. The SDD shows how the software system will be structured to satisfy the requirements. It is the primary reference for code development and, therefore, it must contain all the information required by a programmer to write code. The SDD is performed in two stages. The first is a preliminary design in which the overall system architecture and data architecture is defined. In the second stage, i.e. the detailed design stage, more detailed data structures are defined and algorithms are developed for the defined architecture.

This template is an annotated outline for a software design document adapted from the IEEE Recommended Practice for Software Design Descriptions. The IEEE Recommended Practice for Software Design Descriptions have been reduced in order to simplify this assignment while still retaining the main components and providing a general idea of a project definition report. For your own information, please refer to [IEEE Std 1016-1998](http://www.cs.concordia.ca/~ormandj/comp354/2003/Project/ieee-SDD.pdf)¹ for the full IEEE Recommended Practice for Software Design Descriptions.

¹<http://www.cs.concordia.ca/~ormandj/comp354/2003/Project/ieee-SDD.pdf>

Movie Recommender System

Software Design Document

Names : Jordan Tangy, Ron Sider, Max Krakauer

Lab Section :

Work Station :

12/01/2021

TABLE OF CONTENTS

1.	INTRODUCTION	2
1.1	Purpose	2
1.2	Scope	2
1.3	Overview	2
1.4	Reference Material	2
1.5	Definitions and Acronyms	2
2.	SYSTEM OVERVIEW	2
3.	SYSTEM ARCHITECTURE	2
3.1	Architectural Design	2
3.2	Decomposition Description	3
3.3	Design Rationale	3
4.	DATA DESIGN	3
4.1	Data Description	3
4.2	Data Dictionary	3
5.	COMPONENT DESIGN	3
6.	HUMAN INTERFACE DESIGN	4
6.1	Overview of User Interface	4
6.2	Screen Images	4
6.3	Screen Objects and Actions	4
7.	REQUIREMENTS MATRIX	4
8.	APPENDICES	4

1. INTRODUCTION

1.1 Purpose

This software design document describes the architecture and system design of our Movie Recommender System.

1.2 Scope

Provide a description and scope of the software and explain the goals, objectives and benefits of your project. This will provide the basis for the brief description of your product.

The goal of this software is to provide to a reliable and friendly platform in order to get best results in terms of choices of movies. Our objective is to create a software that will be able to identify similar users and help the user choose the best movie according to its preferences.

1.3 Overview

The goal of this document is to present as accurate as possible our organization in order to create a software designed to help user in choosing movie by explaining how the system should work, which techniques will be used and how to use the data in our possession in order to get the best results and how we will capitalize it.

1.4 Reference Material

Article found on the Internet :

<https://realpython.com/build-recommendation-engine-collaborative-filtering/#:~:text=Collaborative%20filtering%20is%20a%20family,type%20of%20collaborative%20filtering%20approach.>

- *Deep Learning course from Ariel University*
- *Youtube channels of people explaining about Mathematic concepts and Machine Learning:*
- *Machine Learning :* <https://www.youtube.com/watch?v=SD3irxdKfxk&t=419s>
- *Mathematic concepts :* <https://www.youtube.com/watch?v=ZspR5PZemcs>

1.5 Definitions and Acronyms

- **Machine Learning :**

Machine learning (ML) is the study of computer algorithms that improve automatically through experience. It is seen as a subset of [artificial intelligence](#). Machine learning algorithms build a model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.

- **Recommender System :**

A **recommender system**, or a **recommendation system** 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.

- **Collaborative Filtering :**

Collaborative filtering (CF) is a technique used by recommender systems.^[1] Collaborative filtering has two senses, a narrow one and a more general one.^[2] In the newer, narrower sense, collaborative filtering is a method of making automatic predictions (filtering) about the interests of a user by collecting preferences or taste information from many users (collaborating).

2. SYSTEM OVERVIEW

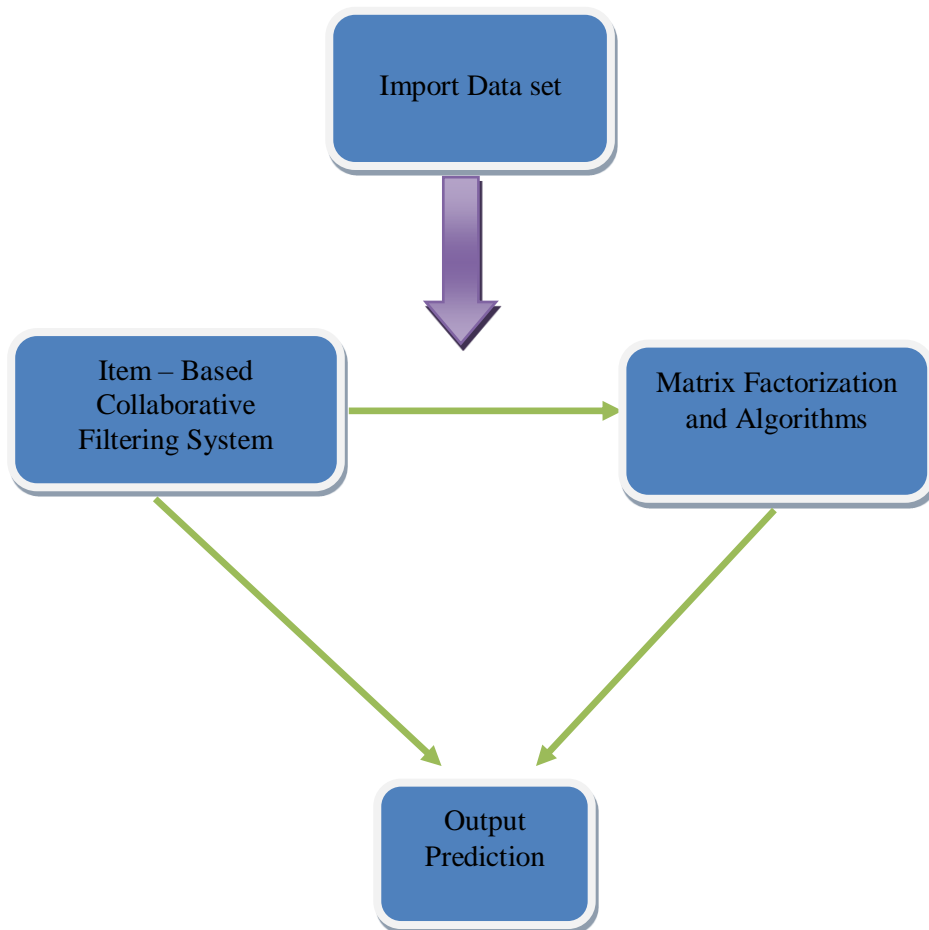
Nowadays , possibilities are almost infinite in terms of entertainment. In the domain of screen entertainment there are so much to TV shows, documentaries, movies, series that sometimes, the users can find themselves lost among all those options. Additionally, there is a huge quantity of data that has been collected over the years, mainly users data. During the last few decades, the field of Machine Learning has known an exponential growth , particularly because the amount of data has grown over the last few years.

In this project we want to create a Movie Recommender System that will allow to the users to get a list of potential movies that they could like with a high probability.

In addition, we would like the user to interact with the platform by indicating and adjusting its taste according to the recommender results.

3. SYSTEM ARCHITECTURE

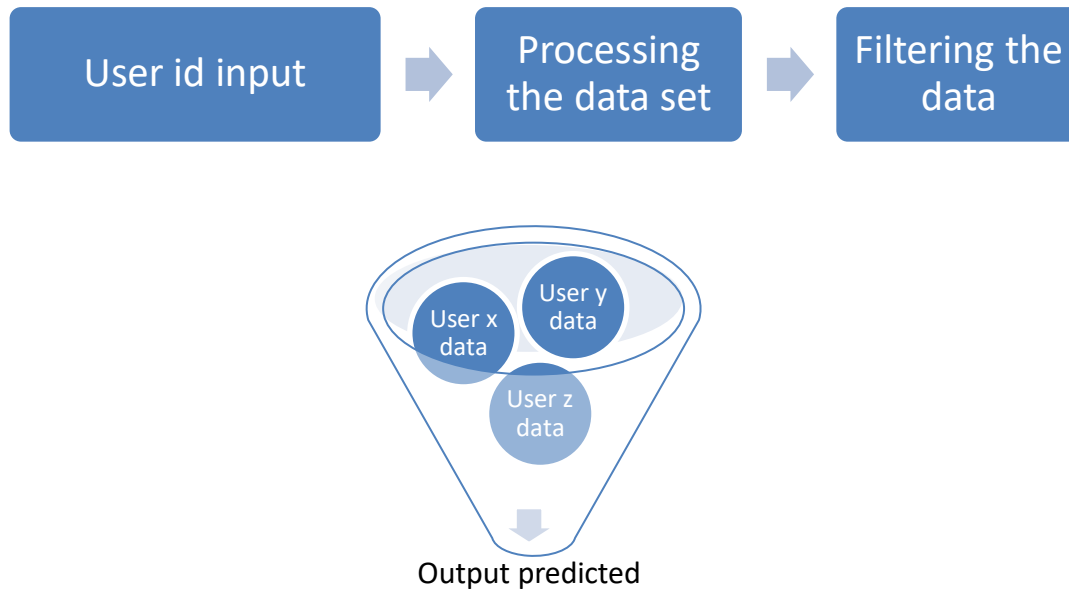
3.1 Architectural Design



The architecture of the Movie Recommender starts by gathering the relevant data and input it into the system. It comes in in form of a matrix. This matrix is processed by a Collaborative Filtering system and its purpose is to detect same items (movies) liked by different user. In order to do so, we use a technique called matrix factorization and algorithms in order to extract similar user i.e. , users that liked the same kind of movie. Based on that, the system will be able to make a prediction.

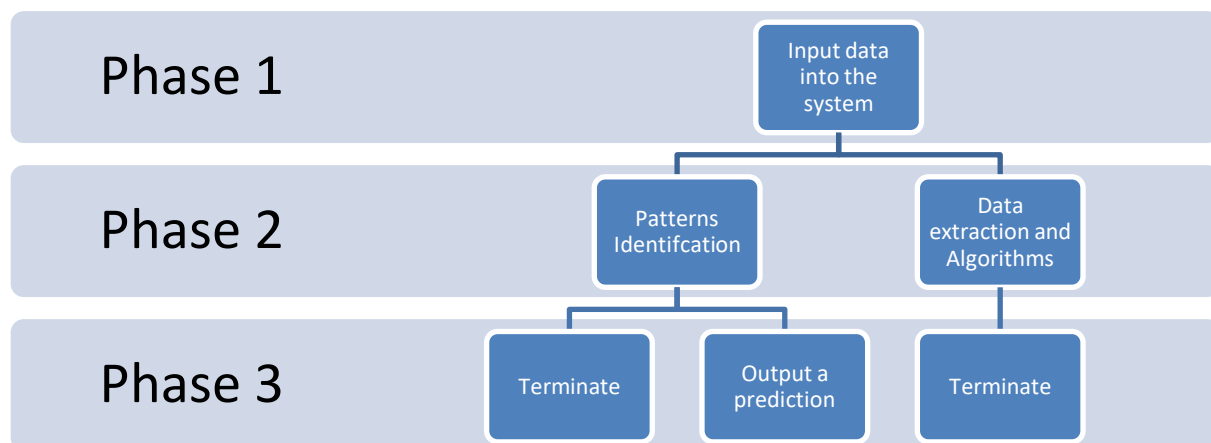
3.2 Decomposition Description

Data flow diagram (DFD):



In this diagram, the data flows through different component. The client should first give its user Id in order to tell to the system that we want to find similar users for this specific client. The data is processed and the system is starting to extract information. Then the systems is filtering the user collaboratively and finally we get a prediction.

Structural Decomposition Diagram :



3.3 Design Rationale

The above architecture was selected because it is one of the most efficient architecture used in this field and gives generally good results. This architecture is commonly used in the field of recommenders and its performances are optimal.

The considered architecture was using a Neural Network that will predict which genres are the most popular among the users and finally recommend the best genres. The problem with that kind of architecture is that the results are not good enough and there are no options to personalize the result and refine the recommendation.

4. DATA DESIGN

4.1 Data Description

The dataset is "Movielens 25M Dataset".

The data is composed of 25 million user-ratings and one million tag applications applied to 62,000 movies by 162,000 users. Each movie consists of 1129 tags. Each tag is scaled in a range of (0-1), showing the correlation level of the tag to that particular movie.

The dataset originally comes in an Excel format and processed through the pandas library in Python. In processing the data, we filter and make a distinction between relevant and irrelevant data for our main goal.

As suggested in previous sections, for collaborative filtering we'll use the matrix factorization method. As the name implies our main data structure is a matrix.

We use matrix as our data structure to infer the relations between features (the matrix below is not part of the algorithm but rather a demonstration of how the concept works)

	Movie1	Movie2	Movie3	Movie4	Movie5
U1		5	4	2	1
U2	1			5	3
U3	1	4	4	1	
U4			2		2
U5	3	1	1		

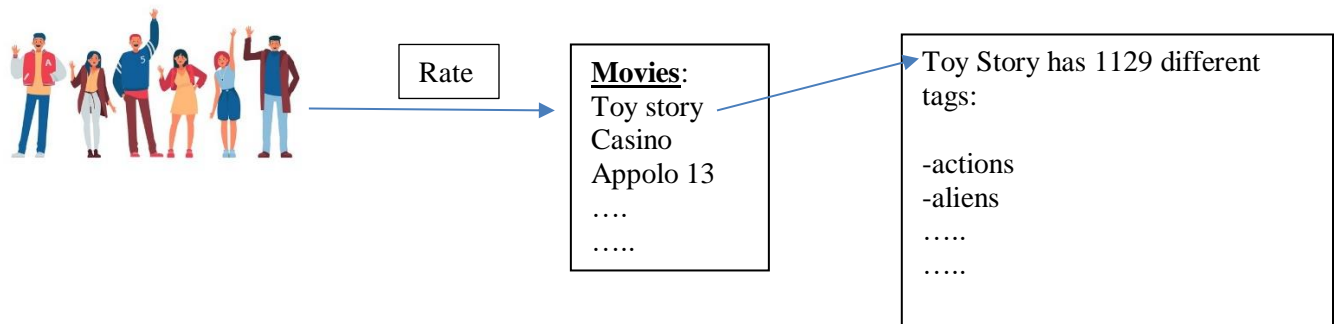
Movie 5:

- Intense
- horror
- Surreal
-
- ...

4.2 Data Dictionary

Users

Example



5. COMPONENT DESIGN

Since not every user gives ratings to all the movies, there are many missing values in the matrix and it results in a sparse matrix. Hence, the null values not given by the users would be filled with 0 such that the filled values are provided for the multiplication. For example, two users give high ratings to a certain move when the movie is acted by their favorite actor and actress or the movie genre is an action one, etc. From the table in section 4.1, we can find that the user1 and user3 both give high ratings to movies 2 and 3. Hence, from the matrix factorization, we are able to discover these latent features to give a prediction on a rating with respect to the similarity in users' preferences and interactions.

Given a scenario, user 4 didn't give a rating to the movie 4. We'd like to know if user 4 would like movie 4. The method is to discover other users with preferences similar to those of user 4 by taking the ratings given by users with similar preferences to the movie 4 and predict whether the user 4 would like the movie 4 or not.

A simplistic sketch of the concept:

User 1

User 2

Rating: 5 stars

A movie

We may infer both users like a movie based on it's features(tags).

Scenario:

User 1 has seen a movie and gave it a 5 star rating whereas user 2 didn't rate this movie. We deduce user 2 will give this movie a high rating based on shared preferences.

6. HUMAN INTERFACE DESIGN

6.1 Overview of User Interface

If the user is a registered user, he will see a list of movies that we recommend for him, when he logs in. If the user is a new user, we recommend the most popular movies, based on how other users' ratings of each movie. If the user has rated movies before, using our service, we recommend to the user movies based on how users with similar tastes rated those movies.

When a user clicks on a movie from the list, a screen appears that allows him to rate the movie between one and five stars (and the user can include half a star, if it is added to a positive integer between 1 and 5). We also offer the user a list of genres the movie might fall into and for each of these genres the user can request to be recommended movies with a higher, lower or an equal degree of this genre. For example, we display the word "sad" and near the word "sad" the user has three options to choose from: "less", "more" or "okay". This feedback is used to refine our selection of movies to recommend to the user the next time he opens the main page where he is shown a list of recommendations.

6.2 Screen Images

Image 1

The image shows a hand-drawn wireframe of a user interface. At the top, the text "Best Pictures" is written in a cursive font. To its right are two input fields labeled "Email" and "Password", followed by a "Log In" button. Below this, the text "Sign Up" is written. Under "Sign Up" are four input fields: "First Name", "Last Name", "Email", and "New password". At the bottom is a "Create Account" button.

Image 2

Movies You Might Like

Houses of Games (1987)

Out of Sight (1998)

Killing, The (1956)






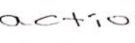




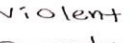

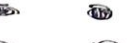







Usual Suspects, The (1995)

Four Rooms (1995)

Black Hawk Down (2001)

Image 3

Black Hawk Down (2001)

					
	1	2	3	4	5
This Movie			less	what I want	more
action					
violent					
comedy					

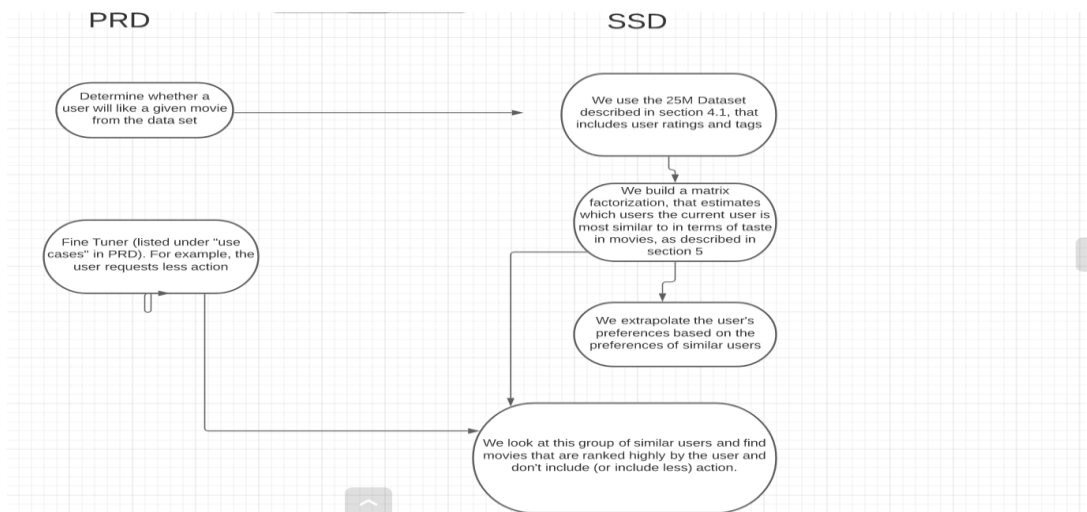
6.3 Screen Objects and Actions

The user starts off on page 1, the sign in page. If the user is already signed up for the service, he can log in. Otherwise he must sign in. Once the user is logged in, he is taken to page 2. Here appears the list of movies we recommend- customized to the user's preferences. The user can select one of these movies and this will take the user to page 3. Here is where the user can provide feedback to the selection of movies we offered.

On page 3, we have the title of the movie, a series of five stars, and a box of genres that the user can request more, less or the same of.

7. REQUIREMENTS MATRIX

Provide a cross-reference that traces components and data structures to the requirements in your SRS document.



Use a tabular format to show which system components satisfy each of the functional requirements from the SRS. Refer to the functional requirements by the numbers/codes that you gave them in the SRS.

PRD Requirement	Component	Dataset
Movie Tuner	Matrix Factorization that helps find similar users	Movie Lens files, "movies" and "ratings" that tell us how users ranked movies and what genres these movies belong to

Appendices may be included, either directly or by reference, to provide supporting details that could aid in the understanding of the Software Design Document.