AN INDUSTRIAL ORIENTED MINI PROJECT REPORT ON

PRODUCT RECOMMENDATION SYSTEM

Submitted in partial fulfillment of requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

Submitted by

Gurrapu Manognya	**20BD1A05CM**
Konte Sadhvika	**20BD1A05CW**
Nidumolu Sri Poorvaja	**20BD1A05DB**
Yachavaram Sri Sai Rithvika Reddy	**20BD1A05DV**

Under the Guidance of

P. Niranjan Kumar
Assistant Professor, Department of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY
(AN AUTONOMOUS INSTITUTE)
Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH.
Narayanaguda, Hyderabad, Telangana-29

2023-24

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTION)
Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH.
Narayanaguda, Hyderabad, Telangana-29

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

CERTIFICATE

This is to certify that this a bonafide record of the project report titled **"Product Recommendation System"** which is being presented as the Industrial Oriented Mini Project report by

GURRAPU MANOGNYA	**20BD1A05CM**
KONTE SADHVIKA	**20BD1A05CW**
NIDUMOLU SRI POORVAJA	**20BD1A05DB**
YACHAVARAM SRI SAI RITHVIKA REDDY	**20BD1A05DV**

In partial fulfillment for the award of the degree Bachelor of Technology in Computer Science and Engineering affiliated to Jawaharlal Nehru Technological University Hyderabad, during the academic year 2023 - 2024.

Faculty Supervisor
P. Niranjan Kumar

Head of Department
(Mr. P. Upendar)

Submitted for the Viva Voce Examination held on _____

External Examiner

Vision & Mission of KMIT

Vision of KMIT

- Producing quality graduates trained in the latest technologies and related tools and striving to make India a world leader in software and hardware products and services.
- To achieve academic excellence by imparting in depth knowledge to the students, facilitating research activities and catering to the fast owing and ever- changing industrial demands and societal needs.

Mission of KMIT

- To provide a learning environment that inculcates problem solving skills, professional, ethical responsibilities, lifelong learning through multi modal platforms and prepares students to become successful professionals.
- To establish an industry institute Interaction to make students ready for the industry.
- To provide exposure to students on latest hardware and software tools.
- To promote research based projects/activities in the emerging areas of technology convergence.
- To encourage and enable students to not merely seek jobs from the industry but also to create new enterprises.
- To induce a spirit of nationalism which will enable the student to develop, understand India's challenges and to encourage them to develop effective solutions.
- To support the faculty to accelerate their learning curve to deliver excellent service to students.

Vision & Mission of CSE

Vision of the CSE

To be among the region's premier teaching and research Computer Science and Engineering departments producing globally competent and socially responsible graduates in the most conducive academic environment**.**

Mission of the CSE

- To provide faculty with state of the art facilities for continuous professional development and research, both in foundational aspects and of relevance to emerging computing trends.
- To impart skills that transform students to develop technical solutions for societal needs and inculcate entrepreneurial talents.
- To inculcate an ability in students to pursue the advancement of knowledge in various specializations of Computer Science and Engineering and make them industry-ready.
- To engage in collaborative research with academia and industry and generate adequate resources for research activities for seamless transfer of knowledge resulting in sponsored projects and consultancy.
- To cultivate responsibility through sharing of knowledge and innovative computing solutions that benefit the society-at-large.
- To collaborate with academia, industry and community to set high standards in academic excellence and in fulfilling societal responsibilities

PROGRAM OUTCOMES (POs)

PO1. Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2. Problem Analysis: Identify formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3. Design/Development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4. Conduct Investigations of Complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5. Modern Tool Usage: Create select, and, apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6. The Engineer and Society: Apply reasoning informed by contextual knowledge to societal, health, safety. Legal und cultural issues and the consequent responsibilities relevant to professional engineering practice.

PO7. Environment and Sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts and demonstrate the knowledge of, and need for sustainable development.

PO8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9. Individual and Team Work: Function effectively as an individual, and as a member or leader in diverse teams and in multidisciplinary settings.

PO10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11. Project Management and Finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12. Life-Long Learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1: An ability to analyze the common business functions to design and develop appropriate Information Technology for social upliftments.

PSO2: Shall have expertise on the evolving technologies like Python, Machine Learning, Deep learning, IOT, Data Science, Full stack development, Social Networks, Cyber Security, Mobile Apps, CRM, ERP, Big Data, etc.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1: Graduates will have successful careers in computer related engineering fields or will be able to successfully pursue advanced higher education degrees.

PEO2: Graduates will try and provide solutions to challenging problems in their profession by applying computer engineering principles.

PEO3: Graduates will engage in life-long learning and professional development by rapidly adapting to the changing work environment.

PEO4: Graduates will communicate effectively, work collaboratively and exhibit high levels of professionalism and ethical responsibility.

Course Outcomes and CO-PO Mapping

PROJECT OUTCOMES

P1: Understanding the working of Matrix Factorization.

P2: Learning algorithms like K-Means Clustering.

P3: Building an accurate model for Recommendation System.

P4: Hands on working with Tensorflow, Tqdm, PyTorch.

MAPPING PROJECT OUTCOMES WITH PROGRAM OUTCOMES

CO-PO Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
P1	1	2	3	2	1	1			2	2	1	2
P2	2	1	3	3	2	1			2			3
P3	2	2	3		3	2			2	3	1	2
P4	2	3	2	3	3				1		2	

Note: L -1 M – 2 H – 3

The average of Each CO with every PO 's and PSO's should lie between 2 and 3 values.

PROJECT OUTCOMES MAPPING WITH PROGRAM SPECIFIC OUTCOMES

CO-PSO Matrix:

PSO	PSO1	PSO2
P1		3
P2	2	3
P3	3	3
P4		3

PROJECT OUTCOMES MAPPING WITH PROGRAM EDUCATIONAL OBJECTIVES

PEO	PEO1	PEO2	PEO3	PEO4
P1				
P2				
P3				
P4				

DECLARATION

We hereby declare that the results embodied in the dissertation entitled "**Product Recommendation System**" has been carried out by us together during the academic year 203-24 in the partial fulfillment for the award of the Degree in Bachelor of Technology in Computer Science and Engineering affiliated to Jawaharlal Nehru Technological University, Hyderabad. We have not submitted this report to any other university or organization for the award of any other degree.

Student Name	Roll No.
GURRAPU MANOGNYA	20BD1A05CM
KONTE SADHVIKA	20BD1A05CW
NIDUMOLU SRI POORVAJA	20BD1A05DB
YACHAVARAM SRI SAI RITHVIKA REDDY	20BD1A05DV

ACKNOWLEDGMENT

We take this opportunity to thank all the people who have rendered their full support to our project work. We render our thanks to **Dr. Maheshwar Dutta**, B.E., M Tech., Ph.D., Principal who encouraged us to do the Project.

We are grateful to **Mr. Neil Gogte**, Director for facilitating all the amenities required for carrying out this project.

We express our sincere gratitude to **Mr. S. Nitin,** Director and **Professor Dodle Jaya Prakash**, Dean Academics for providing an excellent environment in the college.

We are also thankful to **Mr Para Upendar**, Head of the Department for providing us with both time and amenities to make this project a success within the given schedule.

We are also thankful to our guide **Dr. M Anuradha**, for her valuable guidance and encouragement given to us throughout the project work.

We would like to thank the entire CSE Department faculty, who helped us directly and indirectly in the completion of the project. We sincerely thank our friends and family for their constant motivation during the project work.

GURRAPU MANOGNYA	**20BD1A05CM**
KONTE SADHVIKA	**20BD1A05CW**
NIDUMOLU SRI POORVAJA	**20BD1A05DB**
YACHAVARAM SRI SAI RITHVIKA REDDY	**20BD1A05DV**

ABSTRACT

This project is titled “Product Recommendation System using AI”. Artificial intelligence is the simulation of human intelligence processes by machines, especially computer systems. Specific applications of AI include expert systems, natural language processing, speech recognition and machine vision. It is one of the most booming technologies of today and is making life easier by solving a lot of problems.

The AI-based product recommendation system can provide personalized recommendations to each customer. This means the recommendations are tailored specifically for the individual based on their past behavior and preferences. The recommendations made by the AI-based recommendation system can help customers discover new products they may be interested in. AI-based recommendation systems use a variety of data sources to identify patterns and trends to provide personalized recommendations to each user. Compared to traditional systems, AI recommendations are faster, save time, increase conversation, and propel business growth. Python libraries are the technologies used in this project.

Product recommendation systems using Matrix Factorization have become an integral part of modern platforms, providing personalized suggestions to users based on their preferences, browsing history, and purchase behavior. Matrix factorization has been widely adapted by a lot of companies due to its efficiency and adaptability. This algorithm is being used in this model to build the recommender system which generates latent features when multiplying two different kinds of entities, i.e., users and items.

LIST OF FIGURES

S.No	Name of Screenshot	Page No.
1.5.1	System Architecture	
2.8.1	Movies.csv dataset	
2.8.2	Ratings.csv dataset	
4.2.1.1	Use Case Diagram	
4.2.2.1	Sequence Diagram	
4.2.3.1	Class Diagram	
5.2.1	Code Snippet 1	
5.2.2	Code Snippet 2	
5.2.3	Code Snippet 3	
7.1	Output 1	
7.2	Output 2	
7.3	Output 3	
7.4	Output 4	

Content

Description	Page no
Abstract	**i**
Chapters	
1.Introduction	
1.1 Background and Motivation	7
1.2 Problem Statement	7
1.3 Objectives	7
1.4 Scope and Limitations	8
1.5 Architecture diagram	**9**
2.Software Requirement Specifications	
2.1 What is SRS?	11
2.2 Role of SRS?	11
2.3 Requirement Specification Document	11
2.4 Functional Requirements	12
2.5 Non-Functional Requirements	12
2.6 Hardware Requirements	13
2.7 Software Requirements	13
2.8 Dataset	14
3. LITERATURE SURVEY	**17**
4. SOFTWARE DESIGN	
4.1. Introduction to UML	19
4.2. UML Diagrams	20

5. IMPLEMENTATION

5.1. Algorithm 25

5.2. Code Snippets 26

6. TESTING

6.1. Introduction to Testing 29

6.2. Software Testing Life Cycle 30

7. SCREENSHOTS 32

8. FUTURE ENHANCEMENTS 35

9. CONCLUSION 37

10. REFERENCES 39

Chapter 1

1

[1] KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY | Product Recommendation System

1.Introduction

1.1 Background and Motivation

The AI-based recommendation system can provide personalized recommendations to each customer. This means the recommendations are tailored specifically for the individual based on their past behavior and preferences. The recommendations made by the AI-based recommendation system can help customers discover new products they may be interested in. It can also help them save time by finding products that they are more likely to purchase.

AI-based recommendation systems use a variety of data sources to identify patterns and trends to provide personalized recommendations to each user. This information is then used to generate recommendations for similar products or services that the user might be interested in.

1.2 Problem Statement

A product recommendation is basically a filtering system that seeks to predict and show the items that a user would like to purchase and these systems have become increasingly popular in recent years and if set up and configured properly, it can significantly boost revenues, CTRs, other important metrics.

Matrix factorization is a class of collaborative filtering algorithms used in recommender systems. The algorithm works by decomposing the user-item interaction matrix into the product of two lower dimensionality rectangular matrices. The main aim of this project is to build a model that helps customers get recommendations according to their personal choice and make shopping experience easy which will increase customer satisfaction and also revenues.

1.3 Objectives

The primary goal of an AI-based product recommendation system is to provide users with a tailored shopping experience. By analyzing user behavior, preferences, and historical data, the system aims to suggest products that align with individual tastes and needs. The system seeks to boost customer engagement by offering relevant and timely product suggestions. This can lead to longer browsing sessions, increased page views, and ultimately higher conversion rates. An effective recommendation system identifies opportunities to promote complementary products (cross-selling) or suggest higher-end alternatives (up-selling). This not only increases the average transaction value but also enhances customer satisfaction.

1.4 Scope and Limitations

A product recommendation system using AI has a wide range of applications and can provide significant benefits, but it also comes with its own set of scope and limitations.

Scope:

- AI-powered recommendation systems excel at personalizing user experiences. They can analyze a user's behavior, preferences, and past interactions to suggest products that are most likely to interest them.

- By providing relevant recommendations, businesses can increase the chances of customers making a purchase. This can lead to higher sales and conversion rates.

- Recommender systems can enhance the overall user experience by making it easier for customers to find products they are interested in. This can lead to increased customer satisfaction and loyalty.

- In platforms like streaming services, recommendation systems can help users discover new content that matches their preferences, leading to increased user engagement.

- By promoting products that are more likely to be purchased, businesses can optimize their inventory and reduce overstocking or understocking issues.

- AI-powered recommendation systems can provide valuable insights into customer behavior and preferences, which can be used to tailor marketing campaigns.

Limitations:

- Recommender systems may struggle to provide accurate recommendations for new users who have limited interaction history.

- The effectiveness of a recommendation system heavily relies on the quality and diversity of available data. Biased or incomplete data can lead to inaccurate suggestions.

- Data sparsity is another concern when it comes to user-based CF because new users do not have enough historical data and thus this may result in incorrect recommendations.

- User preferences can change over time, and user-based filtering might not adapt well to these changes. As a result, recommendations might become less accurate as the system doesn't effectively capture these evolving preferences.

- If a product is very niche or newly introduced, there may not be enough data available to make accurate recommendations.

- Some advanced recommendation algorithms can be computationally expensive and may require significant resources to implement and maintain.

2

[2] KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY | Product Recommendation System

1.5 Architecture Diagram

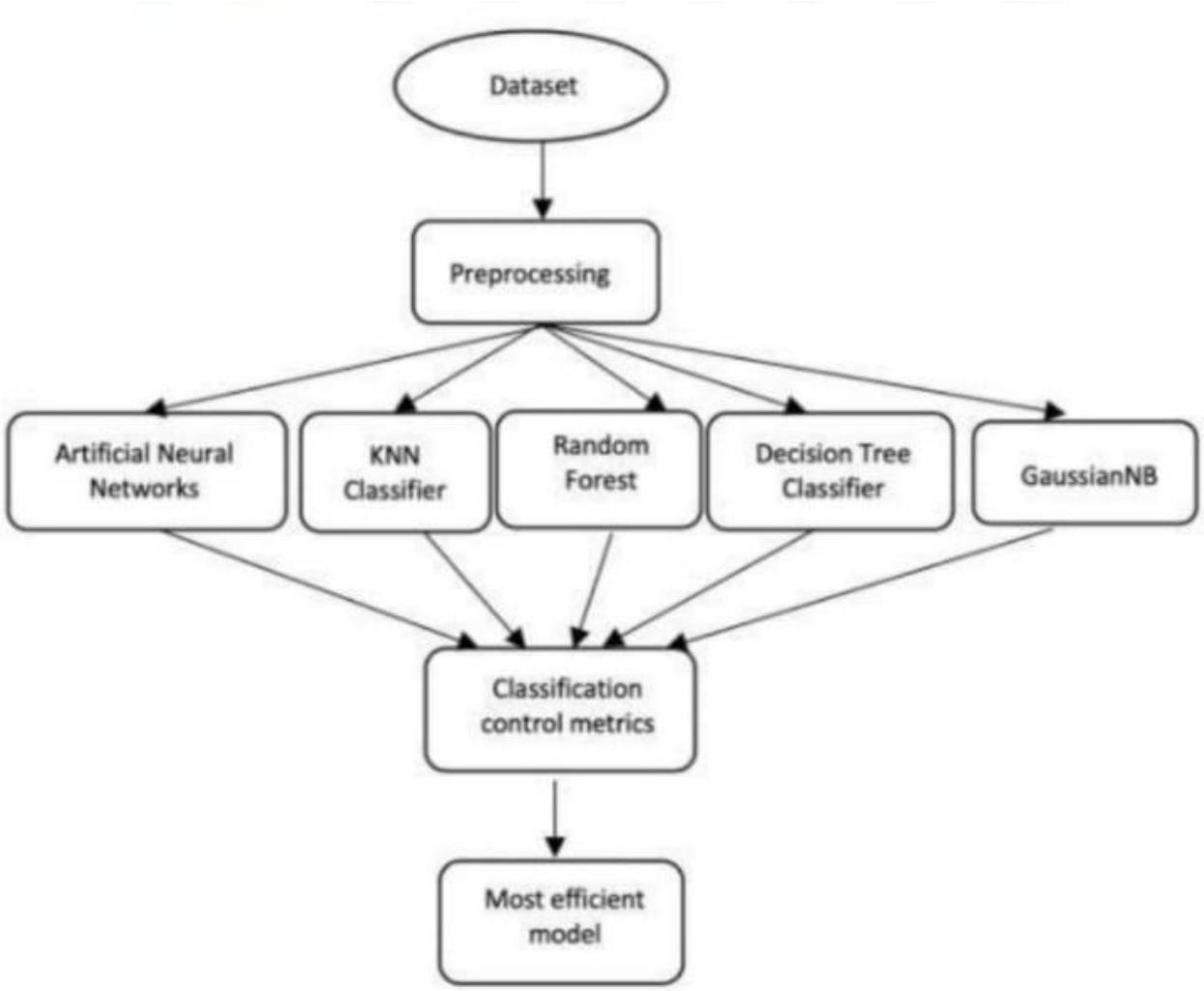

Fig: 1.5.1 - System Architecture

3

[3] KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY | Product Recommendation System

Chapter 2

2.System Requirement Specifications

2.1 What is SRS?

A System Requirements Specification (SRS) is a comprehensive document that outlines the requirements for a software system. It serves as a blueprint for the development team, helping them understand what the software is expected to do, how it should behave, and what constraints or limitations apply.

The SRS phase consists of two basic activities:

Problem/Requirement Analysis: The process is order and more nebulous of the two, deals with understand the problem, the goal and constraints.

Requirement Specification: Here, the focus is on specifying what has been found giving analysis such as representation, specification languages and tools, and checking the specifications are addressed during this activity.

The Requirement phase terminates with the production of the validate SRS document. Producing the SRS document is the basic goal of this phase.

2.2 Role of SRS

An SRS forms the basis of an organization's entire project. It sets out the framework that all the development teams will follow. It provides critical information to all the teams, including development, operations, quality assurance (QA) and maintenance, ensuring the teams are in agreement.

2.3 Requirement specification Document

A Software Requirements Specification (SRS) is a document that describes the nature of a project, software or application. In simple words, SRS document is a manual of a project provided it is prepared before you kick-start a project/application. This document is also known by the names SRS report, software document. A software document is primarily prepared for a project, software or any kind of application.

There are a set of guidelines to be followed while preparing the software requirement specification document. This includes the purpose, scope, functional and non functional requirements, software and hardware requirements of the project. In addition to this, it also contains the information about environmental conditions required, safety and security requirements, software quality attributes of the project etc.

2.4 Functional Requirements

For documenting the functional requirements, the set of functionalities supported by the system are to be specified. A function can be specified by identifying the state at which data is to be input to the system, its input data domain, the output domain, and the type of processing to be carried on the input data to obtain the output data. Functional requirements define specific behaviour or function of the application. Following are the functional requirements:

- All the data must be in the same format as a structured data.
- The data collected will be vectorized and sent across to the classifier.

2.5 Non Functional Requirements

A non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours. Especially these are the constraints the system must work within. Following are the non-functional requirements:

- Product Requirements
- Organizational Requirements
- User Requirements
- Basic Operational Requirements

Product Requirements

Platform Independency: Standalone executables for embedded systems can be created so the algorithm developed using available products could be downloaded on the actual hardware and executed without any dependency to the development and modeling platform.

Correctness: It followed a well-defined set of procedures and rules to compute and also rigorous testing is performed to confirm the correctness of the data.

Ease of Use: Model Coder provides an interface which allows the user to interact in an easy manner.

Modularity: The complete product is broken up into many modules and well-defined interfaces are developed to explore the benefit of flexibility of the product.

Robustness: This software is being developed in such a way that the overall performance is optimized and the user can expect the results within a limited time with utmost relevancy and correctness.

Basic Operational Requirements

The customers are those that perform the eight primary functions of systems engineering, with special emphasis on the operator as the key customer. Operational requirements will define the

basic need and, at a minimum, will be related to these following points:-

Mission profile or scenario: It describes about the procedures used to accomplish mission objective. It also finds out the effectiveness or efficiency of the system.

Performance and related parameters: It points out the critical system parameters to accomplish the mission.

Utilization environments: It gives a brief outline of system usage. Finds out appropriate environments for effective system operation.

Operational life cycle: It defines the system lifetime.

2.6 Hardware Requirements

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatibility and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of hardware requirements. Hardware requirements for present project:

Processor: 2 gigahertz (GHz) or faster processor or SoC.

RAM: 8 gigabyte (GB) for 32-bit or 8 GB for 64-bit.

Hard disk space: 16GB**.**

2.7 Software Requirements

Software Requirements deal with defining software resource requirements and pre-requisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or pre-requisites are generally not included in the software installation package and need to be installed separately before the software is installed. Software requirements for present project:

Operating System: Windows XP/7/8/8.1/10, Linux and Mac Coding

Language: Python

Tools:

1. Pandas

2. Numpy

3. Tensorflow

4. Keras

5. Scikitlearn

Version control: Git for code management

2.8 Data Set

A dataset is a collection of various types of data stored in a digital format. Data is the key component of any Machine Learning project. Datasets primarily consist of images, texts, audio, videos, numerical data points, etc., for solving various Artificial Intelligence challenges such as Image or video classification, Object detection, Face recognition, Emotion classification, Speech analytics, Sentiment analysis, Stock market prediction, etc.

Movie Lens Dataset

- This dataset (ml-latest-small) contains 100836 ratings and 3683 tag applications across 9742 movies. These data were created by 610 users between March 29, 1996 and September 24, 2018. This dataset was generated on September 26, 2018.
- Users were selected at random for inclusion. All selected users had rated at least 20 movies. No demographic information is included. Each user is represented by an id, and no other information is provided.
- The data are contained in the files `links.csv`, `movies.csv`, `ratings.csv` and `tags.csv`.

Movies.csv

- Movie information is contained in the file `movies.csv`. Each line of this file after the header row represents one movie, and has the following format: movieId, title, genres.
- Movie titles are entered manually or imported from <https://www.themoviedb.org/>, and include the year of release in parentheses.

	movieId	title	genres
2	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
3	2	Jumanji (1995)	Adventure|Children|Fantasy
4	3	Grumpier Old Men (1995)	Comedy|Romance
5	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
6	5	Father of the Bride Part II (1995)	Comedy
7	6	Heat (1995)	Action|Crime|Thriller
8	7	Sabrina (1995)	Comedy|Romance
9	8	Tom and Huck (1995)	Adventure|Children
10	9	Sudden Death (1995)	Action
11	10	GoldenEye (1995)	Action|Adventure|Thriller
12	11	American President, The (1995)	Comedy|Drama|Romance
13	12	Dracula: Dead and Loving It (1995)	Comedy|Horror
14	13	Balto (1995)	Adventure|Animation|Children
15	14	Nixon (1995)	Drama
16	15	Cutthroat Island (1995)	Action|Adventure|Romance

Fig: 2.8.1 - movie.csv dataset

Ratings.csv

- All ratings are contained in the file `ratings.csv`. Each line of this file after the header row represents one rating of one movie by one user, and has the following format: userId,movieId,rating,timestamp.
- The lines within this file are ordered first by userId, then, within user, by movieId. Ratings are made on a 5-star scale, with half-star increments (0.5 stars - 5.0 stars).

1	userId	movieId	rating	timestamp
2	1	1	4	9.65E+08
3	1	3	4	9.65E+08
4	1	6	4	9.65E+08
5	1	47	5	9.65E+08
6	1	50	5	9.65E+08
7	1	70	3	9.65E+08
8	1	101	5	9.65E+08
9	1	110	4	9.65E+08
10	1	151	5	9.65E+08
11	1	157	5	9.65E+08
12	1	163	5	9.65E+08
13	1	216	5	9.65E+08
14	1	223	3	9.65E+08
15	1	231	5	9.65E+08

Fig: 2.8.2 - Rating.csv dataset

Chapter 3

3. Literature Survey

1. Product Recommendation System using Data Mining, By Ruey-Shun Chen, Yung-Shun Tsai

In this project, Product Recommender is developed using Data Mining techniques. The Bayesian network concept is used to build up a personalized product recommender system in order to generate different recommendations, ranking from high to low, to help reader to locate information most suitable to his requirement.

2. Product Recommendation System using AI, By Qian Zhang

This project uses NLP, Reinforcement Learning and Deep Learning in order to develop the Recommendation System. In this project, eight fields of AI are reviewed to introduce their applications in recommender systems, discuss the open research issues, and give directions of possible future research on how AI techniques will be applied in recommender systems.

3. Image Based Product Recommendation System, By Zaki Mustafa

The project deals with recommending the most suitable products to users based on the current selection and choice of the product. This model relies on the fact that multiple features can be extracted from images and used for similarity computation.

4. Building a Product Recommendation System with your Sales Data, By Eric Yang

This system was built to power e-commerce products to product recommendations. For example when a customer clicks on a product, most sites will show a product detail page (PDP) and commonly you might see more products shared on that page under headings such as '*You Might Also Like'* or *'Similar Products'*.

5. Amazon Recommender System, By JH (Janghyun) Baek, John Tsai, Justin Shamoun, Muriel Marable, Ying Cui

35% of Amazon web sales were generated through their recommended items [source: McKinsey]. This study aims to construct an apparel recommender system for Amazon users through user-rating history, product images and product title text. Multiple deep learning models were built on both readily-available and engineered datasets resulting in a multi-step recommender system.

Chapter 4

4. System Design

4.1. Introduction to UML

UML Diagrams are the output of the Unified Modelling Language. It is a pictorial representation of classes, objects, and relationships between them. UML diagram is a model that describes a part of a system. It is used to define the functionality or a design of a system. UML is a way to visually represent the architecture, design, and implementation of complex software systems.

UML diagrams are divided into three different categories such as,

1. Structural diagram

2. Behavioral diagram

3. Interaction diagram

4.1.1. Structural diagrams in UML

Structural diagrams are used to represent a static view of a system. It represents a part of a system that makes up the structure of a system. A structural diagram shows various objects within the system.

Following are the various structural diagrams in UML:

1. Class diagram

2. Object diagram

3. Package diagram

4. Component diagram

5. Deployment diagram

4.1.2. Behavioral diagrams in UML

Any real-world system can be represented in either a static form or a dynamic form. A system is said to be complete if it is expressed in both the static and dynamic ways. The behavioral diagram represents the functioning of a system.

UML diagrams that deal with the static part of a system are called structural diagrams. UML diagrams that deal with the moving or dynamic parts of the system are called behavioral diagrams.

Following are the various behavioral diagrams in UML:

1. Activity diagram

2. Use case diagram

3. State machine diagram

4.1.3. Interaction diagrams in UML

Interaction diagram is nothing but a subset of behavioral diagrams. It is used to visualize the flow between various use case elements of a system. Interaction diagrams are used to show an interaction between two entities and how data flows within them.

Following are the various interaction diagrams in UML:

1. Timing diagram

2. Sequence diagram

3. Collaboration diagram

4.2. UML Diagrams

4.2.1. Use-case diagram

A use case illustrates a unit of functionality provided by the system. The main purpose of the usecase diagram is to help development teams visualize the functional requirements of a system, including the relationship of "actors" (human beings who will interact with the system) to essential processes, as well as the relationships among different use cases.

A use-case diagram is typically used to communicate the high-level functions of the system and the system's scope. Use-case diagrams generally show groups of use cases — either all use cases for the complete system, or a breakout of a particular group of use cases with related functionality (e.g., all security administration-related use cases).

- To show a use case on a use-case diagram: We draw an oval in the middle of the diagram and put the name of the use case in the center of, or below, the oval.
- To draw an actor (indicating a system user) on a use-case diagram: We draw a stick person to the left or right of your diagram.
- Use simple lines to depict relationships between actors and use cases

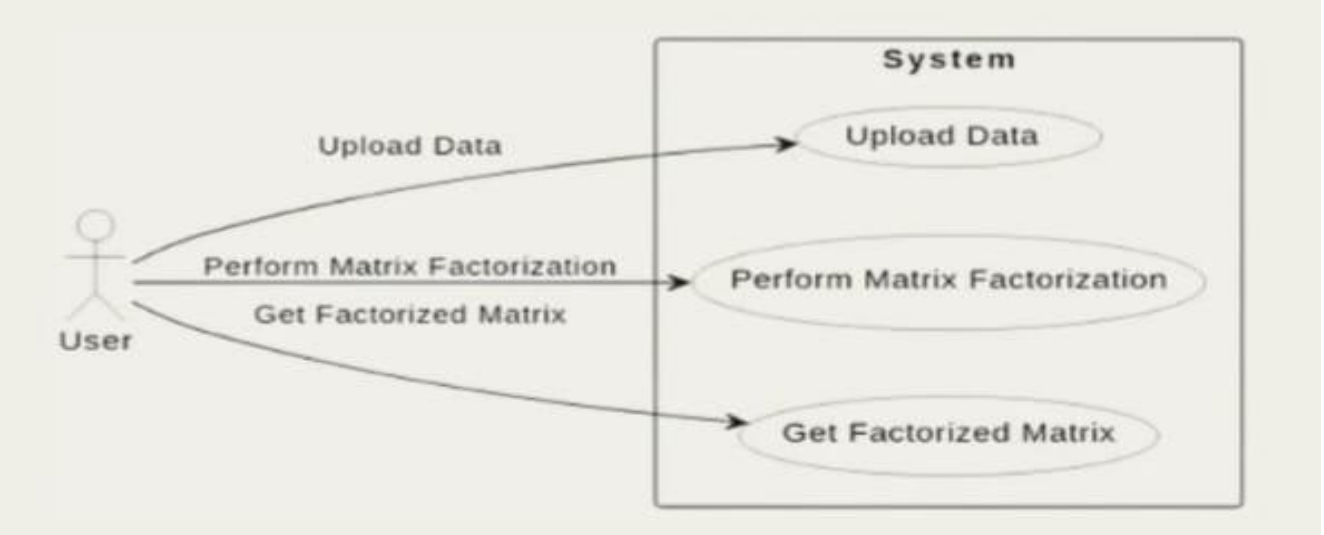

Fig: 4.2.1.1 - Use Case Diagram

4.2.2. Sequence diagram

A Sequence diagram is an interaction diagram that shows how processes operate with one another and what is their order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios. Sequence diagram are an easy and intuitive way of describing the behavior of a system by viewing the interaction between the system and the environment. A sequence diagram shows an interaction arranged in a time sequence. A sequence diagram has two dimensions: vertical dimension represents time, the horizontal dimension represents the objects existence during the interaction.

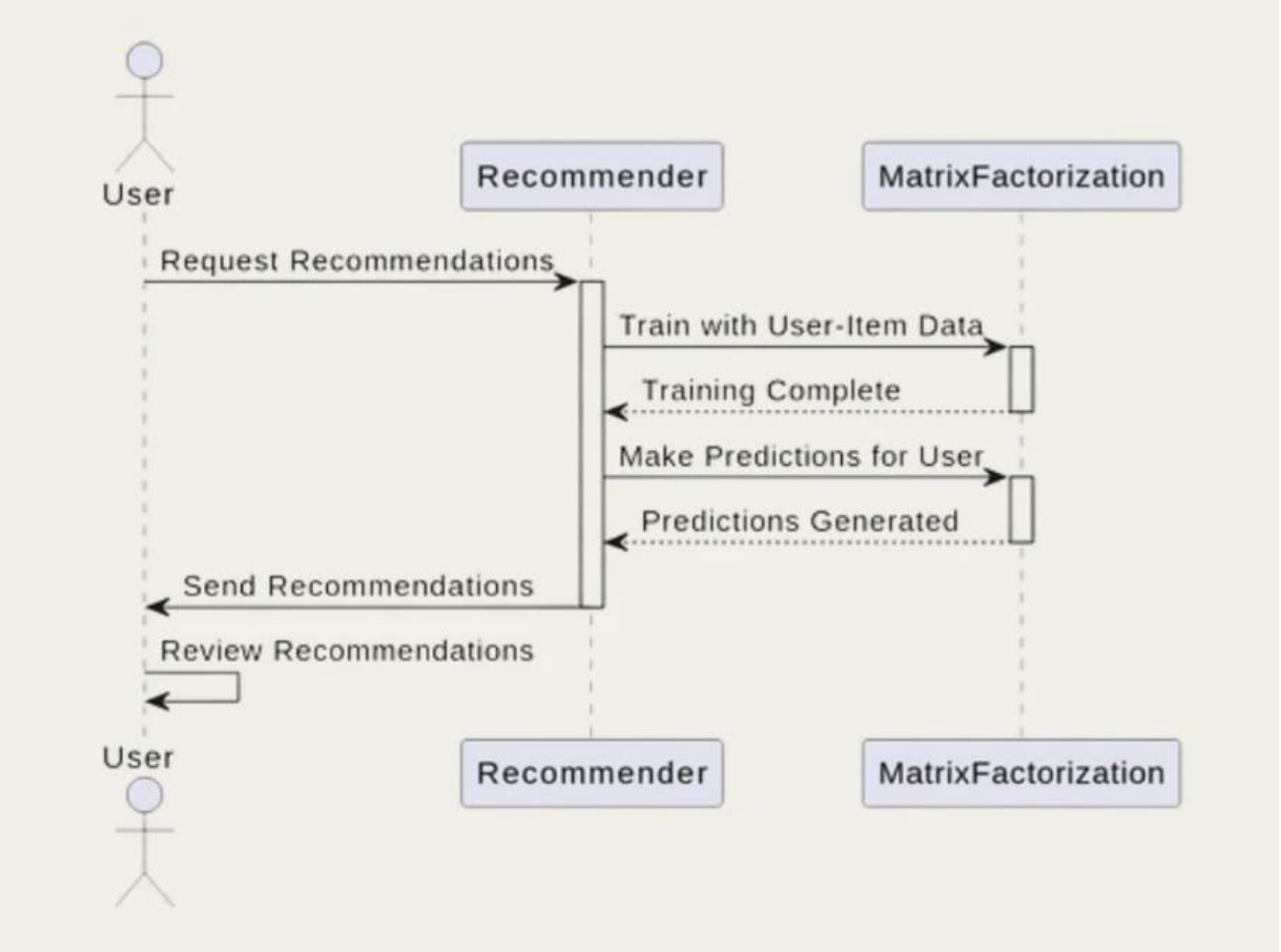

Fig: 4.2.2.1 - Sequence Diagram

4.2.3. Class diagram

The class diagram shows how the different entities (people, things, and data) relate to each other; in other words, it shows the static structures of the system. A class diagram can be used to display logical classes, which are typically the kinds of things the business people in an organization talk about. Class diagrams can also be used to show implementation classes, which are the things that programmers typically deal with. An implementation class diagram will probably show some of the same classes as the logical classes diagram. The implementation class diagram won't be drawn with the same attributes, however, because it will most likely have references to things like Vectors and HashMaps. Class diagrams show the static structure of a system, including classes, their attributes and behaviors, and the relationships between each class. A class is depicted on the class diagram as a rectangle with three horizontal sections. The upper section shows the class's name; the middle section contains the class's attributes; and the lower section contains the class's operations (or "methods").

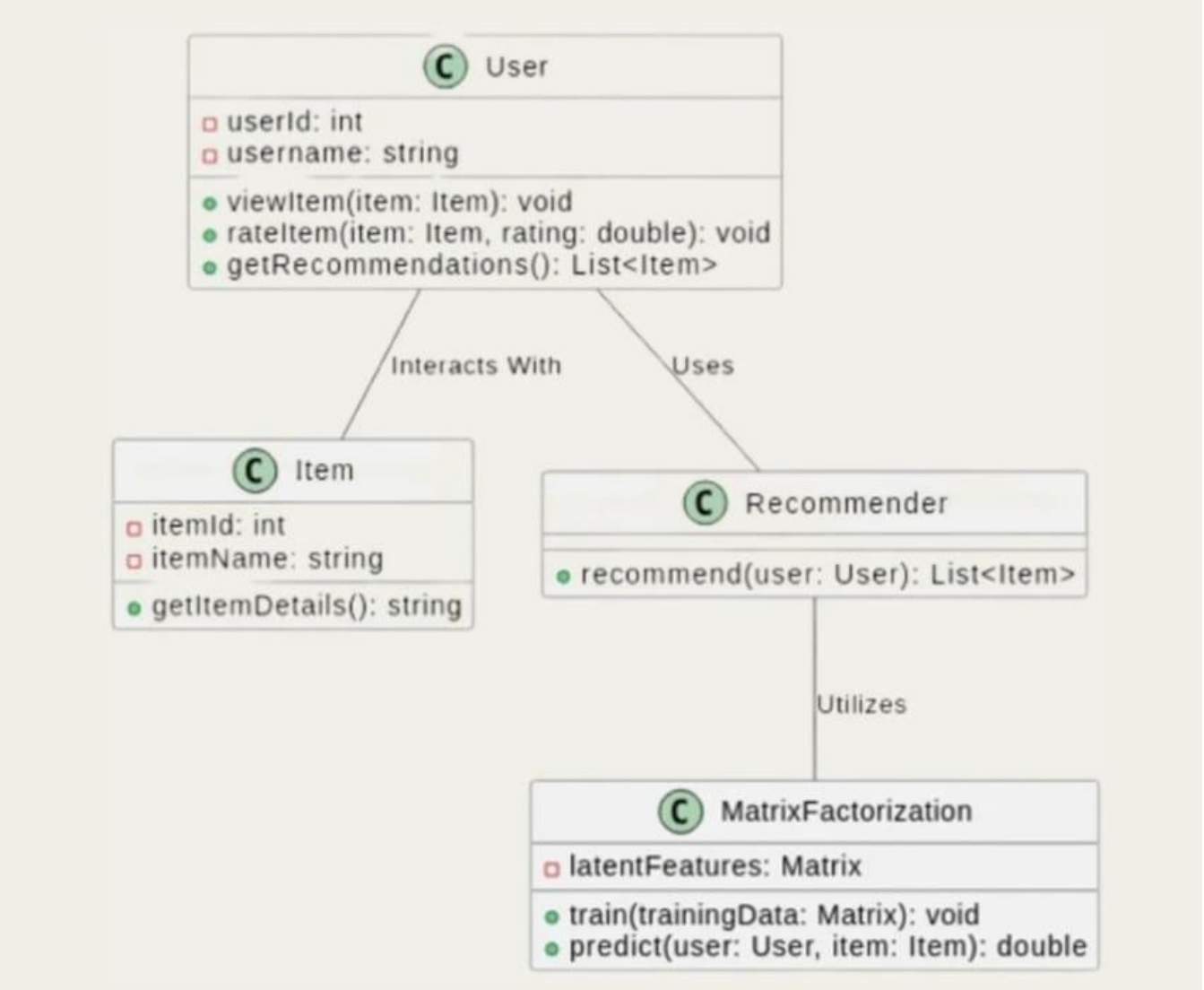

Fig: 4.2.3.1 - Class Diagram

Chapter 5

5. Implementation

Python: Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is widely used to develop Artificial Intelligence projects and systems.

Some libraries used in this recommendation system are:

Torch: To provide flexible, n-dimensional arrays or tensors.

Numpy: To perform a wide variety of mathematical operations on arrays.

5.1 Algorithm

Login to google colab at https://colab.research.google.com.

Step 1: Create a new notebook.

Step 2: Import the necessary packages to your python environment.

Step 3: Import the dataset.

Step 4: Map the text attributes in the dataset to 0s and 1s.

Step 5: Replace the null(NaN) values with the median of the respective attribute.

Step 6: Scale each value within the dataset from 0 to 1.

Step 7: Extract important features.

Step 8: Split the dataset into training and testing datasets.

Step 9: Build and Compile the model.

Step 10: Train the model.

Step 11: Evaluate various model metrics.

Step 12: Save the model.

Step 13: Use Tensorflow Js module convert the model.h5 into equivalent model.json.

Step 14: Integrate the model with the Web Application inorder to make it available for the users.

5.2 Code snippets

1) **Matrix Factorization:** Matrix factorization is a mathematical technique which involves decomposing a matrix into the product of multiple matrices, typically with the goal of simplifying the representation of data or extracting meaningful patterns and features from the original data. Prediction is made by calculating the dot product of the two vectors corresponding to the values. Here it takes the values from the dataset and combines them to form a full rating matrix by getting a dot product matrix of both movies and ratings file.

Fig: 5.2.1 - Code Snippet 1

2) Dropping and enumerating data as a part of data pre-processing. Enumeration in data preprocessing refers to the process of converting categorical or discrete data into a numerical format that can be used in machine learning algorithms.This process is done so that the data is consistently formatted and follows a standardized structure, making it easier to work with and analyze.

```
class Loader(Dataset):
    def __init__(self):
        self.ratings = ratings_df.copy()

        # Extract all user IDs and movie IDs
        users = ratings_df.userId.unique()
        movies = ratings_df.movieId.unique()

        #--- Producing new continuous IDs for users and movies ---

        # Unique values : index
        self.userid2idx = {o:i for i,o in enumerate(users)}
        self.movieid2idx = {o:i for i,o in enumerate(movies)}

        # Obtained continuous ID for users and movies
        self.idx2userid = {i:o for o,i in self.userid2idx.items()}
        self.idx2movieid = {i:o for o,i in self.movieid2idx.items()}

        # return the id from the indexed values as noted in the lambda function down below.
        self.ratings.movieId = ratings_df.movieId.apply(lambda x: self.movieid2idx[x])
        self.ratings.userId = ratings_df.userId.apply(lambda x: self.userid2idx[x])


        self.x = self.ratings.drop(['rating', 'timestamp'], axis=1).values
        self.y = self.ratings['rating'].values
        self.x, self.y = torch.tensor(self.x), torch.tensor(self.y) # Transforms the data to tensors (ready for torch models.)

    def __getitem__(self, index):
        return (self.x[index], self.y[index])

    def __len__(self):
        return len(self.ratings)
```

Fig: 5.2.2 - Code Snippet 2

3) K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. The goal of clustering is to divide the population or set of data points into a number of groups so that the data points within each group are more comparable to one another and different from the data points within the other groups. It is essentially a grouping of things based on how similar and different they are to one another. It is a user-defined parameter representing the number of clusters you want to identify.

Working of K-Means Algorithm :

a) Initialization : You start by selecting K initial cluster centroids, which are points in the feature

space. These centroids can be randomly chosen from the data or using other strategies, like

k-means++ for better initialization.

b) Assignment: For each data point, you calculate its distance to all K centroids and assign the point to the nearest centroid. This results in the formation of K clusters.

c) Update: After all data points have been assigned to clusters, you recalculate the centroid for each cluster by taking the mean of all the data points in that cluster. The new centroids represent the center of the cluster.

d) Iteration: Steps 2 and 3 are repeated iteratively until a stopping criterion is met. Typically, this criterion is based on the convergence of centroids or a maximum number of iterations.

The final output of the K-means algorithm is a set of K cluster centroids, each representing the center of a cluster, and each data point is assigned to one of the K clusters. The algorithm aims to minimize the within-cluster sum of squares, which means that data points within a cluster should be as close to their cluster's centroid as possible.

```
[ ] trained_movie_embeddings = model.item_factors.weight.data.cpu().numpy()
    len(trained_movie_embeddings) # unique movie factor weights

    9724
```

```
from sklearn.cluster import KMeans
# Fit the clusters based on the movie weights
kmeans = KMeans(n_clusters=10, random_state=0).fit(trained_movie_embeddings)
for cluster in range(10):
  print("Cluster #{}".format(cluster))
  movs = []
  for movidx in np.where(kmeans.labels_ == cluster)[0]:
    movid = train_set.idx2movieid[movidx]
    rat_count = ratings_df.loc[ratings_df['movieId']==movid].count()[0]
    movs.append((movie_names[movid], rat_count))
  for mov in sorted(movs, key=lambda tup: tup[1], reverse=True)[:10]:
    print("\t", mov[0])
```

Fig: 5.2.3 - Code Snippet 3

4) Extracting clusters from the trained model using K-Means Clustering. The model gives out

clusters that are recommended to the users based on similar data points obtained after calculating a full matrix.

Chapter 6

6. Testing

6.1 Introduction to Testing

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. Testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements. According to ANSI/IEEE 1059 standard, Testing can be defined as - A process of analyzing a software item to detect the differences between existing and required conditions (that is defects/errors/bugs) and to evaluate the features of the software item.

Who does Testing?

It depends on the process and the associated stakeholders of the project(s). In the IT industry, large companies have a team with responsibilities to evaluate the developed software in context of the given requirements. Moreover, developers also conduct testing which is called Unit Testing.

In most cases, the following professionals are involved in testing a system within their respective capacities:

- Software Tester
- Software Developer
- Project Lead/Manager
- End User

Levels of testing include different methodologies that can be used while conducting software testing. The main levels of software testing are:

- Functional Testing
- Non-functional Testing

Functional Testing

This is a type of black-box testing that is based on the specifications of the software that is to be tested. The application is tested by providing input and then the results are examined that need to conform to the functionality it was intended for. Functional testing of a software is conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.

6.2 Software Testing Life Cycle

The process of testing a software in a well planned and systematic way is known as software testing lifecycle (STLC). Different organizations have different phases in STLC however generic Software Test Life Cycle (STLC) for waterfall development model consists of the following phases.

1. Requirements Analysis
2. Test Planning
3. Test Analysis
4. Test Design

● **Requirements Analysis**

In this phase testers analyze the customer requirements and work with developers during the design phase to see which requirements are testable and how they are going to test those requirements. It is very important to start testing activities from the requirements phase itself because the cost of fixing defect is very less if it is found in requirements phase rather than in future phases.

● **Test Planning**

In this phase all the planning about testing is done like what needs to be tested, how the testing will be done, test strategy to be followed, what will be the test environment, what test methodologies will be followed, hardware and software availability, resources, risks etc. A high level test plan document is created which includes all the planning inputs mentioned above and circulated to the stakeholders.

● **Test Analysis**

After test planning phase is over test analysis phase starts, in this phase we need to dig deeper into project and figure out what testing needs to be carried out in each SDLC phase. Automation activities are also decided in this phase, if automation needs to be done for software product, how will the automation be done, how much time will it take to automate and which features need to be automated. Nonfunctional testing areas (Stress and performance testing) are also analyzed and defined in this phase.

● **Test Design**

In this phase various black-box and white-box test design techniques are used to design the test cases for testing, testers start writing test cases by following those design techniques, if automation testing needs to be done then automation scripts also need to be written in this phase.

Chapter 7

7. Screenshots

Extracted clusters from the module are:

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` e
  warnings.warn(
Cluster #0
        Jurassic Park (1993)
        Terminator 2: Judgment Day (1991)
        Toy Story (1995)
        Independence Day (a.k.a. ID4) (1996)
        Apollo 13 (1995)
        Fugitive, The (1993)
        Batman (1989)
        Aladdin (1992)
        Sixth Sense, The (1999)
        True Lies (1994)
Cluster #1
        Alien (1979)
        Kill Bill: Vol. 1 (2003)
        Kill Bill: Vol. 2 (2004)
        Shining, The (1980)
        Interview with the Vampire: The Vampire Chronicles (1994)
        Taxi Driver (1976)
        Dr. Strangelove or: How I Learned to Stop Worrying and Love the Bomb (1964)
        Nightmare Before Christmas, The (1993)
        Natural Born Killers (1994)
        Mars Attacks! (1996)
```

Fig: 7.1 - Output 1

```
Cluster #2
        Shrek (2001)
        Star Wars: Episode I - The Phantom Menace (1999)
        Titanic (1997)
        Harry Potter and the Sorcerer's Stone (a.k.a. Harry Potter and the Philosopher's Stone) (2001)
        Sleepless in Seattle (1993)
        Harry Potter and the Chamber of Secrets (2002)
        Harry Potter and the Prisoner of Azkaban (2004)
        Shrek 2 (2004)
        300 (2007)
        Game, The (1997)
Cluster #3
        Speed (1994)
        Mrs. Doubtfire (1993)
        Twister (1996)
        Home Alone (1990)
        American Pie (1999)
        Wizard of Oz, The (1939)
        Back to the Future Part III (1990)
        Back to the Future Part II (1989)
        Broken Arrow (1996)
        Maverick (1994)
```

Fig: 7.2 - Output 2

```
Cluster #4
          Forrest Gump (1994)
          Shawshank Redemption, The (1994)
          Silence of the Lambs, The (1991)
          Matrix, The (1999)
          Star Wars: Episode IV - A New Hope (1977)
          Schindler's List (1993)
          Star Wars: Episode V - The Empire Strikes Back (1980)
          Usual Suspects, The (1995)
          Raiders of the Lost Ark (Indiana Jones and the Raiders of the Lost Ark) (1981)
          Lord of the Rings: The Fellowship of the Ring, The (2001)
Cluster #5
          Ace Ventura: Pet Detective (1994)
          Pirates of the Caribbean: The Curse of the Black Pearl (2003)
          Dumb & Dumber (Dumb and Dumber) (1994)
          Fifth Element, The (1997)
          Happy Gilmore (1996)
          Avatar (2009)
          Matrix Reloaded, The (2003)
          Iron Man (2008)
          Star Trek: First Contact (1996)
          Monty Python's Life of Brian (1979)
Cluster #6
          Pulp Fiction (1994)
          Fight Club (1999)
          American Beauty (1999)
          Seven (a.k.a. Se7en) (1995)
          Godfather, The (1972)
          Fargo (1996)
          Twelve Monkeys (a.k.a. 12 Monkeys) (1995)
          Memento (2000)
          One Flew Over the Cuckoo's Nest (1975)
          Reservoir Dogs (1992)
```

Fig: 7.3 - Output 3

```
Cluster #7
          Braveheart (1995)
          Gladiator (2000)
          Mission: Impossible (1996)
          Inception (2010)
          Good Will Hunting (1997)
          Batman Forever (1995)
          Pretty Woman (1990)
          GoldenEye (1995)
          Clear and Present Danger (1994)
          Crimson Tide (1995)
Cluster #8
          Mask, The (1994)
          Die Hard: With a Vengeance (1995)
          Stargate (1994)
          Waterworld (1995)
          Net, The (1995)
          Outbreak (1995)
          Cliffhanger (1993)
          Armageddon (1998)
          Mummy, The (1999)
          Santa Clause, The (1994)
Cluster #9
          Godzilla (1998)
          Joe Dirt (2001)
          Fantastic Four: Rise of the Silver Surfer (2007)
          Honey, I Blew Up the Kid (1992)
          Speed 2: Cruise Control (1997)
          Battlefield Earth (2000)
          Rocky V (1990)
          Karate Kid, Part III, The (1989)
          Dungeons & Dragons (2000)
          Rambo III (1988)
```

Fig: 7.4 - Output 4

Chapter 8

8. Future Scope

A product recommendation system using AI has a promising future, as it addresses a critical aspect of modern commerce - personalization. Here are some key areas where the future of product recommendation systems using AI looks bright:

- AI-driven recommendation systems will become increasingly personalized, taking into account a wider range of user data such as browsing history, location, device type, and even contextual data like weather or time of day.
- With advancements in natural language processing (NLP) and computer vision, recommendation systems will be able to process and understand more complex inputs. This could include voice commands, images, or even videos.
- AR and VR technologies will provide a new dimension for product recommendations. Users might be able to virtually try on clothes or visualize furniture in their own homes before making a purchase.
- As AI models become more sophisticated, there will be a growing need for transparency and explainability. Users will want to understand why a particular recommendation is being made.
- Recommendation systems will be integrated into larger ecosystems. For example, they might work in conjunction with smart home devices, digital assistants, and other IoT technologies to provide a seamless user experience.
- As new types of products and services emerge, recommendation systems will adapt to accommodate these changes. This could include things like recommending digital services, experiences, or subscription-based products.

Chapter 9

9. Conclusion

In our daily life the use of internet is increasing and everything is possible through internet money sharing, online shopping and much more. The focus of this paper is on product recommendation. For recommendation of product different algorithm and machine learning techniques were used. Through these techniques the algorithm is used to predict or similar items according to user's likeness based on his information. Even sometimes if the recommended product isn't due to the likeness of the user the whole recommendation system can be considered as a spam, because each and every person has specific likes or dislikes so a model is designed keeping in view to avoid such kind of things in future. No new techniques were proposed in this article, this paper is basically a review of the previous machine learning techniques which were used for product recommendation. Product recommendations are important because they improve metrics like conversion rates, revenue per visitor, and average order value (AOV) while also improving the customer experience. In conclusion, there are lot of technical explanations that can be made on the types of product recommendation engines. All that the users or buyers mostly care are the products and the quality of recommendations that the engine will give. Such cognitive computing methods can take the quality of your recommenders to the next level.

Chapter 10

10. References

- https://www.youtube.com/watch?v=G4MBc40rQ2k
- https://towardsdatascience.com/what-are-product-recommendation-engines-and-the-various-versions-of-them-9dcab4ee26d5
- http://www.quuxlabs.com/blog/2010/09/matrix-factorization-a-simple-tutorial-and-implementation-in-python/
- https://dl.acm.org/doi/10.1145/1454008.1454049

[1] Dutta, R. & Mukhopadhyay, D., 2008. Offering a Recommendation System in E-commerce, IEEE.

[2] Driskill, R. & Riedl, J., 1999. Recommender Systems for E-Commerce: Challenges and Opportunities. American Association for Artificial Intelligence, pp.73–76. Available at: http://aaaipress.org/Papers/Workshops/1999/WS-99-01/WS99-01-012.pdf.

[3] Aciar, S. et al., 2007. Recommender System Based on Consumer Product Reviews.pdf.

[4] H.Li, F.Cai and Z.Liao, "Content Based-Filtering Recommendation Algorithm Using", in Proceedings of the HMM, pp. 275-277, 2012 View at: Google Scholar.

[5] Dr. Senthil Kumar Thangavel, Neetha Susan Thampi "Performance Analysis of Various Recommendation Algorithms Using Apache Hadoop Mahout", International Journal of Scientific & Engineering Research Vol 4, 2013. Available at: academia.edu.

[6] Swati Pandey, T.Senthil Kumar "Costomization of recommendation System using Collaborative Filtering Algorithm on Cloud using Mahout", International Journal of Research in Engineering and Technology Available at: academia.edu.