

# **MOVIE RECOMMENDATION SYSTEM**

*Project Report*

*submitted to SARASWATHI NARAYANAN COLLEGE*

*(Affiliated to Madurai Kamaraj University, Madurai)*

*in partial fulfillment of the requirements*

*for the award of the Degree of*

## **BACHELOR OF COMPUTER APPLICATIONS**

**By**

**M. VIGNESHWARAN**

**Reg. No. 21USBC12**

**Under the guidance of**

**MR. R. SURYAPRAKASH M.C.A.,**

**Assistant Professor**



**DEPARTMENT OF COMPUTER APPLICATIONS**

**SARASWATHI NARAYANAN COLLEGE,**

**(Autonomous Institution Affiliated to Madurai Kamaraj University)**

**MADURAI- 625 022.**

**APRIL – 2024**

**MR. R. SURYAPRAKASH M.C.A.,**  
**Assistant professor,**  
**Department of Computer Science,**  
**Saraswathi Narayanan College,**  
**Madurai – 625 022.**

### **CERTIFICATE**

This is to certify that the dissertation entitled “**MOVIE RECOMMENDATION SYSTEM**” is a Bonafede research work carried out by **M. VIGNESHWARAN, (Reg. No. 21USBC12)** under my guidance, during the year 2023 -2024 in partial fulfilment of the requirements for the award of the Degree of Bachelor of Science in Computer Applications.

**Place:**

**Date:**

**Signature of the Guide**

---

**Head of the Department**

**Controller of Examinations**

**Principal**



27th March 2024

## TO WHOM SO EVER IT MAY CONCERN

This is to certify that **Mr. M. Vigneshwaran**, doing final year B.C.A., **Saraswathi Narayanan College**, Madurai, has successfully completed her project in our organization during the period from December 2023 to March 2024.

For Sprout Knowledge Solutions Pvt Ltd

Poornima A

HR – Assistant Manager



## **DECLARATION**

I hereby declare that the dissertation entitled “**MOVIE RECOMMENDATION SYSTEM**” submitted towards the Degree of Bachelor of Computer Applications is my original work and that the dissertation has not formed the basis for the award of any degree, associateship, fellowship or any other similar titles.

**Place:**

**Date:**

Signature of the Candidate

**M. VIGNESHWARAN,  
(Reg. No. 21USBC12)**

| CHAPTER   | CONTENTS                      | PAGE NO   |
|-----------|-------------------------------|-----------|
| <b>1.</b> | <b>INTRODUCTION</b>           | <b>1</b>  |
|           | 1.1 INTRODUCTION              | <b>1</b>  |
|           | 1.2 ABOUT THE PROJECT         | <b>2</b>  |
|           | 1.3 ORGANIZATION PROFILE      | <b>3</b>  |
| <b>2.</b> | <b>SYSTEM ANALYSIS</b>        | <b>4</b>  |
|           | 2.1 EXISTING SYSTEM           | <b>4</b>  |
|           | 2.2 PROPOSED SYSTEM           | <b>5</b>  |
|           | 2.3 FEASIBILITY STUDY         | <b>6</b>  |
|           | 2.3.1 ECONOMICAL FEASIBILITY  | <b>7</b>  |
|           | 2.3.2 TECHNICAL FEASIBILITY   | <b>7</b>  |
|           | 2.3.3 SOCIAL FEASIBILITY      | <b>8</b>  |
|           | 2.3.4 OPERATIONAL FEASIBILITY | <b>8</b>  |
| <b>3.</b> | <b>SYSTEM SPECIFICATION</b>   | <b>9</b>  |
|           | 3.1 HARDWARE REQUIREMENTS     | <b>9</b>  |
|           | 3.2 SOFTWARE REQUIREMENTS     | <b>9</b>  |
| <b>4.</b> | <b>SOFTWARE DESCRIPTION</b>   | <b>10</b> |
|           | 4.1 HTML                      | <b>10</b> |
|           | 4.2 CSS                       | <b>12</b> |
|           | 4.3 JAVASCRIPT                | <b>14</b> |
|           | 4.4 PYTHON                    | <b>16</b> |
|           | 4.5 PYTHON FLASK              | <b>20</b> |
|           | 4.6 GOOGLE COLAB              | <b>21</b> |
|           | 4.7 VISUAL STUDIO CODE        | <b>22</b> |
| <b>5.</b> | <b>MODULE DESCRIPTION</b>     | <b>23</b> |
|           | 5.1 MODULES                   | <b>23</b> |

|            |                                     |           |
|------------|-------------------------------------|-----------|
|            | 5.2 DATA FLOW DIAGRAM               | <b>24</b> |
|            | 5.3 SYSTEM DESIGN AND DEVELOPMENT   | <b>26</b> |
| <b>6.</b>  | <b>CODING DESIGN</b>                | <b>28</b> |
| <b>7.</b>  | <b>SYSTEM TESTING</b>               | <b>45</b> |
|            | 7.1 TESTING                         | <b>45</b> |
|            | 7.2 UNIT TESTING                    | <b>45</b> |
|            | 7.3 INTEGRATION TESTING             | <b>46</b> |
|            | 7.4 VALIDATION TESTING              | <b>46</b> |
|            | 7.5 SYSTEM TESTING                  | <b>46</b> |
| <b>8.</b>  | <b>SYSTEM IMPLEMENTATION</b>        | <b>47</b> |
| <b>9.</b>  | <b>SCREEN LAYOUT AND CONCLUSION</b> | <b>51</b> |
|            | 9.1 SCREEN LAYOUT                   | <b>51</b> |
|            | 9.2 CONCLUSION                      | <b>52</b> |
| <b>10.</b> | <b>FUTURE ENHANCEMENT</b>           | <b>53</b> |
| <b>11</b>  | <b>BIBLIOGRAPHY</b>                 | <b>54</b> |

## **ABSTRACT**

In this hustling world, entertainment is a necessity for each one of us to refresh our mood and energy. Entertainment regains our confidence for work and we can work more enthusiastically. For revitalizing ourselves, we can watch movies of our choice. For watching favourable movies online, we can utilize movie recommendation systems, which are more reliable. Movie Recommendation system helps us to search our preferred movies among all of the different types of movies and hence reduce the trouble of spending a lot of time searching our favourable movies using a multivariate linear regression algorithm. We will analyse the data using the real-time data sets and train the machine using a multivariate linear regression machine learning algorithm. We can recommend the most approximate movie for the persons.

# CHAPTER 1

## INTRODUCTION

### 1.1 Introduction

In today's digital age, the sheer volume of available movies presents both an exciting opportunity and a significant challenge for audiences worldwide. With countless films spanning genres, eras, and cultures, the quest to find the perfect movie can often feel like searching for a needle in a haystack. However, the marriage of machine learning algorithms with the versatile capabilities of Python programming offers a promising solution: personalized movie recommendations.

At the heart of this technological innovation lies the fusion of data science principles and cinematic appreciation. By harnessing the power of Python libraries such as pandas, NumPy, and scikit-learn, alongside advanced machine learning techniques, enthusiasts embark on a journey to create intelligent recommendation systems capable of understanding and predicting individual movie preferences.

Central to this endeavour are collaborative filtering and content-based filtering algorithms, each offering unique insights into user behaviour and movie characteristics. Collaborative filtering analyses user-item interactions to identify patterns and similarities among users, enabling the system to suggest movies based on the preferences of similar users. Meanwhile, content-based filtering examines movie features such as genre, actors, and plot keywords to recommend films that align with the user's past viewing history and stated preferences.

Beyond its technical intricacies, this endeavour underscores the transformative potential of machine learning in enhancing entertainment experiences. By empowering audiences to discover new and exciting films aligned with their interests, these recommendation systems democratize access to cinematic treasures, fostering a deeper appreciation for the art of filmmaking across diverse communities.

In essence, the fusion of Python programming and machine learning algorithms represents a paradigm shift in the way we engage with cinema. It is a testament to the endless possibilities when art converges with technology, offering a glimpse into a future where movie discovery is not merely a search but an immersive and enriching experience tailored to the individual. Join us on this journey as we unlock the secrets of movie recommendation using machine learning with Python, where every film is a discovery waiting to be made.



## **1.2 About The Project**

In this hustling world, entertainment is a necessity for each one of us to refresh our mood and energy. Entertainment regains our confidence for work and we can work more enthusiastically. For revitalizing ourselves, we can watch movies of our choice. For watching favourable movies online, we can utilize movie recommendation systems, which are more reliable. Movie Recommendation system helps us to search our preferred movies among all of the different types of movie s and hence reduce the trouble of spending a lot of time searching our favourable movies using a multivariate linear regression algorithm. We will analyse the data using the real-time data sets and train the machine using a multivariate linear regression machine learning algorithm. We can recommend the most approximate movie for the persons.

### **Languages/Frameworks**

Machine Learning Model: Python, Pandas, Matplotlib, Scikit Learn, Pickle.

### **1.3 Organization Profile**

Sprout knowledge solutions Pvt Ltd is a leading provider of educational technology solutions based in Coimbatore. The company was founded in 2020 with the aim of transforming education by leveraging technology to create innovative and effective learning experiences.

Sprout Knowledge Solutions Pvt Ltd an EdTech Firm that provides Upskilling Courses for Students in Software Coding & Development Programmes across India & various other countries like Australia, the US, France, Saudi Arabia & Far East.

The company is stepping into 11 countries of the African continent. The company Develops propriety content that suits the current & future trends, which will help the student and the professional community navigate the ever-changing future.

Sprout's Courses are designed to achieve the following for the students - Future Ready, Develop Problem Solving Skills, Enhance Collaboration & Creativity, and Develop Communication. Coding Skills are important for kids in the future as these transferable skills enable kids to learn relevant and competitive skills whilst developing a problem-solving mindset from an early age.

Technology awaits invention and this is why at Sprout, we offer computer coding courses for learners in various domains such as Artificial Intelligence, Machine Learning, Web Development, and guiding various Projects. Sprout aims to create engaging and enriching learning experiences that empower the next generation of computer coders, web developers, and makers of the future.

## CHAPTER 2

### SYSTEM ANALYSIS

Machine learning algorithms used for natural language processing (NLP) currently take too long to complete their learning function. This slow learning performance tends to make the model ineffective for an increasing requirement for real time applications such as voice transcription, language translation, text summarization topic extraction and sentiment analysis. Moreover, current implementations are run in an offline batch-mode operation and are unfit for real time needs. Newer machine learning algorithms are being designed that make better use of sampling and distributed methods to speed up the learning performance. In my thesis, I identify unmet market opportunities where machine learning is not employed in an optimum fashion. I will provide system level suggestions and analyses that could improve the performance, accuracy and relevance.

#### 2.1 Existing System

Though we have many recommendation systems in the form of Netflix, amazon prime, etc., Suppose that in a house we have many persons having different mode of watching. Here in the proposed system consider that a single person tries to create a recommendation system.

#### Disadvantages

- **Limited Serendipity:** Content-based systems may struggle to introduce users to entirely new or unexpected content outside their established preferences, as recommendations are based on the features of previously liked movies.
- **Limited Diversity:** Due to their reliance on movie features and user preferences, content-based systems may recommend similar types of movies, potentially leading to a lack of diversity in recommendations.
- **Dependency on Feature Quality:** The effectiveness of content-based systems heavily relies on the quality and granularity of movie features. Inaccurate or incomplete feature data may result in suboptimal recommendations.
- **Difficulty Capturing Subjective Preferences:** Content-based systems may have difficulty capturing subjective aspects of user preferences that are not explicitly represented in movie features, such as mood, atmosphere, or personal sentiment towards specific themes.

## 2.2 Proposed System

The dataset consists of the file `movies_metadata.csv`: The main Movies Metadata file. Contains information on 4803 movies. Features include posters, backdrops, budget, revenue, release dates, languages, production countries and companies. This dataset is an ensemble of data collected from TMDB and Group Lens.

The Movie Details, Credits and Keywords have been collected from the TMDB Open API. This product uses the TMDB API but is not endorsed or certified by TMDB. Their API also provides access to data on many additional movies, actors and actresses, crew members, and TV shows. You can try it for yourself [here](#). We make use of cosine similarity algorithm to build a machine learning model and provide a recommendation system.

The platforms already available provide multiple features such as a collection of hand-chosen cinema available to you through its app and some are picked based on your taste, as well as taste analysis based on your ratings and also by selecting all the movies you've watched from a pretty extensive list and then choosing which movies you want to watch from another list. These apps mainly recommend movies based on ratings. In this project I have collected real-time data from various sources and analysed them using data analysis processes such as data cleaning, data feature engineering and outlier removals. After the data is prepared, we will create a machine-learning model using a multivariate linear regression algorithm. When the user gives a name of his favourite movie name and based on this input we are going to recommend certain movies then.

### Advantages

- **Personalized Recommendations:** Content-based systems offer personalized recommendations tailored to individual users' tastes and preferences. By analysing the features of movies and comparing them to users' profiles, these systems can suggest movies that align closely with users' interests.
- **Independence from User Data:** Content-based systems do not require historical user-item interactions to generate recommendations. This makes them suitable for scenarios where there is limited or no user data available, such as for new users or items (the cold start problem).

## 2.3 Feasibility Analysis

The feasibility of the project is analysed in this phase and the business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis, the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

The feasibility study investigates the problem and the information needs of the stakeholders. It seeks to determine the resources required to provide an information systems solution, the cost and benefits of such a solution, and the feasibility of such a solution. The analyst conducting the study gathers information using a variety of methods, the most popular of **which are:**

- Interviewing users, employees, managers, and customers.
- Developing and administering questionnaires to interested stakeholders, such as potential users of the information system.
- Observing or monitoring users of the current system to determine their needs as well as their satisfaction and dissatisfaction with the current system.
- Collecting, examining, and analysing documents, reports, layouts, procedures, manuals, and any other documentation relating to the operations of the current system.
- Modelling, observing, and simulating the work activities of the current system.

The goal of the feasibility study is to consider alternative information systems solutions, evaluate their feasibility, and propose the alternative most suitable to the organization. The feasibility of a proposed solution is evaluated in terms of its components.

**These components are:**

- Economical Feasibility
- Technical Feasibility
- Social Feasibility
- Operational Feasibility

### **2.3.1 Economical Feasibility**

This study is carried out to check the economic impact that the system will have on the organization. The amount of funds that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### **2.3.2 Technical Feasibility**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available.

**Technical resources:** This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

### **2.3.3 Social Feasibility**

The aspect of the study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity.

The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

### **2.3.4 Operational Feasibility**

The ability, desire, and willingness of the stakeholders to use, support, and operate the proposed computer information system. The stakeholders include management, employees, customers, and suppliers. The stakeholders are interested in systems that are easy to operate, make few, if any, errors, produce the desired information, and fall within the objectives of the organization.

## **CHAPTER 3**

### **SYSTEM SPECIFICATION**

#### **3.1 Hardware Requirement**

OS : Windows 10 or 11  
RAM : Minimum 4GB.  
Processor : i3 and above, Ryzen 3  
Hard Disk : 250 GB

#### **3.2 Software Requirement**

Front End : HTML, CSS, JavaScript.  
Back End : Python.  
Database : Excel  
Tools Used : Google Colab  
Software Used: VS Code

## **CHAPTER 4**

### **SOFTWARE DESCRIPTION**

#### **4.1 HTML**

HTML stands for Hyper Text Markup Language, which is the most widely used language on Web to develop web pages. HTML was created by Berners-Lee in late 1991 but "HTML 2.0" was the first standard HTML specification which was published in 1995. HTML 4.01 was a major version of HTML and it was published in late 1999. Though HTML 4.01 version is widely used currently we are having the HTML-5 version which is an extension to HTML 4.01, and this version was published in 2012.

HTML stands for Hyper Text Markup Language. HTML is the standard markup language for creating Web pages. HTML describes the structure of a Web page. HTML consists of a series of elements. HTML elements tell the browser how to display the content. HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc. Originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers. Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.

HTML is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. I will list down some of the key advantages of learning HTML:

- Create Web site - You can create a website or customize an existing web template if you know HTML well.
- Become a web designer - If you want to start a career as a professional web designer, HTML and CSS designing is a must skill.
- Understand web - If you want to optimize your website, to boost its speed and performance, it is good to know HTML to yield best results.
- Learn other languages - Once you understand the basic of HTML then other related technologies like JavaScript, php, or angular are become easier to understand. Web pages can be created and modified by using professional HTML editors.



- However, for learning HTML we recommend a simple text editor like Notepad (PC) or TextEdit (Mac).
- We believe that using a simple text editor is a good way to learn HTML.
- I'm 100 percent positive you have viewed a website that uses HTML code. How do I know? This article is made up of HTML tags and elements on the backend.
- This browser doesn't display the actual HTML tags but instead uses them to render specific content on the page so that visitors to the web page can view it. Usually, the average website includes several different pages with HTML code, like the homepage, about us page, and contact us page, each with its own unique HTML documents.

## Html Tag

As you may have guessed, this tag is written as: `<html>`. This element wraps all the content on the entire page, and you may hear it be referred to as the *root element*. This tag will be below the document tag and will need to be closed out, using `</html>`, at the end of the web page.

## Title Tag

What is the title of the page you're creating? Whatever it is, will be wrapped in the `<title>` element and will be towards the top of your tags.

```
<title>Page Title</title>
```

## Body Tag

The `<body>` element contains all the content that you want to show the visitors of your web page. This includes text, images, videos, playable audio tracks, and more. It will also need to be closed out, using `</body>`, at the end of the web page.

## Heading Tag

This is written as `<h1>` and it defines a larger heading on the page. You can have more than one heading tag and each element will have a corresponding number.

```
<h1>This is a heading. </h1>
```

```
<h2>This is also a heading. </h2>
```

## Paragraph Tag

The paragraph tag defines which text will be displayed as a paragraph using the element `<p>`.

`<p>`This is a paragraph about dogs. `</p>`

`<p>`This another paragraph about dogs. `</p>`

## Lists

Using the tag `<ul>` will define what text will be written as either a numbered list `<ol>` or a bulleted list `<li>`.

An example of a bulleted list would be:

`<ul>`

`<li>`Flour`</li>`

`<li>`Sugar`</li>`

`<li>`Butter`</li>`

`</ul>`

## 4.2 CSS

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. I will list down some of the key advantages of learning CSS:

- Create Stunning Web site - CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.

- Become a web designer - If you want to start a career as a professional web designer, HTML and CSS designing is a must skill.
- Control web - CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.
- Learn other languages - Once you understand the basic of HTML and CSS then other related technologies like JavaScript, php, or angular are become easier to understand.

## **Applications Of CSS**

As mentioned before, CSS is one of the most widely used style language over the web. I'm going to list few of them here:

- CSS saves time - You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
- Pages load faster - If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So, less code means faster download times.
- Easy maintenance - To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
- Superior styles to HTML - CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
- Multiple Device Compatibility - Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.
- Global web standards - Now HTML attributes are being deprecated and it is being recommended to use CSS. So, it's a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.

## **CSS Syntax**

- The selector points to the HTML element you want to style.
- The declaration block contains one or more declarations separated by semicolons.
- Each declaration includes a CSS property name and a value, separated by a colon.

- Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

## The CSS Id Selector

- The id selector uses the id attribute of an HTML element to select a specific element.
- The id of an element is unique within a page, so the id selector is used to select one unique element!
- To select an element with a specific id, write a hash (#) character, followed by the id of the element.

## 4.3 JavaScript

JavaScript is a high-level programming language primarily used for web development. It's commonly employed to add interactivity and dynamic behavior to web pages. JavaScript allows developers to manipulate elements on a webpage, respond to user actions, and create dynamic content. It's also used in server-side development (Node.js) and for building mobile and desktop applications.

### Applications Of JavaScript

**Web Development:** JavaScript is primarily used for client-side scripting in web development. It enables dynamic content, interactive elements, form validation, animations, and much more on websites.

**Frontend Frameworks And Libraries:** Frameworks like Angular, React, and Vue.js, as well as libraries like jQuery, provide powerful tools for building complex user interfaces and single-page applications (SPAs) using JavaScript.

**Server-Side Development:** With Node.js, JavaScript can be used for server-side programming, allowing developers to build scalable and efficient backend systems, APIs, and server applications.

**Mobile App Development:** Frameworks like React Native and Ionic allow developers to build mobile applications for iOS and Android using JavaScript, providing a way to reuse code across platforms.

**Desktop App Development:** Tools like Electron enable developers to build cross-platform desktop applications using web technologies, including HTML, CSS, and JavaScript.

Game Development: Game engines like Phaser and Three.js leverage JavaScript for creating browser-based games and interactive 3D graphics.

**Data Visualization:** JavaScript libraries such as D3.js and Chart.js are widely used for creating interactive and dynamic data visualizations on the web.

**Browser Extensions:** JavaScript is used to develop browser extensions/add-ons, extending the functionality of web browsers like Chrome, Firefox, and Edge.

**Backend As a Service (BaaS):** Services like Firebase allow developers to build serverless applications by providing backend functionality through APIs, allowing for real-time data synchronization and authentication.

IoT (Internet of Things) With platforms like Johnny-Five and Tessel, JavaScript can be used to program and control IoT devices, enabling interaction with sensors, actuators, and other hardware components.

## History Of JavaScript

JavaScript was created by Brendan Eich in 1995 while he was working at Netscape Communications Corporation. The language was originally named "Mocha" and later renamed "LiveScript" before finally being named JavaScript.

The creation of JavaScript was prompted by Netscape's desire to add interactivity to its web browser, Navigator. At the time, web pages were mostly static, with minimal interactivity. Eich was tasked with creating a scripting language that could be embedded directly into web pages and executed by the browser.

JavaScript was designed in just 10 days and was first released as part of Netscape Navigator 2.0 in December 1995. Its initial purpose was to provide simple scripting capabilities for enhancing web pages with client-side interactivity.

In 1996, Microsoft introduced a similar scripting language called JScript for its Internet Explorer browser. Despite some differences in implementation, both JavaScript and JScript shared a similar syntax and purpose, leading to the standardization efforts.

In November 1996, JavaScript was submitted to the European Computer Manufacturers Association (ECMA) for standardization. This led to the creation of the ECMAScript standard, with JavaScript being the most well-known implementation of this standard.

Over the years, JavaScript evolved significantly, with multiple versions of the ECMAScript standard being released to introduce new features and improvements. Notable versions include ECMAScript 3 (1999), ECMAScript 5 (2009), ECMAScript 6 (2015, also known as ES2015 or ES6), and subsequent yearly updates.

## 4.4 Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

### Python Applications

Python is known for its general-purpose nature that makes it applicable in almost every domain of software development. Python makes its presence in every emerging field. It is the fastest-growing programming language and can develop any application.

#### 4.4.1 Web Applications

We can use Python to develop web applications. It provides libraries to handle internet protocols such as HTML and XML, JSON, Email processing, request, BeautifulSoup, Feedparser, etc. One of Python web-framework named Django is used on **Instagram**. Python provides many useful frameworks, and these are given below:

- Django and Pyramid framework (Use for heavy applications)
- Flask and Bottle (Micro-framework)
- Plone and Django CMS (Advance Content management)

## 4.4.2 Desktop Gui Applications

The GUI stands for the Graphical User Interface, which provides a smooth interaction to any application. Python provides a **Tk GUI library** to develop a user interface. Some popular GUI libraries are given below.

- Tkinter or Tk
- wxWidgetM
- Kivy (used for writing multitouch applications)
- PyQt or Pyside

## 4.4.3 Console-Based Application

Console-based applications run from the command-line or shell. These applications are computer program which are used commands to execute. This kind of application was more popular in the old generation of computers. Python can develop this kind of application very effectively. It is famous for having REPL, which means **the Read-Eval-Print Loop** that makes it the most suitable language for the command-line applications. Python provides many free library or module which helps to build the command-line apps. The necessary **IO** libraries are used to read and write. It helps to parse argument and create console help text out-of-the-box. There are also advance libraries that can develop independent console apps.

## 4.4.4 Software Development

Python is useful for the software development process. It works as a support language and can be used to build control and management, testing, etc.

- **SCons** is used to build control.
- **Buildbot** and **Apache Gumps** are used for automated continuous compilation and testing.
- **Round** or **Trac** for bug tracking and project management.

#### 4.4.5 Scientific And Numeric

This is the era of Artificial intelligence where the machine can perform the task the same as the human. Python language is the most suitable language for Artificial intelligence or machine learning. It consists of many scientific and mathematical libraries, which makes easy to solve complex calculations. Implementing machine learning algorithms require complex mathematical calculation. Python has many libraries for scientific and numeric such as NumPy, Pandas, SciPy, Scikit-learn, etc. If you have some basic knowledge of Python, you need to import libraries on the top of the code. Few popular frameworks of machine libraries are given below.

- SciPy
- Scikit-learn
- NumPy
- Pandas
- Matplotlib

#### 4.4.6 Business Applications

Business Applications differ from standard applications. E-commerce and ERP are an example of a business application. This kind of application requires extensively, scalability and readability, and Python provides all these features.

Oddo is an example of the all-in-one Python-based application which offers a range of business applications. Python provides a **Tryton** platform which is used to develop the business application.

#### 4.4.7 Audio or Video-based Applications

Python is flexible to perform multiple tasks and can be used to create multimedia applications. Some multimedia applications which are made by using Python are **TimPlayer**, **cplay**, etc. The few multimedia libraries are given below.

- Gstreamer
- Pyglet
- QT Phonon



#### **4.4.8 3d Cad Applications**

The CAD (Computer-aided design) is used to design engineering related architecture. It is used to develop the 3D representation of a part of a system. Python can create a 3D CAD application by using the following functionalities.

- Fandango (Popular)
- CAMVOX
- HeeksCNC
- AnyCAD
- RCAM

#### **4.4.9 Enterprise Applications**

Python can be used to create applications that can be used within an Enterprise or an Organization. Some real-time applications are OpenERP, Tryton, Picalo, etc.

#### **4.4.10 Image Processing Application**

Python contains many libraries that are used to work with the image. The image can be manipulated according to our requirements. Some libraries of image processing are given below.

- OpenCV
- Pillow
- SimpleITK

In this topic, we have described all types of applications where Python plays an essential role in the development of these applications. In the next tutorial, we will learn more concepts about Python.

## 4.5 Python Flask

Flask is a lightweight and versatile web framework for Python that provides tools, libraries, and technologies for building web applications. It is known for its simplicity and minimalistic design, making it easy to get started with and flexible for a wide range of web development tasks. Here are some key aspects of Flask:

**Microframework:** Flask is often referred to as a microframework because it keeps the core simple and extensible. It provides only the essential features needed for web development, allowing developers to add functionalities as needed using Flask extensions or third-party libraries.

**Routing:** Flask uses a decorator-based syntax for defining routes. Routes map URLs to Python functions, making it easy to create endpoints for handling HTTP requests.

**Templates:** Flask supports Jinja2 templates, a powerful and designer-friendly template engine. Jinja2 templates allow for easy generation of HTML pages by combining static content with dynamic data from Python code.

**Http Request Handling:** Flask provides convenient methods for handling HTTP requests and accessing request data such as form data, query parameters, request headers, etc.

**Http Response Generation:** Flask allows developers to generate HTTP responses easily using functions like `render_template` for rendering templates, `jsonify` for returning JSON responses, `redirect` for redirecting requests, etc.

**Extensions:** Flask has a rich ecosystem of extensions that provide additional functionalities such as authentication, database integration, form validation, caching, and more. These extensions make it easy to add complex features to Flask applications without reinventing the wheel.

**Integrated Development Server:** Flask comes with a built-in development server that makes it easy to run and test applications locally during development.

**Werkzeug And Jinja2:** Flask is built on top of the Werkzeug WSGI toolkit and Jinja2 template engine. Werkzeug provides low-level utilities for handling HTTP requests, routing, and other

web-related tasks, while Jinja2 offers a powerful and flexible template engine for generating HTML content.

**Restful APIs:** Flask is well-suited for building RESTful APIs due to its lightweight nature and flexibility. Developers can easily create endpoints that return JSON or XML data to be consumed by client applications.

**Community And Documentation:** Flask has a large and active community of developers who contribute to its ecosystem by creating tutorials, plugins, extensions, and providing support. The Flask documentation is comprehensive and well-maintained, making it easy for developers to learn and use Flask effectively.

Flask Tutorial provides the basic and advanced concepts of the Python Flask framework. Our Flask tutorial is designed for beginners and professionals. Flask is a web framework that provides libraries to build lightweight web applications in python. It is developed by Armin Ronacher who leads an international group of python enthusiasts (POCCO). Flask is a web framework that provides libraries to build lightweight web applications in python. It is developed by Armin Ronacher who leads an international group of python enthusiasts (POCCO).

It is based on WSGI toolkit and jinja2 template engine. Flask is considered as a micro framework. It is an acronym for web server gateway interface which is a standard for python web application development. It is considered as the specification for the universal interface between the web server and web application.

## 4.6 Google Colab

### What Is Colab?

Colab, or "Colaboratory", allows you to write and execute Python in your browser, with

- Zero configuration required
- Access to GPUs free of charge
- Easy sharing

Whether you're a student, a data scientist or an AI researcher, Colab can make your work easier.

## Colab And Pandas

Colaboratory, or “Colab” for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser and is especially well suited to machine learning, data analysis and education. More technically, Colab is a hosted Jupyter notebook service that requires no setup to use, while providing access free of charge to computing resources including GPUs.

## 4.7 Visual Studio

**Visual Studio Code** (famously known as **VS Code**) is a free open-source text editor by Microsoft. VS Code is available for Windows, Linux, and macOS. Although the editor is relatively lightweight, it includes some powerful features that have made VS Code one of the most popular development environment tools in recent times.

### Features

VS Code supports a wide array of programming languages from Java, C++, and Python to CSS, Go, and Dockerfile. Moreover, VS Code allows you to add on and even creating new extensions including code linters, debuggers, and cloud and web development support.

The VS Code user interface allows for a lot of interaction compared to other text editors. To simplify user experience, VS Code is divided into five main regions:

- The activity bar
- The side bar
- Editor groups
- The panel
- The status bar

# **CHAPTER 5**

## **MODULE DESCRIPTION**

### **5.1 Modules**

1. Data Set collection
2. Data analysis
3. Building a machine learning model
4. Training the model using training data and testing data
5. Building a recommendation system
6. Testing the accuracy score of the data
7. Developing a prediction function

#### **5.1.1 Data Set Collection**

Gathering relevant data from various sources for analysis.

#### **5.1.2 Data Analysis**

Examining and interpreting the collected data to extract insights and patterns.

#### **5.1.3 Building A Machine Learning Model**

Constructing a mathematical representation of data that can make predictions or decisions.

#### **5.1.4 Training the Model Using Training Data and Testing Data**

Teaching the model to make accurate predictions by adjusting its parameters based on labeled training data, then evaluating its performance on unseen testing data.

#### **5.1.5 Building A Recommendation System**

Creating a system that suggests items or actions to users based on their preferences or behaviour.

### **5.1.6 Testing the Accuracy Score of The Data**

Assessing the performance of the model by comparing its predictions to known outcomes and calculating metrics like accuracy.

### **5.1.7 Developing A Prediction Function**

Creating a function within the model that takes input data and generates predictions or recommendations based on learned patterns.

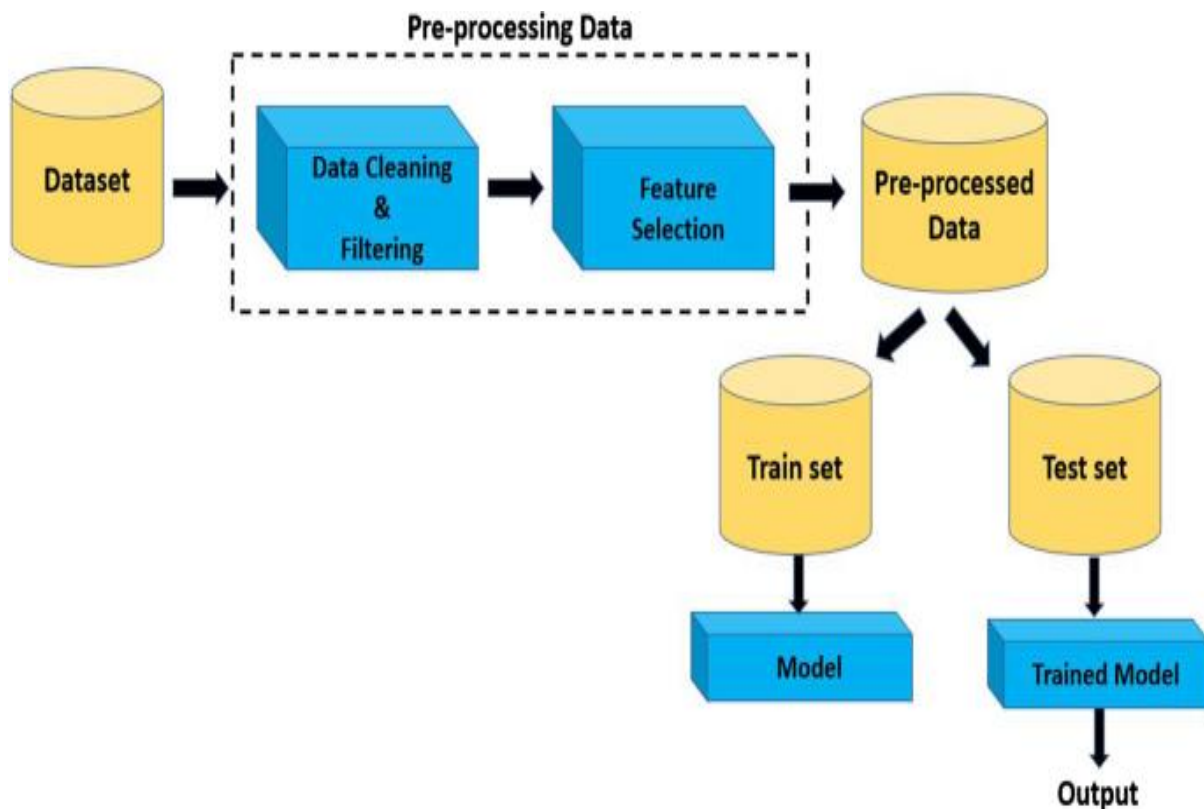
## **5.2 Data Flow**

- Data flow represents the input (or output) of data to (or from) a process ("data in motion"). Data flows only data, not control. Represent the minimum essential data the process needs. Using only the minimum essential data reduces the dependence between processes. Data flows must begin and/or end at a process.
- Data flows are always named. The name is not to include the word "data". Should be given unique names. Names should be some identifying noun. For example, order, payment, complaint.

## **Data Stores**

- Data Stores are repositories for data that are temporarily or permanently recorded within the system. It is an "inventory" of data. These are the common link between data and process models. Only processes may connect with data stores.
- There can be two or more systems that share a data store. This can occur in the case of one system updating the data store, while the other system only accesses the data.
- Data stores are named with an appropriate name, not to include the word "file", Names should consist of plural nouns describing the collection of data. Like customers, orders, and products. These may be duplicated. These are detailed in the data dictionary or with data description diagram.

## Dataflow Diagram



**Fig 5.2 Dataflow Diagram**

## Input Design

- The necessity for the success of a project is the input design. Input design is the process of converting user-originated inputs into machine-readable form. Input specification describes the way in which the data is arranged to enter into the system for processing.
- The goal of input design is to make data entry easier, logical, and error free. The decisions made during the input design are:
  - To provide cost-effective methods of input
  - To achieve the highest possible level of accuracy
  - To ensure that input is understood by the user
- The most important aspect concerned with input design is that the data must be correct. For this input, validation must be performed. Validations are done in the testing phase of the project. The input screen is in such a way that it is user-friendly and easy to use. The input actions performed in this project are

preprocessing for checking the delimiters and new file uploading is given as input in this application

## **Output Design**

- Output design produces the hardcopy regarding the information requested or displays the output in a predefined format. In a web-based application hard copy reports are not mostly generated. Only the user can view the needed information. It is the direct source of information to the end user. Efficient and intelligible outputs improve the system's relationships with the users and help in decision and design making.
- The nature of processing and procedure related to the system were classified and gives the output results. Output from the computer storage is required primarily to communicate the result of processing to the users and provide a permanent copy of the result for later reference.
- The outputs generated are similarity record already available and confirms whether the users to take duplicates or not. The existing record with the attributes is listed to avoid duplication and memory wastage.

## **Database Design**

- Store and sync data with our NoSQL cloud database. Data is synced across all clients in Realtime and remains available when your app goes offline. The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in Realtime to every connected client. When you build cross-platform apps with our Apple platforms, Android, and JavaScript SDKs, all of your clients share one Realtime Database instance and automatically receive updates with the newest data.

## **5.3 System Design And Development**

### **External Entities**

- External entities determine the system boundary. They are external to the system being studied. They are often beyond the area of influence of the developer.
- These can represent another system or subsystem. These go on margins/edges of data flow diagram. External entities are named with appropriate name.



## **Processes**

- Processes are work or actions performed on incoming data flows to produce outgoing data flows. These show data transformation or change. Data coming into a process must be “worked on” or transformed in some way. Thus, all processes must have inputs and outputs. In some (rare) cases, data inputs or outputs will only be shown at more detailed levels of the diagrams. Each process is always “running” and ready to accept data.
- The major functions of processes are computations and making decisions. Each process may have dramatically different timing: yearly, weekly, and daily.

## **Naming Processes**

- Processes are named with one carefully chosen verb and an object of the verb. There is no subject. The name is not to include the word “process”. Each process should represent one function or action. If there is an “and” in the name, you likely have more than one function (and process). For example, get invoice, update customer and create Order Processes are numbered within the diagram as convenient. Levels of detail are shown by decimal notation. For example, top level process would be Process 14, next level of detail Processes 14.1-14.4, and next level with Processes 14.3.1-14.3.6. Processes should generally move from top to bottom and left to right.

## CHAPTER 6

### CODING DESIGN

#### App.py

```
import flask
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
app = flask.Flask(__name__, template_folder='templates')
df2 = pd.read_csv('tmdb.csv')
tfidf = TfidfVectorizer(stop_words='english', analyzer='word')
#Construct the required TF-IDF matrix by fitting and
transforming the data
tfidf_matrix = tfidf.fit_transform(df2['soup'])
print(tfidf_matrix.shape)
#construct cosine similarity matrix
cosine_sim = cosine_similarity(tfidf_matrix, tfidf_matrix)
print(cosine_sim.shape)
df2 = df2.reset_index()
indices = pd.Series(df2.index,
index=df2['title']).drop_duplicates()
# create array with all movie titles
all_titles = [df2['title'][i] for i in
range(len(df2['title']))]
def get_recommendations(title):
    # Get the index of the movie that matches the title
    idx = indices[title]
    # Get the pairwise similarity scores of all movies with
that movie
    sim_scores = list(enumerate(cosine_sim[idx]))
    # Sort the movies based on the similarity scores
    sim_scores = sorted(sim_scores, key=lambda x: x[1],
reverse=True)
    # Get the scores of the 10 most similar movies
```

```

sim_scores = sim_scores[1:11]
# print similarity scores
print("\n movieId      score")
for i in sim_scores:
    print(i)
# Get the movie indices
movie_indices = [i[0] for i in sim_scores]
# return list of similar movies
return_df = pd.DataFrame(columns=['Title'])
return_df['Title'] = df2['title'].iloc[movie_indices]
return return_df
# Set up the main route
@app.route('/', methods=['GET', 'POST'])
def main():
    if flask.request.method == 'GET':
        return(flask.render_template('index.html'))
    if flask.request.method == 'POST':
        m_name = "
".join(flask.request.form['movie_name'].title().split())
#check =
difflib.get_close_matches(m_name,all_titles,cutout=0.50,n=1)
        if m_name not in all_titles:

return(flask.render_template('notFound.html',name=m_name))
        else:
            result_final = get_recommendations(m_name)
            print(result_final) # print the result to check
            names = []
            for i in range(len(result_final)):
                names.append(result_final.iloc[i][0])
            return
flask.render_template('found.html',movie_names=names,search_name=m_name)
if __name__ == '__main__':

```

```
app.run(debug=False)
#app.run()
```

## Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Movie Recommender</title>
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <link                                rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/normalize/5.0.0/n
ormalize.min.css">
  <link                                rel='stylesheet'
href='https://fonts.googleapis.com/css?family=Poppins:300,500&
amp;display=swap'>
  <style>
    :root {
      --primary-color: #000000;
      --secondary-color: #FFFFFF;
      --text-color: #000000;
      --input-border-color: #C0C0C0;
    }
    header {
      background-color: #333;
      color: #fff;
      text-align: center;
      padding: 20px 100;
      position: fixed;
      width: 100%;
      left: 0;
      top: 0;
    }
```

```

html, body {
  margin: 0;
  padding: 0;
  height: 100%;
}
body {
  font-family: "Poppins", sans-serif;
  font-weight: 300;
  color: var(--text-color);
  background-color: #FFFFFF0;
  display: flex;
  justify-content: center;
  align-items: center;
}
.container {
  max-width: 500px;
  padding: 20px;
  border-radius: 8px;
  background-color: #C0C0C0;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}
.background-container {
background-image: url("mob.jpeg");
background-size: cover;
background-position: center;
padding: 50px;
color: white;
text-align: center;
}
h1, h2 {
  text-align: center;
  margin-bottom: 20px;
}

```

```

}
.movie-form {
  text-align: center;
}
.movie-input {
  width: calc(100% - 22px);
  padding: 10px;
  border: 1px solid var(--input-border-color);
  border-radius: 5px;
  margin-bottom: 20px;
  font-size: 16px;
}
.btn {
  width: calc(50% - 5px);
  padding: 10px;
  border-radius: 5px;
  color: #000000;
  background-color: var(--secondary-color);
  border: none;
  cursor: pointer;
  font-size: 16px;
  transition: background-color 0.3s ease;
}
.btn:hover {
  background-color: #0F52BA;
}
.creds {
  text-align: center;
  margin-top: 20px;
}
.creds a {
  color: var(--secondary-color);
}

```

```

        text-decoration: none;
    }
    .creds a:hover {
        text-decoration: underline;
    }
    footer {
        background-color: #333;
        color: #fff;
        text-align: center;
        padding: 10px 0;
        position: fixed;
        width: 100%;
        bottom: 0;
    }
</style>
</head>
<body>
<div class="header">
    <header>
        <h1>Movie Recommendation System</h1>
    </header>
</div>
<div class="background-container">
</div>
<div class="container">
    <h1>Popcorn ready? <br> Let's find your next movie!</h1>
    <div class="movie-form">
        <form action="{ { url_for('main') } }" method="POST">
            <input type="text" class="movie-input" name="movie_name"
placeholder="Enter a movie name" autocomplete="off"
autocorrect="off" autocapitalize="off" required>
            <button type="submit" class="btn">Submit</button>
            <button type="reset" class="btn">Reset</button>
        </form>
    </div>
</div>

```

```

        </form>
    </div>
    <p class="creds">Type the similar movies you want to find.</p>
</div>
<footer>
    <p>&copy; 2024 Movie Recommendation System</p>
</footer>
</body>
</html>

```

## Found.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Movie Recommendation System</title>
    <link rel="stylesheet" href="styles.css">
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
        }
        header {
            background-color: #333;
            color: #fff;
            padding: 20px 0;
            text-align: center;
        }
    </style>

```



```

.container {
    max-width: 1200px;
    margin: 20px auto;
    padding: 0 20px;
}

.movie {
    border: 1px solid #ddd;
    border-radius: 5px;
    padding: 10px;
    margin-bottom: 20px;
}

.movie h3 {
    margin-top: 10px;
}

button{
margin-left: 10vh;
margin-top: 3vh;
}

footer {
    background-color: #333;
    color: #fff;
    text-align: center;
    padding: 10px 0;
    position: fixed;
    width: 100%;
    bottom: 0;
}

</style>
</head>
<body>
<header>

```

```

    <h1>Movie Recommendation System</h1>
</header>

<section class="container">

    <h2>We've uncovered the ultimate movie match for "{{
search_name }}"!</h2>

    <div class="movie">

        <h3>{{ movie_names[0] }}</h3>
    </div>

    <div class="movie">

        <h3>{{ movie_names[1] }}</h3>
    </div>

    <div class="movie">

        <h3>{{ movie_names[2] }}</h3>
    </div>

    <div class="movie">

        <h3>{{ movie_names[3] }}</h3>
    </div>

    <div class="movie">

        <h3>{{ movie_names[4] }}</h3>
    </div>

    <div class="movie">

        <h3>{{ movie_names[5] }}</h3>
    </div>

    <div class="movie">

        <h3>{{ movie_names[6] }}</h3>
    </div>

    <div class="movie">

        <h3>{{ movie_names[7] }}</h3>
    </div>

    <div class="movie">

        <h3>{{ movie_names[8] }}</h3>

```

```

</div>

<div class="movie">
    <h3>{{ movie_names[9] }}</h3>
</div>

<!-- Add more recommended movies here -->

</section>

<button id="back_btn" onClick={window.history.back()}>Back
to Home Page</button>

<footer>
    <p>&copy; 2024 Movie Recommendation System</p>
</footer>
</body>
</html>

```

### **NOTFOUND.html:**

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Error - Movie Not Found</title>
    <style>
        body {
            font-family: "Poppins", sans-serif;
            font-weight: 300;
            font-size: 16px;
            background-color: #f0f0f0;
            color: #333;
            margin: 0;
            padding: 0;
        }
    </style>

```

```

header {
    background-color: #333;
    color: #fff;
    text-align: center;
    padding: 20px 100;
    position: fixed;
    width: 100%;
    left: 0;
    top: 0;
}

.container {
    max-width: 800px;
    margin: 50px auto;
    padding: 20px;
    background-color: #fff;
    border-radius: 8px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

h1 {
    font-size: 24px;
    margin-bottom: 20px;
    color: #fff;
}

h2 {
    font-size: 24px;
    margin-bottom: 20px;
    color: #000000;
}

input[type="text"] {
    width: 100%;
    padding: 10px;
    margin-bottom: 20px;
}

```

```

        border: 1px solid #ddd;
        border-radius: 5px;
        box-sizing: border-box;
    }
    button {
        padding: 10px 20px;
        background-color: #1e90ff;
        color: #fff;
        border: none;
        border-radius: 5px;
        cursor: pointer;
        text-align: center
        ;
    }
    button:hover {
        background-color: #0066cc;
    }
    hr {
        border: none;
        border-top: 1px solid #ddd;
        margin: 20px 0;
    }
    footer {
        background-color: #333;
        color: #fff;
        text-align: center;
        padding: 10px 0;
        position: fixed;
        width: 100%;
        bottom: 0;
    }
</style>

```

```

</head>

<body>

    <div class="container">

        <h2>Error: movie "{{ name }}" not found</h2>

        <input type="hidden" value="{{ name }}"
id="movie_name" readonly />

        <hr>

        <span id="suggestions"></span>

    </div>

<button id="back_btn" onClick="goBack()">Go Back to Home Page
&amp; Try Again.</button>

<script>

    let all_titles = ["Avatar", "Pirates of the Caribbean:
At World's End", "Spectre", "The Dark Knight Rises", "John
Carter", "Spider-Man 3", "Tangled", "Avengers: Age of Ultron",
"Harry Potter and the Half-Blood Prince", "Batman v Superman:
Dawn of Justice", "Superman Returns", "Quantum of Solace",
"Pirates of the Caribbean: Dead Man's Chest", "The Lone
Ranger", "Man of Steel", "The Chronicles of Narnia: Prince
Caspian", "The Avengers", "Pirates of the Caribbean: On
Stranger Tides", "Men in Black 3", "The Hobbit: The Battle of
the Five Armies", "The Amazing Spider-Man", "Robin Hood", "The
Hobbit: The Desolation of Smaug", "The Golden Compass", "King
Kong", "Titanic", "Captain America: Civil War", "Battleship",
"Jurassic World", "Skyfall", "Spider-Man 2", "Iron Man 3",
"Supporting Characters", "Absentia", "The Brothers McMullen",
"The Dirties", "Gabriela", "Tiny Furniture", "Hayride", "The
Naked Ape", "Counting", "The Call of Cthulhu", "Bending
Steel", "The Signal", "The Image Revolution", "This Is Martin
Bonner", "A True Story", "George Washington", "Smiling Fish &
Goat On Fire", "Dawn of the Crescent Moon", "Raymond Did It",
"The Last Waltz", "Run, Hide, Die", "The Exploding Girl", "The
Legend of God's Gun", "Mutual Appreciation", "Her Cry: La
Llorona Investigation", "Down Terrace", "Clerks", "Pink
Narcissus", "Funny Ha Ha", "In the Company of Men", "Manito",
"Rampage", "Slacker", "Dutch Kills", "Dry Spell", "Flywheel",
"Backmask", "The Puffy Chair", "Stories of Our Lives",
"Breaking Upwards", "All Superheroes Must Die", "Pink
Flamingos", "Clean", "The Circle", "Tin Can Man", "Cure", "On
The Downlow", "Sanctuary: Quite a Conundrum", "Bang",
"Primer", "Cavite", "El Mariachi", "Newlyweds", "Signed,
Sealed, Delivered", "Shanghai Calling", "My Date with Drew",]

    function goBack() {

        window.history.back();

```

```

    }
    //main function to check for similarity
    function checkSimilarity() {
        filtered_names = []
        //get user input data
        var input =
document.getElementById("movie_name").value;
        //compare user input data with all titles
        for (var i = 0; i < all_titles.length; i++) {
            let simscore= similarity(input, all_titles[i])
            //save to array if similarity >= 0.8
            if (simscore >= 0.8) {
                console.log("movie: ", all_titles[i],"
similarity score: ",simscore)
                filtered_names.push([all_titles[i] + '<br
/>', simscore]);
            }
        }
        // sort according to similarity scores
        filtered_names.sort((a,b)=>b[1]-a[1])
        // code for website ui
        if (filtered_names.length < 1) {

document.getElementById("suggestions").innerHTML = "The
entered movie name is either misspelled badly or not recorded
in database. <br/><br/>";

        } else {
            let didyoumean=""
            for (key in filtered_names) {
                didyoumean=
didyoumean.concat(filtered_names[key][0])
            }

document.getElementById("suggestions").innerHTML =
"<strong>Did you mean:
</strong><br/><br/>"+didyoumean+"<br/>";

```

```

    }
}
//subfunction to check for similarity
function similarity(input, recc){
    //find frequency of each letter for the input
    const termfreqInput = termFreqMap(input);
    //find frequency of each letter for all titles
    const termfreqRecc = termFreqMap(recc);
    let dict = {};
    //add letters to dictionary
    addLettersToDict(termfreqInput, dict);
    //add letters to dictionary
    addLettersToDict(termfreqRecc, dict);
    //vectorise each letter in the input according to
dictionary
    const inputVector =
vectorizeTermFreqMap(termfreqInput, dict);
    //vectorise each letter in all titles according to
dictionary
    const reccVector =
vectorizeTermFreqMap(termfreqRecc, dict);
    //return cosine similarity matrix for the input
title and all titles
    return dotProduct(inputVector, reccVector) /
(magnitude(inputVector) * magnitude(reccVector));
}
//vectorise each letter in given string
function termFreqMap(str) {
    var chars = str.split('');
    var termFreq = {};
    chars.forEach(function (w) {
        termFreq[w] = (termFreq[w] || 0) + 1;
    });
    return termFreq;
}

```



```

}
//add letters to dictionary
function addLettersToDict(map, dict) {
    for (var key in map) {
        dict[key] = true;
    }
}
//vectorise each letter according to dictionary
function vectorizeTermFreqMap(map, dict) {
    var termFreqVector = [];
    for (var term in dict) {
        termFreqVector.push(map[term] || 0);
    }
    return termFreqVector;
}
//return dot product of two vectors
function dotProduct(vecA, vecB) {
    var product = 0;
    for (var i = 0; i < vecA.length; i++) {
        product += vecA[i] * vecB[i];
    }
    return product;
}
//return magnitude of vector
function magnitude(vec) {
    var sum = 0;
    for (var i = 0; i < vec.length; i++) {
        sum += vec[i] * vec[i];
    }
    return Math.sqrt(sum);
}
//call main function to check for similarity

```

```
        checkSimilarity()
    </script>
    <div class="header">
        <header>
            <h1>Movie Recommendation System</h1>
        </header>
    </div>
    <footer>
        &copy; 2024 Movie Suggestions
    </footer>
</body>
</html>
```

# **CHAPTER 7**

## **SOFTWARE TESTING**

### **7.1 Testing**

The most important phase in system development life cycle is system testing. The number and nature of errors in a newly designed system depends on the system specifications and the time frame given for the design.

A newly designed system should have all the subsystems working together, but in reality, each subsystems work independently. During this phase, all the subsystems are gathered into one pool and tested to determine whether it meets the user requirements.

Testing is done at two level -Testing of individual modules and testing the entire system. During the system testing, the system is used experimentally to ensure that the software will run according to the specifications and in the way the user expects. Each test case is designed with the intent of finding errors in the way the system will process it.

Testing plays a very critical role in determining the reliability and efficiency of software and hence is a very important stage in software development. Software testing is done at different levels. They are the unit testing and system testing which comprises of integration testing and acceptance testing.

### **Types Of Testing**

#### **7.2 Unit Testing**

This is the first level of testing. The different modules are tested against the specifications produced during the integration. This is done to test the internal logic of each module. Those resulting from the interaction between modules are initially avoided. The input received and output generated are also tested to see whether it falls in the expected range of values. Unit testing is performed from the bottom up, starting with the smallest and lowest modules and proceeding one at a time.

The units in a system are the modules and routines that are assembled and integrated to perform a specific function. The programs are tested for the correctness of logic applied and the detection of errors in coding. Each of the modules was tested and errors are rectified. They were then found to function

### **7.3 Integration Testing**

In Integration testing, the tested modules are combined into sub-systems, which are then tested. The goal of integration testing is to check whether the modules can be integrated properly emphasizing the interfaces between modules. The different modules were linked together, and integration testing is done on them.

### **7.4 Validation Testing**

The objective of the validation test is to tell the user about the validity and reliability of the system. It verifies whether the system operates as specified and whether the integrity of important data is maintained. User motivation is very important for the successful performance of the system.

All the modules were tested individually using both test data and live data. After each module was ascertained that it was working correctly, and it had been "integrated" with the system. Again, the system was tested as a whole. We hold the system tested with different types of users. The System Design, Data Flow Diagrams, procedures etc. were well documented so that the system can be easily maintained and upgraded by any computer professional at a later

### **7.5 System Testing**

The integration of each module in the system is checked during this level of testing. The objective of system testing is to check if the software meets its requirements. System testing is done to uncover errors that were not found in earlier tests. This includes forced system failures and validation of the total system as the user in the operational environment implements it. Under this testing, low volumes of transactions are generally based on live data. This volume is increased until the maximum level for each transactions type is reached. The total system is also tested for recovery after various major failures to ensure that no data are lost during the breakdown.

## **CHAPTER 8**

### **SYSTEM IMPLEMENTATION**

Implementation is the process that actually yields the lowest-level system elements in the system hierarchy (system breakdown structure). The system elements are made, bought, or reused. Production involves the hardware fabrication processes of forming, removing, joining, and finishing; or the software realization processes of coding and testing; or the operational procedures development processes for operators' roles. If implementation involves a production process, a manufacturing system which uses the established technical and management processes may be required.

The purpose of the implementation process is to design and create (or fabricate) a system element conforming to that element's design properties and/or requirements. The element is constructed employing appropriate technologies and industry practices. This process bridges the system definition processes and the integration process.

System Implementation is the stage in the project where the theoretical design is turned into a working system. The most critical stage is achieving a successful system and in giving confidence on the new system for the user that it will work efficiently and effectively. The existing system was long time process.

The proposed system was developed using .Net the existing system caused long time transmission process but the system developed now has a very good user-friendly tool, which has a menu-based interface, graphical interface for the end user. After coding and testing, the project is to be installed on the necessary system. The executable file is to be created and loaded in the system. Again, the code is tested in the installed system. Installing the developed code in system in the form of executable file is implementation.

#### **Traditional Programming Vs Machine Learning**

Traditional programming means the usual programming we use to solve a problem such as C, C++, Java, Python etc... In traditional programming, we try to define the rules for a particular problem and try to solve it. For example, you can refer to the below image.

In the above figure you can see the traditional programming means we can define the rules and we will give the data as the input and the rules will take the user input and provide the required answer.

## Machine Learning:



Fig 8.1 Machine Learning

## Recommendation Systems

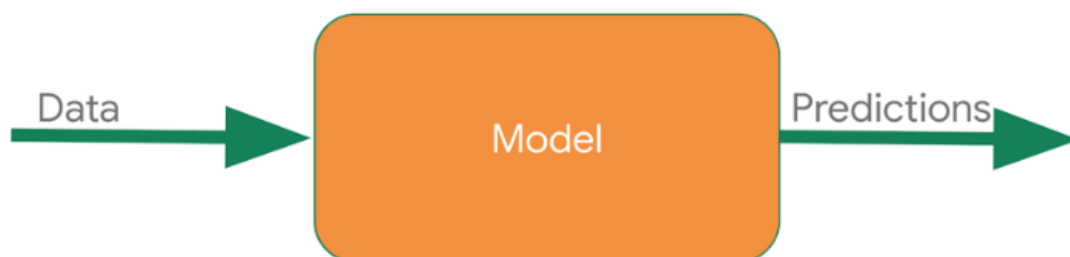


Fig 8.2 Recommendation Systems Model

Recommendation system is an information filtering technique, which provides users with information, which he/she may be interested in.

Examples:



Fig 8.3 Recommendation Systems Example



Fig 8.4 Collaborative Recommendation Systems

## What Defines a Good Recommendation?

Identifying what defines a good recommendation is a problem in itself that many companies struggle with. This definition of “good” recommendations help evaluates the performance of the recommender you built. The quality of a recommendation can be assessed through various tactics which measure coverage and accuracy. Accuracy is the fraction of correct recommendations out of total possible recommendations while coverage measures the



Fig 9.1.1 Traditional Programming

fraction of objects in the search space the system is able to provide recommendations for. The method of evaluation of a recommendation is solely dependent on the dataset and approach used to generate the recommendation. Recommender systems share several conceptual similarities with the classification and regression modelling problem. In an ideal situation, you would want to see how real users react to recommendations and track metrics around the user to improve your recommendation

## **Implementation**

There are three main recommendation system techniques available in Machine Learning.

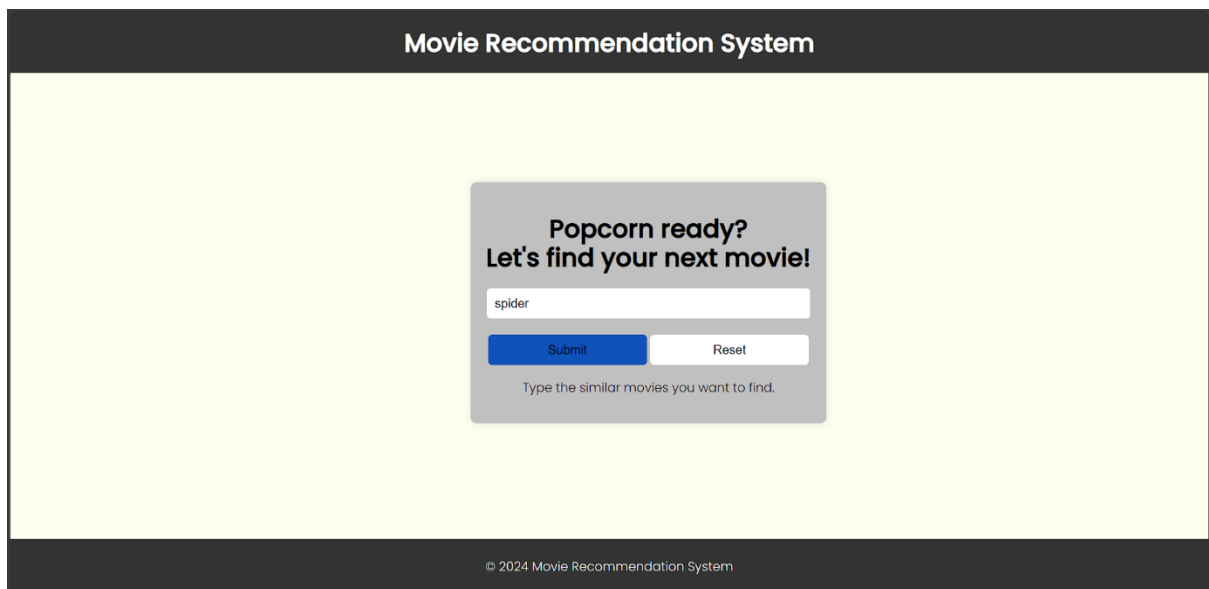
1. Content based recommendation system
2. Popularity based recommendation system
3. Collaborative recommendation system



# CHAPTER 9

## SCREEN LAYOUT AND CONCLUSION

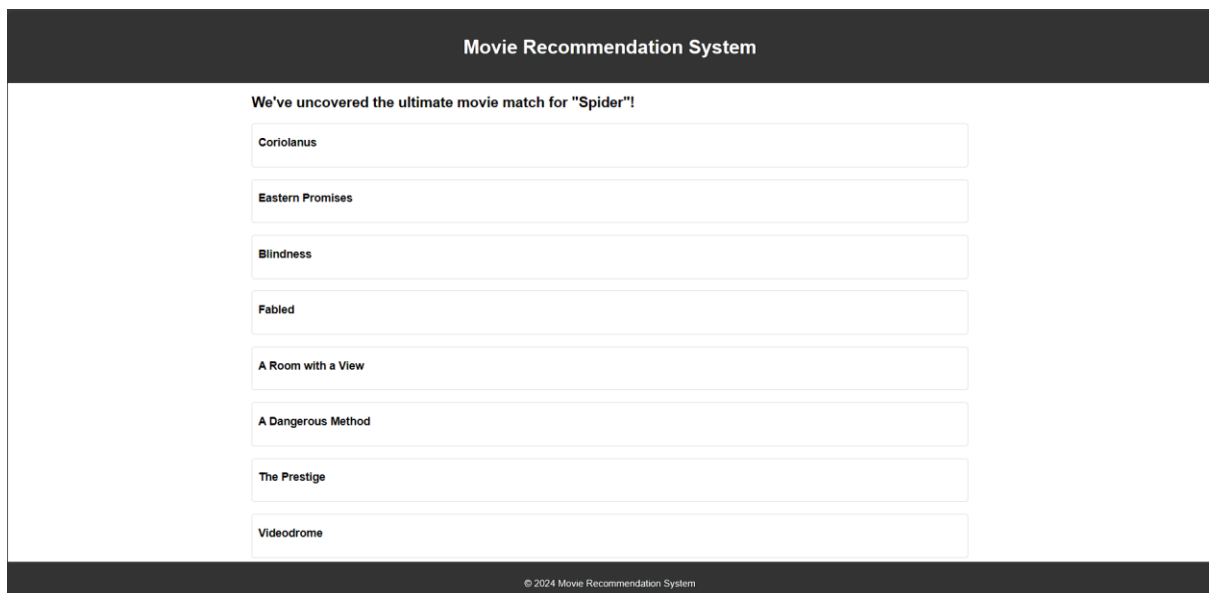
### 9.1.1 Index Screen



The Index Screen features a dark grey header with the text "Movie Recommendation System". The main content area has a light yellow background. In the center, there is a grey rounded rectangle containing the text "Popcorn ready? Let's find your next movie!". Below this text is a search input field with the word "spider" entered. To the right of the input field are two buttons: a blue "Submit" button and a white "Reset" button. Below the buttons, there is a small text prompt: "Type the similar movies you want to find." At the bottom of the screen, there is a dark grey footer with the copyright notice "© 2024 Movie Recommendation System".

Fig 9.1.1 Index Screen

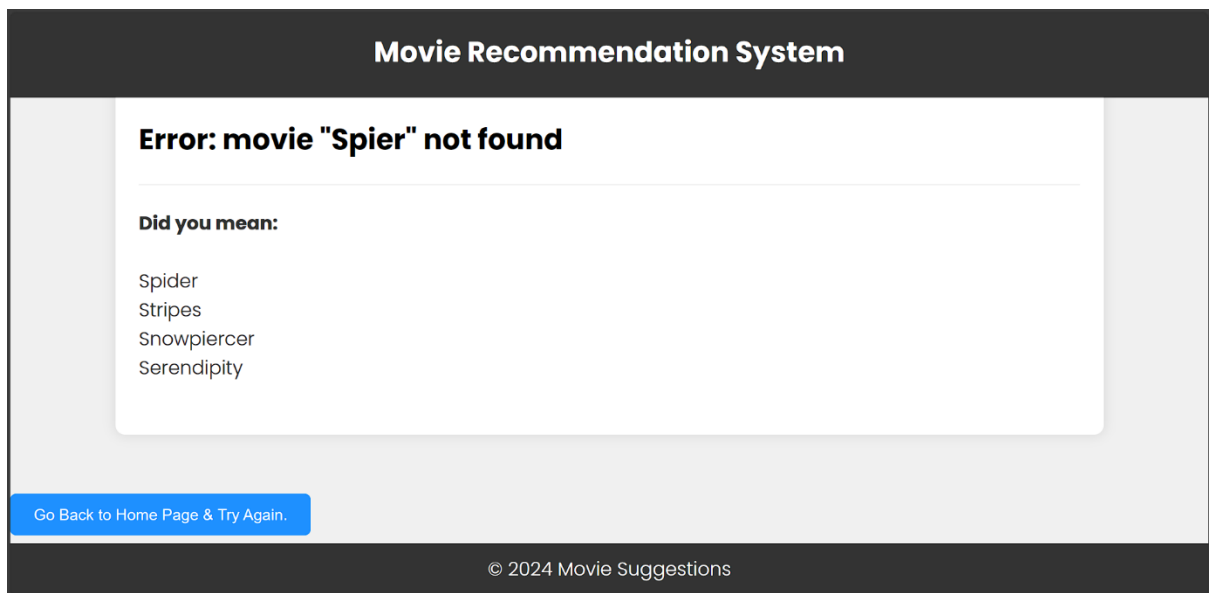
### 9.1.2 Suggestion Screen



The Suggestion Screen features a dark grey header with the text "Movie Recommendation System". The main content area has a light yellow background. At the top, there is a text prompt: "We've uncovered the ultimate movie match for 'Spider'!". Below this prompt is a list of eight movie titles, each in a white rounded rectangle: "Coriolanus", "Eastern Promises", "Blindness", "Fabled", "A Room with a View", "A Dangerous Method", "The Prestige", and "Videodrome". At the bottom of the screen, there is a dark grey footer with the copyright notice "© 2024 Movie Recommendation System".

Fig 9.1.2 Suggestion Screen

### 9.1.3 Error Screen



**Fig 9.1.2 Error Screen**

## **9.2 Conclusion**

When the user searches for a movie that he/she has already watched the Movies Recommendation System will recommend list of movies that are most similar to the searched movie. All these features will save user's time which otherwise would have been wasted on finding a movie that he/she may or may not like. Every month several movies are released, the movies database only gets bigger and bigger. This would help the system to provide a more accurate recommendation to the user and in turn increase customer satisfaction.

## **CHAPTER 10**

### **FUTURE ENHANCEMENT**

In the proposed approach, it has considered Genres of movies but, in future we can also consider age of user as according to the age movie preferences also changes, like for example, during our childhood we like animated movies more as compared to other movies. There is a need to work on the memory requirements of the proposed approach in the future.

# CHAPTER 11

## BIBLIOGRAPHY

1. Kumar Manoj, D.K Yadav, Ankur Singh and Vijay Kr. Gupta, "A Movie Recommender System: MOVREC", International Journal of Computer Applications (0975 – 8887), vol. 124, no. 3, August 2015.
2. Luis M. de Campos, Juan M. Fernandez-Luna, Juan F. Huete and Miguel A. Rueda-Morales, "Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks", International Journal of Approximate Reasoning, 2010.
3. Harpreet Kaur Virk and Er. Maninder Singh, "Analysis and Design of Hybrid Online Movie Recommender System", International Journal of Innovations in Engineering and Technology (IJIET), vol. 5, no. 2, April 2015.
4. Utkarsh Gupta and Nagamma Patil, "Recommender System Based on Hierarchical Clustering Algorithm Chameleon", 2015 IEEE International Advance Computing Conference (IACC).
5. Urszula Kuzelewska, "Clustering Algorithms in Hybrid Recommender System on MovieLens Data", Studies in Logic Grammar and Rhetoric, 2014.
6. Costin-Gabriel Chiru, Vladimir-Nicolae Dinu, Ctina Preda and Matei Macri, "Movie Recommender System Using the User's Psychological Profile", IEEE International Conference on ICCP, 2015.
7. Nicolas Hug, Surprise a Python library for recommender systems, [online] Available: <http://surpriselib.com>.
8. S. R. S. Reddy, S. Nalluri, S. Kunisetti, S. Ashok and B. Venkatesh, "Content-based movie recommendation system using genre correlation" in Smart Intelligent Computing and Applications, Singapore:Springer, pp. 391-397, 2019.
9. A Van den Oord, S Dieleman and B Schrauwen, "Deep content-based music recommendation", Advances in neural information processing systems, pp. 2643-2651, 2013.
10. P. Mathew, B. Kuriakose and V. Hegde, "Book Recommendation System through content based and collaborative filtering method", 2016 International Conference on Data Mining and Advanced Computing (SAPIENCE), pp. 47-52, 2016.