SOFTWARE REQUIREMENT SPECIFICATION

1 Introduction

1.1 Purpose

The purpose of this project is to demonstrate how recommendation systems use machine learning techniques to recommend relevant information to the users based on their interests and the interests of similar users. This case study focuses mainly on Content-based filtering.

1.2 Scope

The scope of this project is to provide relevant movie/TV series recommendations to the users based on the name of the movie/TV series name that they enter in the search bar. For this purpose, we are going to use Content-based filtering. Content-based filtering is based on a single user's interactions and preference. Recommendations are based on the metadata collected from a user's history and interactions. With an approach like this, the more information that the user provides, the higher the accuracy. A recommendation like 'products similar to this', is a typical instance of this type of approach.

Through this method, the user will be provided with a vast range of recommendations that are similar to their interests, thereby saving the time of the user. The user may also come across new movies/TV series that were previously unknown to them. This way the streaming websites will be able to boost their revenues and ensure high customer satisfaction and retention.

1.3 Definitions, acronyms, and abbreviations

Flask	It is a lightweight python web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications.
Kaggle Dataset	A CSV file taken from Kaggle website that contains all the movie details.
Python	The programming language for the implementation of the web application, retrieving data from dataset and displaying the end results to the user.
User	Any person who uses our application.
Cosine Similarity Function	The feature that helps us generate a similarity to group a list of movies together.

1.4 References

https://medium.com/voice-tech-podcast/a-simple-way-to-explain-the-recommendation-engine-in-ai-d1a609f59d97

https://www.educative.io/edpresso/what-is-content-based-filtering

https://www.geeksforgeeks.org/ml-content-based-recommender-system/

1.5 Document Overview

In this section, we tried to briefly discuss the aim of our project and give an insight on how we plan on approaching the task at hand. We put forward some important definitions that we will mention frequently throughout this document, in the glossary. In the General Overview section, we will talk about our development methodology and the user characteristics. The Software Requirements Specification section will be about external interface requirements(user interfaces, hardware interfaces, etc…), functional requirements, non-functional requirements, performance requirements, software system attributes and finally safety requirements. The document comes to an end with references.

2 General description

2.1 Product perspective

The name of our project is Movie Recommendation System. The user will be provided with the movie recommendations based on the movie that they have searched.

2.2 Product function

The end product will provide movie recommendations to the user when the user enters the title of the movie that they have watched in the search box, they'll be provided with the information regarding the searched movie and 15 movie recommendations based on the searched movie. This way, the user will be able to find movies that are similar to their tastes and movies that they have watched.

2.3 User characteristics

User classes will be Dataset(Kaggle Movies dataset), User, Server.

2.4 General constraints

There are no constraints at this point of time.

2.5 Assumptions and dependencies

We assume that extra documentation beyond this SRS would not be necessary in order for the user to utilize this product.

3 Specific requirements

3.1 Functional requirements

3.1.1 Introduction

Movie Name Search and Cosine Similarity Function are used to provide the recommendations to the user.

3.1.2 Input

A search box will be provided on the homepage to search for a movie by its name (exact name ie, case-sensitive also). The movie name should be typed in the search box provided and select any one of the suggestions displayed. Typing a movie not present in the database or entering a wrong spelling will result in the movie not found error.

3.1.3 Processing

- **Movie Name Search**: When the user enters the movie's name in the search box, a trie based searching is performed on the dataset and displays the top ten suggestions for the user to select. Once, user selects the movie name, its complete details are fetched from the dataset.
- **Cosine Similarity Function:** The tags of the selected movie are sent to the similarity function. The function processes all the tags and creates a similarity to search recommendations. This similarity used again to search for similar movies in the data set.

3.1.4 Output

The details of the selected movie will be displayed below the search box. Below that, the top 15 recommended movies are displayed as a sidebar sorted in descending order of similarity for the selected movie. Along the sidebar, the cast of the selected movie will also be displayed.

3.2 External Interface Requirements

3.2.1 User interface

The interface is kept as simple as possible. The interface does not include any signin or register option. The web application is made as a single page web application. The Home screen consists of a search bar for searching a desired movie from the data set taken from Kaggle. We obtain the search results first and display the main leads of the film. Then, we display the top 15 recommendations for the search result. If we try to search for

3.2.2 Hardware interface

Though not necessarily interfacing with the hardware, the system must make use with an internet connection.

3.2.3 Software interface

Along with the internet connection, the project makes indirect use of an internet browser. Kaggle data set, that is a CSV file taken from https://www.kaggle.com/ acts as our Database.

Operating System: Windows XP/7/8/10, Linux, MacOS.

3.2.4 Communication interfaces

The system uses an internet connection to connect to the dataset.

3.3 Performance Requirements

For optimal performance, the web browser must be of the latest version and the stable internet connection is mandatory.

3.4 Design Constraints

This application is created using python programming language and uses the sklearn, pandas, flask and csv modules.

3.5 Security Requirements

The user directly interacts with the web application without any login/register.

3.6 Maintainability Requirements

No modifications are done on the dataset. Dataset is only used to fetch the data for processing. Maintenance of this dataset is not required.

3.7 Reliability Requirements

For the case study to be reliable, an internet connection with a decent speed is required. The latest versions of Google Chrome, Mozilla Firefox and Edge are preferable.

3.8 Availability Requirements

As it is a web application, it is available at a local host network on the computer with the IP address http://127.0.0.1:5000/ on the browsers.

3.9 Database Requirements

We are going to use the Kaggle Dataset for our project.

3.10 Documentation Requirements

This document is sufficient to learn about the specifications of the product.

3.11 Safety Requirements

The user's data/searches aren't stored anywhere on the website. Therefore the safety and the privacy of the user is ensured.

3.12 Operational Requirements

This document is sufficient to learn about the operations of the product.

3.13 Site Adaptation

This website can be accessed from laptops, desktops and smartphones.