A

Minor Project Report
On

AI-POWERED RECOMMENDATION SYSTEM: BUILD A RECOMMENDATION ENGINE USING BIG DATA AND PREDICTIVE ANALYTICS

Submitted in partial fulfillment of the requirements for the award of Degree

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

by

T.SRINIDHI (228R1A6752)

Under the Guidance of

Mrs. A. Shravani Reddy

Assistant Professor Department of CSE-DATA SCIENCE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

CMR ENGINEERING COLLEGE

UGC AUTONOMOUS

(Approved by AICTE, NEW DELHI, Affiliated to JNTU, Hyderabad)

(Kandlakoya, Medchal Road, R.R. Dist. Hyderabad-501 401)

2024-2025

CMR ENGINEERING COLLEGE

(UGC Autonomous)

(Accredited by NBA, Approved by AICTE NEW DELHI, Affiliated to JNTU, Hyderabad)

Kandlakoya, Medchal Road, Hyderabad-501 401)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

CERTIFICATE

This is to certify that the project entitled **"AI-POWERED RECOMMENDATION SYSTEM:BUILD A RECOMMENDATIONENGINE USING BIG DATA AND PREDICTIVE ANALYTICS"** is a bonafide work carried out by

T.SRINIDHI (228R1A6752)

in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering (Data Science) to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by them under our guidance and supervision during the year 2024-25.

Internal Guide	Mini Project Coordinator	Head of the Department
Mrs.A. Shravani Reddy	**Mrs.M.Chandana**	**Dr. M. Laxmaiah**
Assistant Professor	Assistant Professor	Professor & H.O. D
CSE(DS), CMREC	CSE(DS), CMREC	CSE(DS), CMREC

DECLARATION

This is to certify that the work reported in the present project entitled "**AI-Powered Recommendation System: Build a recommendation engine using big data and predictive analytics"** is a record of bona fide work done by us in the Department of Computer Science and Engineering, CMR Engineering College, JNTU Hyderabad. The reports are based on the project work done entirely by us and not copied from any other source. We submit our project for further development by any interested students who share similar interests to improve the project in the future.

The results embodied in this project report have not been submitted to any other University or Institute for the award of any degree or diploma to the best of our knowledge and belief.

T.SRINIDHI (228R1A6752)

ACKNOWLEDGMENT

We are extremely grateful to **Dr.A. Srinivasula Reddy**, Principal and **Dr.M. Laxmaiah**, **HOD**, **Department of CSE-DATA SCIENCE, CMR Engineering College** for their constant support**.**

We are extremely thankful to **Mrs. A. Shravani Reddy,** Assistant Professor, Internal Guide, Department of **CSE-DATA SCIENCE**, for her constant guidance, encouragement and moral support throughout the project.

We will be failing in duty if I do not acknowledge with grateful thanks to the authors of the references and other literatures referred in this Project.

We thank **Mrs.M.Chandana** Assistant Professor, Mini Project Coordinator for her constant support in carrying out the project activities and reviews.

We express my thanks to all staff members and friends for all the help and co-ordination extended in bringing out this project successfully in time.

T.SRINIDHI (228R1A6752)

CONTENTS

ABSTRACT

This project focuses on developing an AI-powered recommendation system using big dataand predictive analytics to deliver personalized recommendations based on user behavior, preferences, and historical interactions. The system combines Collaborative Filtering, Content-Based Filtering, and Hybrid Approaches to provide accurate and dynamic suggestions.

Collaborative filtering analyzes user-item interactions and similarities to recommend products or services that similar users have enjoyed. Content-based filtering suggests items based on their features and a user's past preferences. The hybrid approach blends both methods for enhanced accuracy.

The system will be designed for scalability and efficiency, utilizing big data tools like Apache Hadoop, Spark, and PySpark. Predictive analytics algorithms, including machine learning models, will improve the relevance of recommendations.

This recommendation engine is suitable for a wide range of applications, such as:

●E-commerce platforms, to drive sales through personalized product suggestions.

●Streaming services, for recommending content based on user preferences.

●Online learning platforms, offering personalized course recommendations.

LIST OF FIGURES

1. INTRODUCTION

1.1 Introduction & Objectives:

In the digital age, where vast amounts of content and products are available at the fingertips of users, filtering and presenting relevant information is a significant challenge. Recommender systems have emerged as one of the most effective solutions to this challenge, playing a vital role in delivering personalized experiences to users. Whether in e-commerce, streaming platforms, or content-based websites, recommender systems help users discover items that align with their preferences, leading to higher engagement, satisfaction, and retention.

Recommender Systems:

A recommender system is a type of information filtering system that seeks to predict the most relevant items for users from a large pool of available options. These systems have evolved from simple rule-based approaches to more sophisticated machine learning and AI-driven models that can analyze large volumes of data, such as user behavior, preferences, past interactions, and even demographic information.

Importance of Recommender Systems:

As businesses and services transition into the digital realm, one of the key challenges faced by platforms is managing the overwhelming volume of content. Without a proper filtering mechanism, users may be overwhelmed or struggle to find items that suit their interests. Recommender systems solve this problem by offering highly personalized suggestions, which improves the user experience by making it easier for users to find what they are looking for, thus increasing the likelihood of interaction and conversion. These systems are widely used across several industries, including:

●**E-commerce platforms**: Suggesting products based on browsing history or previous purchases.

●**Streaming services**: Recommending movies, TV shows, or music based on user preferences and behavior.

●**Social media platforms**: Suggesting friends, pages, or posts based on the user's interactions.

●**News and content platforms**: Recommending articles, blogs, or news pieces based on user preferences.

1.2 Objective of the Project:

The primary objective of this project is to design and develop an AI-powered recommender system that uses advanced machine learning models to predict and recommend items such as products, movies, articles, or services based on user data. The system will leverage historical data, user preferences, and behavioral insights to deliver accurate and relevant suggestions. By utilizing machine learning algorithms and big data techniques, the system will continuously adapt to evolving user preferences and provide personalized recommendations that improve over time.

The specific objectives of this project include:

1. **Personalization:** To create a personalized user experience by delivering item recommendations based on the individual user's preferences, behavior, and historical interactions.

2. **Model Integration:** To combine multiple recommendation techniques (such as collaborative filtering, content-based filtering, and hybrid methods) to provide accurate and diverse recommendations.

3. **Scalability:** To design a system that can handle large amounts of data, ensuring that the recommendations remain relevant and timely as the user base grows.

4. **Real-Time Recommendations:** To build a recommendation engine capable of updating suggestions in real-time, using new user data and interactions.

5. **Efficiency and Accuracy:** To ensure that the recommendation algorithms are optimized for efficiency and high accuracy, making the system not only fast but also capable of predicting the most relevant suggestions for users.

6. **Evaluation:** To establish a system for evaluating the effectiveness of the recommendations using standard metrics (such as precision, recall, and F1 score), ensuring that the system provides meaningful suggestions that enhance the user experience. Through the implementation of predictive analytics, machine learning, and big data processing techniques, this project aims to develop a robust, scalable, and accurate recommendation system that can be applied across various industries. The system will provide a seamless and personalized user experience, enhancing engagement and satisfaction while improving business outcomes such as conversion rates and customer retention.

The overall goal of this project is not just to design a recommendation engine but to create a framework that can be easily adapted to a wide range of domains, from retail and entertainment to education and social platforms. By focusing on the needs and preferences of users, the system will ensure that every recommendation feels intuitive and relevant, thus driving user engagement and retention.

1.3 Purpose of the project:

To enhance user satisfaction by offering personalized content recommendations based on data-driven insights. By analyzing user behavior, preferences, and historical interactions, the system tailors suggestions that align with individual needs. It combines collaborative filtering, content-based filtering, and hybrid approaches to deliver highly accurate and relevant recommendations. The system leverages advanced predictive analytics and big data processing to ensure scalability and efficiency, making it suitable for various platforms such as e-commerce, streaming services, and online learning. Ultimately, this AI-powered recommendation engine improves user experience by continuously adapting to evolving preferences, fostering deeper engagement.

1.4 Existing System with Disadvantages:

Existing recommendation systems primarily rely on either collaborative filtering or content-based filtering, each having its own limitations. Collaborative filtering struggles with cold start problems, where new users or items lack sufficient interaction data, making it difficult to generate recommendations. Content-based filtering, while useful for new items,often leads to over specialization, offering repetitive suggestions that lack diversity. Furthermore, both approaches fail to adapt quickly to new data, resulting in outdated or irrelevant recommendations over time. Collaborative systems also suffer from scalability issues when handling large datasets, as they require significant computational resources for processing user-item interactions. These shortcomings hinder the effectiveness of current systems in providing accurate and personalized experiences across dynamic and growing platforms.

Disadvantages:

Current recommendation systems have key limitations:

1. Cold Start: Struggle with new users or items due to lack of data.

2. Scalability: High computational costs as data grows.

3. Limited Personalization: Generic suggestions that may not reflect individual preferences.

4. Slow Adaptation: Inability to quickly adjust to new trends or user behavior.

5. Overfitting/Underfitting: Over-focus on popular items or fail to explore new ones.

6. Data Sparsity: Insufficient interaction or metadata for accurate recommendations.

7. Bias: Reinforces existing preferences, limiting diversity.

8. Data Integration: Difficulty combining multiple data sources for better insights.

These drawbacks highlight the need for more dynamic, hybrid systems.

1.5 Proposed System with Features:

The proposed hybrid recommendation engine is designed to combine multiple recommendation techniques to deliver more accurate, personalized, and context-aware suggestions to users. By leveraging both **collaborative filtering** (which analyzes user behavior and preferences in comparison to others) and **content-based filtering** (which focuses on item attributes and user profiles), the system overcomes the limitations of individual methods, such as cold-start problems and sparsity issues. This combination allows the engine to offer recommendations even when user data is limited and ensures that the suggestions are both relevant and diverse.A key feature of this system is its capability for **real-time processing**. The recommendation engine is optimized to handle incoming data instantly—such as user clicks, searches, and ratings—and update its suggestions accordingly without significant delays. This real-time adaptability enhances user satisfaction by providing fresh and timely content tailored to current interests. The underlying architecture supports stream processing and low-latency computation to maintain system responsiveness at all times.

In addition, the engine is designed to be **scalable using big data technologies**. It incorporates distributed computing frameworks such as Apache Spark and Hadoop to manage and analyze large volumes of user and item data efficiently. The system architecture is built to scale horizontally, ensuring that it can accommodate increasing data loads and a growing number of users without performance degradation. By using scalable storage solutions like HDFS and cloud-based services, the engine ensures data reliability and accessibility, which are crucial for enterprise-level deployment.

Overall, this hybrid recommendation engine aims to provide a smart, efficient, and scalable solution for delivering personalized experiences, making it ideal for e-commerce platforms, streaming services, and other data-intensive applications.

INPUT AND OUTPUT DESIGN

INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

•Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

•It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

•When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

• Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

- Select methods for presenting information.

• Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

• Convey information about past activities, current status or projections of the Future.

• Signal important events, opportunities, problems, or warnings.

• Trigger an action.

• Confirm an action.

2. LITERATURE SURVEY

1. AI Powered Job Recommendation System

Author: Durgesh Kumar Kushwaha (20-21 December 2024)

Artificial Intelligence (AI) is revolutionizing the recruiting sector by facilitating more intelligent and effective job matching procedures. This research proposes an AI-powered job recommendation system using machine learning and natural language processing (NLP). It matches job seekers with opportunities that best fit their profiles and preferences. By utilizing sophisticated algorithms that blend content-based and collaborative filtering, the system efficiently navigates large job listings and produces highly accurate and customized job suggestions. The effectiveness of the system is thoroughly assessed using precision, recall, and F1-score metrics, demonstrating its potential to transform the job search experience. Additionally, this paper examines the broader impact of AI-driven recommendation systems in recruitment, providing insights into future developments that could enhance the accuracy of job matching.

2. Comparing Collaborative Filtering Algorithms in AI-Powered Recommendation Systems for E-Commerce

Authors: Najmuddin Aamer, Tanusha Mittal (18-20 February 2025)

The quick growth of e-commerce has brought attention to how important efficient recommendation systems are to enhancing user experience and accomplishing business objectives. This paper investigates the effectiveness of collaborative filtering algorithms in AI-powered recommendation systems, with a focus on user-based collaborative filtering (UBCF). To decrease dimensionality and maintain the most significant patterns of user-item interaction, Principal Component Analysis (PCA) is employed for feature selection. The classification system uses similarity measures and UBCF to predict user preferences based on historical data. The algorithms are evaluated on key metrics like precision, recall, and RMSE using publically available datasets. The outcomes demonstrate the impact of preprocessing and dimensionality reduction on UBCF's performance as well as its efficacy in providing customised recommendations. This research contributes to the improvement of collaborative filtering algorithms for e-commerce systems, which will raise customer satisfaction and enable scalability in dynamic online markets.

3. AI Powered Recommender Systems for E-commerce

Authors: Gunjan Singh, Chintamani Sahasrabudhe (15-16 November 2024)

The disruptive influence that artificial intelligence-powered recommender systems can have on e-commerce platforms. The delivery of largely individualized buying gestures is accomplished by these systems by the utilization of sophisticated machine learning algorithms, which analyze enormous numbers of stoner data. When it comes to artificial intelligence, we investigate several methods, such as cooperative filtering, content-based filtering, and mongrel models, that improve the quality of recommendations and the level of satisfaction experienced by stoners. The research underlines the importance of having a high level of reading to comprehend the tastes of stoners and to forecast the purchases of unborn children. In the same vein, it examines the difficulties that will be encountered while implementing AI recommender systems. These difficulties include data sequestration businesses, algorithmic impulses, and scalability. The article explains how e-commerce businesses may optimize their recommendation machines to promote client engagement and increase deals by using case studies and empirical analysis to illustrate the effectiveness of these optimization strategies. In the end, this investigation highlights the significance of artificial intelligence in revolutionizing online buying, providing valuable insight for organizations that are looking to improve their competitive edge in the digital market.

4. An AI-Powered Digital Foundation Recommender System

Authors: D. M. Ruiz, A. Watson, Y. Kumar (22-25 October 2024)

This paper addresses the significant challenge of selecting suitable foundation shades, particularly for darker skin tones, which have been inadequately represented and catered to in the beauty industry. It introduces an innovative system designed to offer individualized makeup recommendations, significantly diminishing the time and effort traditionally demanded of consumers when searching for appropriate products. Utilizing machine learning methodologies and computer vision techniques, the system analyzes image data to precisely identify a range of skin tones, enabling it to propose compatible foundation shades automatically. Unlike sophisticated Large Language Models, such as ChatGPT, Gemini, Microsoft Copilot, or Claude, which are not equipped to undertake such visually driven tasks due to ethical guidelines that prohibit them from processing personal images without explicit consent and the absence of face image processing capabilities, this technological development represents a step forward in fostering inclusivity, illustrating the transformative potential of AI in accommodating the unique beauty preferences of all individuals.

5. Enhancing Search and Recommendation Systems

Authors: Xinke Wang, Yu Sun (09-11 October 2024)

This research paper explores the development and evaluation of a recommendation and search algorithm using OpenAI's embedding API and PineconeDB, focusing on creating an accessible, efficient, and scalable system capable of handling multimodal content [1]. The study is divided into several sections, including a comprehensive introduction to the challenges and proposed solutions for modern search systems, detailed method analysis, and experimental validation. Two primary experiments were conducted: the first assessed the search algorithm's semantic indexing accuracy with different datasets, and the second tested the system's capability to handle searches involving both textual and visual data. Results from the first experiment indicated that while the system performs exceptionally well with semantically rich content (92% accuracy), it struggles with content of lower semantic quality (58% accuracy), highlighting the need for advanced semantic processing techniques. The second experiment demonstrated the system's proficiency in multi-modal searches, achieving an average precision of 87% and recall of 80%, confirming its suitability for complex search tasks across various data types.

6. AI-Driven Personalized Dietary Recommendation

Authors: Divya J, Aakash G (05-07 February 2025)

Diet related health issues are a growing concern, with individuals often struggling to make informed dietary choices. Existing dietary recommendation systems rarely include regional foods, making it difficult for users to access or prepare the recommended meals. Despite the availability of a diverse array of nutritious local foods, this gap necessitates the development of a system that provides culturally relevant and accessible dietary recommendations. In this project, an AI based dietary recommendation system is designed to cater to regional needs. Using machine learning algorithms, the system calculates daily caloric requirements based on factors such as age, BMI, and individual health goals. Personalized meal plans are generated from regional foods, ensuring they are culturally appealing, nutritionally balanced, and aligned with seasonal availability. The outcome of this research is to bridge the gap in existing systems by generating meal plans rooted in regional foods, empowering users to achieve healthier diets through accessible and culturally resonant recommendations.

7. Enhanced Experiences: Benefits of AI-Powered Recommendation Systems

Authors: Kouayep Sonia Carole, Tagne Poupi Theodore Armand (2024)

Today, information technology has brought various innovations and developments in almost every field of advancement. Recommendation Systems (RS) have achieved a significant milestone in the service business during the information systems era. Regarding online services, RS has been crucial in enhancing product availability and offering prospective customers a wider range of luxurious options. Conversely, online retailers face increasing their sales volume and achieving greater product prices than their rivals. One solution is to use recommendation systems that leverage artificial intelligence (AI) to provide personalized recommendations to users. These systems employ machine learning algorithms to examine user data, including search history, purchasing patterns, and preferences, to anticipate the products that consumers are most likely to be interested in. AI-powered recommendation systems have demonstrated their immense value as tools for decision-making, enhancing user experience, and fostering corporate success. This comprehensive review explores the multifaceted world of recommendation systems, delving into their mechanisms, applications, and transformative impact across diverse domains. From e-commerce to content streaming and beyond, these systems have redefined how we discover, choose, and engage with products, content, and services.

8. AI-based College Course Selection Recommendation System

Authors: Yu Hsuan Wu, Eric Hsiaokuang Wu (13-16 November 2020)

Recent advances of AI applications in various of industries have led to remarkable performance and efficiency. Driven by the great success of datasets and experience sharing, people are exploring more precious datasets with diverse features and longer time range. The promising reasoning information of well-curated student grade datasets is expected to assist young students to find the best of themselves and then improve their learning outcome and study experience. Through data and experience sharing, young students can have a better understanding of their learning condition and possible learning outcomes. Existing course selection systems in Taiwan which offer limited basic enrolling functions fail to provide performance prediction and course arrangement guidance based on their own learning condition. Students now selecting courses with unawareness of their expecting performance. A personalized guide for students on course selection is crucial for how they structure professional knowledge and arrange study schedule. In this paper, we first analyzed what factors can be used on defining learning curve, and discovered the difference between students with different properties and background.

3. SOFTWARE REQUIREMENTS ANALYSIS

3.1 Problem Specification

In today's digital era, users are continuously interacting with a vast amount of content across various platforms, including e-commerce, streaming services, social media, and news portals. With the exponential growth of data, users often face challenges in discovering relevant content tailored to their preferences. This has created a critical need for accurate, real-time, and scalable recommendation systems that can efficiently process large datasets and deliver personalized suggestions. Traditional recommendation methods often struggle with issues such as cold-start problems, data sparsity, and lack of scalability, resulting in poor user experience and reduced engagement. Moreover, as user preferences evolve over time, recommendation systems must adapt dynamically to deliver updated and context-aware recommendations. The increasing demand for real-time interactions further emphasizes the necessity for low-latency systems capable of generating insights almost instantaneously. Therefore, the development of an intelligent, adaptive, and resource-efficient recommendation system is essential for enhancing user satisfaction, boosting platform engagement, and driving business growth. This problem calls for innovative solutions that combine machine learning, big data processing, and real-time analytics to address the limitations of existing models and meet the evolving expectations of users on modern digital platforms.

3.2 Modules and Their Functionalities

• **USER:**

The User can register the first. While registering he required a valid user email and mobile for further communications. Once the user register then admin can activate the customer. Once admin activated the customer then user can login into our system.First user has to give the input as image to the system. The python library will extract the features and appropriate emotion of the image. If given image contain more than one faces also possible to detect. The image processing completed the we are going to start the live stream. In the live stream also we can get the facial expression more that one persons also. Compare to tensorlflow live stream the tesnorflow live stream will fast and better results. Once done the we are loading the dataset to perform the knn classification accuracy precession scores.

• Admin:

Admin can login with his credentials. Once he login he can activate the users. The activated user only login in our applications. The admin can set the training and testing data for the project dynamically to the code. The admin can view all users detected results in hid frame. By click in gan hyperlink in the screen he can detect the emotions of the images. The admin can also view the knn classification detected results. The dataset in the excel format. By authorized persons we can increase the dataset size according the imaginary values.

• DATA PREPROCESS:

Dataset contains grid view of already stored dataset consisting numerous properties, by Property Extraction newly designed dataset appears which contains only numerical input variables as a result of Principal Component Analysis feature selection transforming to 6 principal components which are Condition (No stress, Time pressure, Interruption), Stress, Physical Demand, Performance and Frustration.

• MACHINE LEARNING:

K-Nearest Neighbor (KNN) is used for classification as well as regression analysis. It is a supervised learning algorithm which is used for predicting if a person needs treatment or not. KNN classifies the dependent variable based on how similar it is; independent variables are to a similar instance from the already known data.theKnn Classification can be called as a statistical model that uses a binary dependent variable. In classification analysis, KNN is estimating the parameters of a KNN model. Mathematically, a binary KNN model has a dependent variable with two possible value, which is represented by an indicator variable, where the two values are labeled "0" and "1".

3.3 Functional Requirements

1. Data Ingestion from Multiple Sources

    ○ Support for Diverse Sources: APIs, databases, IoT devices, flat files, and web scraping.

    ○ ETL Process: Extract, transform, and load data efficiently.

    ○ Data Validation & Error Handling: Ensure data quality and manage failures.

2. Real-time Prediction

    ○ Continuous Data Streaming: Real-time data ingestion with low latency.

○ Predictive Models: Use machine learning models for dynamic predictions.

○ Scalability: Infrastructure that handles fluctuating data loads.

○ Prediction API: Expose real-time predictions for integration.

3. User Feedback Loop

○ Feedback Collection: Interface for users to provide feedback.

○ Model Improvement: Use feedback to retrain models and improve accuracy.

○ Personalization: Adjust predictions based on user input over time.

3.4 Non Functional Requirements

Scalability

● The ability of a system to handle increased load by adding resources, either vertically

High Availability (HA)

● Ensures continuous operation with minimal downtime through redundancy, failover mechanisms, and load balancing.

Fault Tolerance

● Allows a system to continue functioning despite component failures through error detection, redundancy, and graceful degradation.

3.5 FEASIBILITY STUDY

The project is technically feasible using existing big data frameworks (e.g., Hadoop, Spark) and machine learning libraries (e.g., TensorFlow, PyTorch, scikit-learn). These frameworks efficiently handle large-scale data processing and complex models, enabling the extraction of insights and predictions from massive datasets.

Three key considerations involved in the feasibility analysis are,

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILIT

4. SOFTWARE AND HARDWAREREQUIREMENTS

4.1 Software Requirements

Operating system : Windows 10

Coding Language : Python

Frontend : HTML

Libraries & Frameworks

- Apache Spark: Big data processing
- Pandas: Data manipulation
- NumPy: Numerical computing
- Scikit-learn: Machine learning

4.2 Hardware Requirements

System	:	Intel Core i5
Hard Disk	:	1 TB.
Monitor	:	15” LED
Input Devices	:	Keyboard, Mouse
Ram	:	16 GB.

5. SOFTWARE DESIGN

5.1 System Architecture

The methodology of review consists following steps

- Data Collection
- Data Pre-Processing
- Feature Extraction
- Model Training
- Testing
- Recommendation Generation

5.2 Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It may be used as a communication tool between a system analyst andany person who plays a part in the order that acts as a starting point for redesigning a system. The DFD is also called as a data flow graph or bubble chart.

The Basic Notation used to create a DFD's are as follows:

• Dataflow: Data move in a specific direction from an origin to a destination.

• Process: People, procedures, or devices that use or produce (Transform) Data.

• Source: External sources or destination of data, which may be People, programs.

• Data Store: Here data are stored or reference by a process in the System.

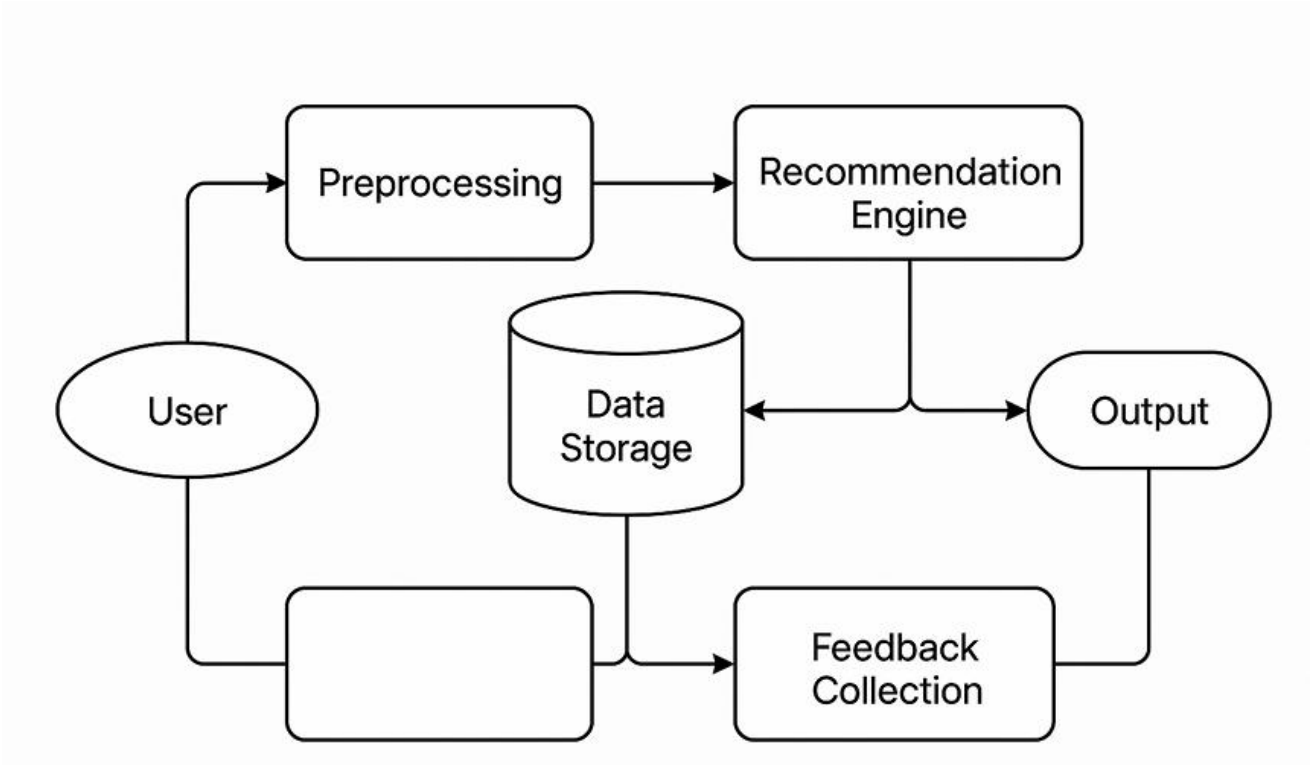


Figure 5.2 Data flow diagram

5.3 UML Diagrams

UML (Unified Modeling Language) diagrams are visual representations that help in designing and understanding complex software systems. They showcase the structure, behavior, and interactions within a system. Common types of UML diagrams include:

Below are the types of UML diagrams included in this project:

1. Use Case Diagram

- Illustrates the interactions between the user and the system.
- Captures the core functionalities: uploading a CSV file, data preprocessing, visual analysis, and mathematical insights generation.
- Helps understand user requirements and system boundaries.

2. Class Diagram

- Represents the static structure of the system.

• Shows classes like SalesDataProcessor, Visualizer, and their attributes and methods.

• Displays relationships such as associations and dependencies between classes.

3. Sequence Diagram

• Demonstrates the flow of interactions between system components over time.

• Captures the sequence of operations from data upload to visualization rendering.

• Useful for understanding system behavior and component collaboration.

4. Activity Diagram

• Describes the workflow of the application.

• Covers major stages such as data input, preprocessing, visualization selection, and output display.

• Useful for modeling dynamic aspects and conditional flows.

Class Diagram

A Class Diagram describes the static structure of a system by showing its classes, their attributes, methods, and the relationships among them. It forms the foundation of objectoriented modeling and is crucial for understanding how data is organized, reused, and interacted with across different parts of the system

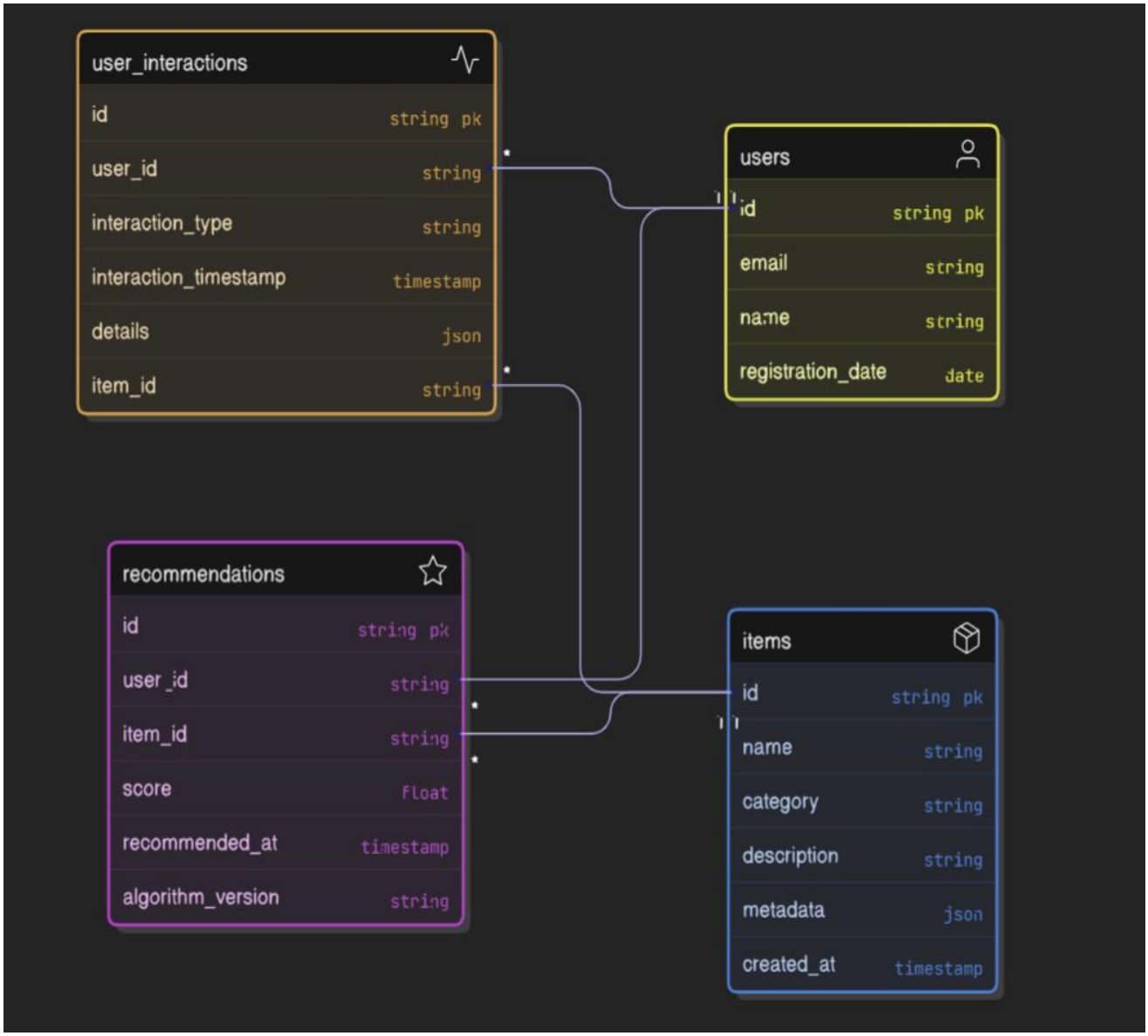


Figure 5.3.1 Class Diagram

Sequence Diagram

A Sequence Diagram illustrates the flow of messages and interactions between objects or components in a system over time. It helps visualize the order of operations, making it easier to understand how different parts of the system collaborate during a specific process.

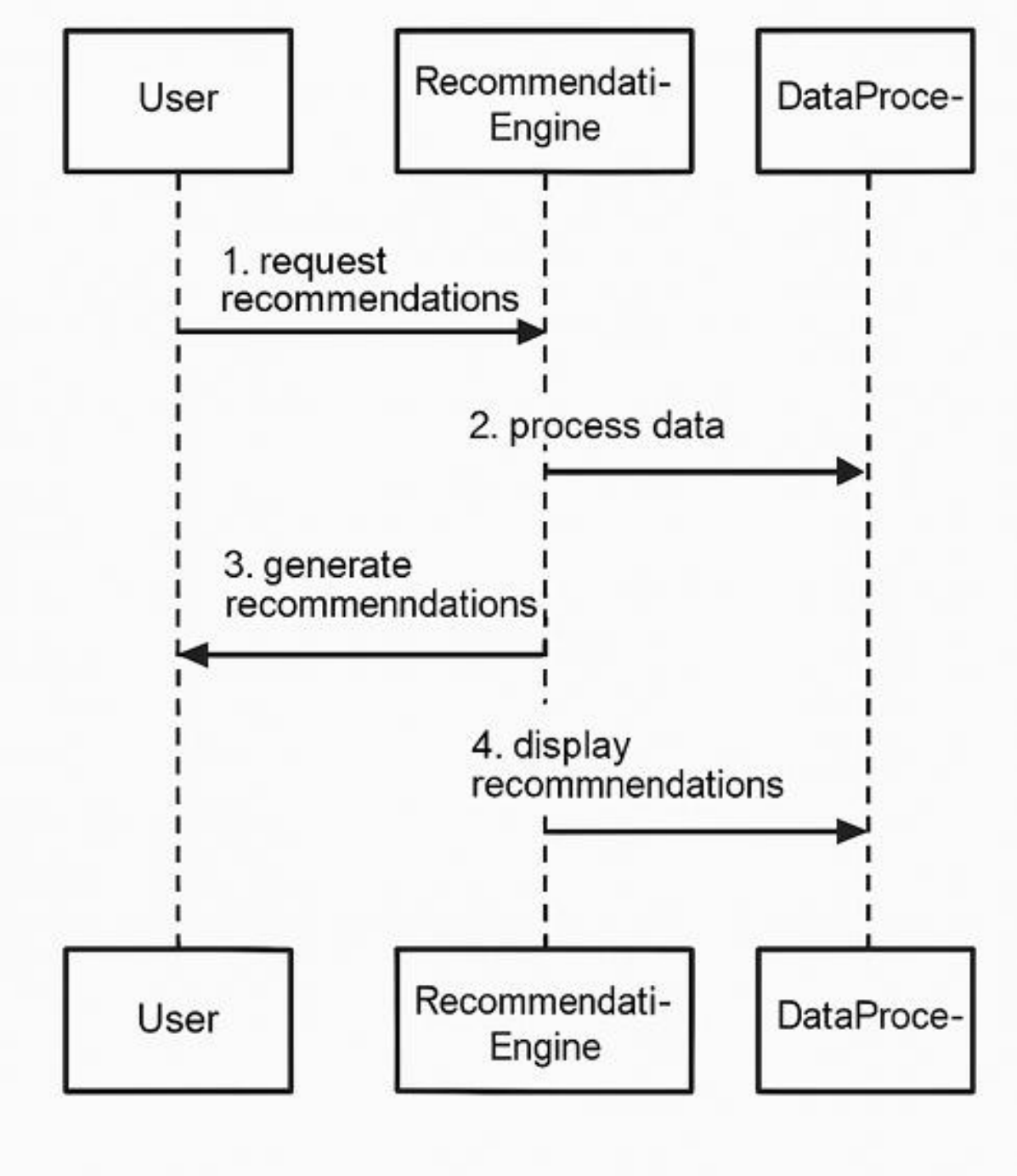


Figure 5.3.2 Sequence Diagram

Use Case diagram

A Use Case Diagram represents the functionality of a system from the end user's perspective. It shows the interactions between actors (users or external systems) and use cases (system functionalities), helping to identify user needs and the system's scope clearly.

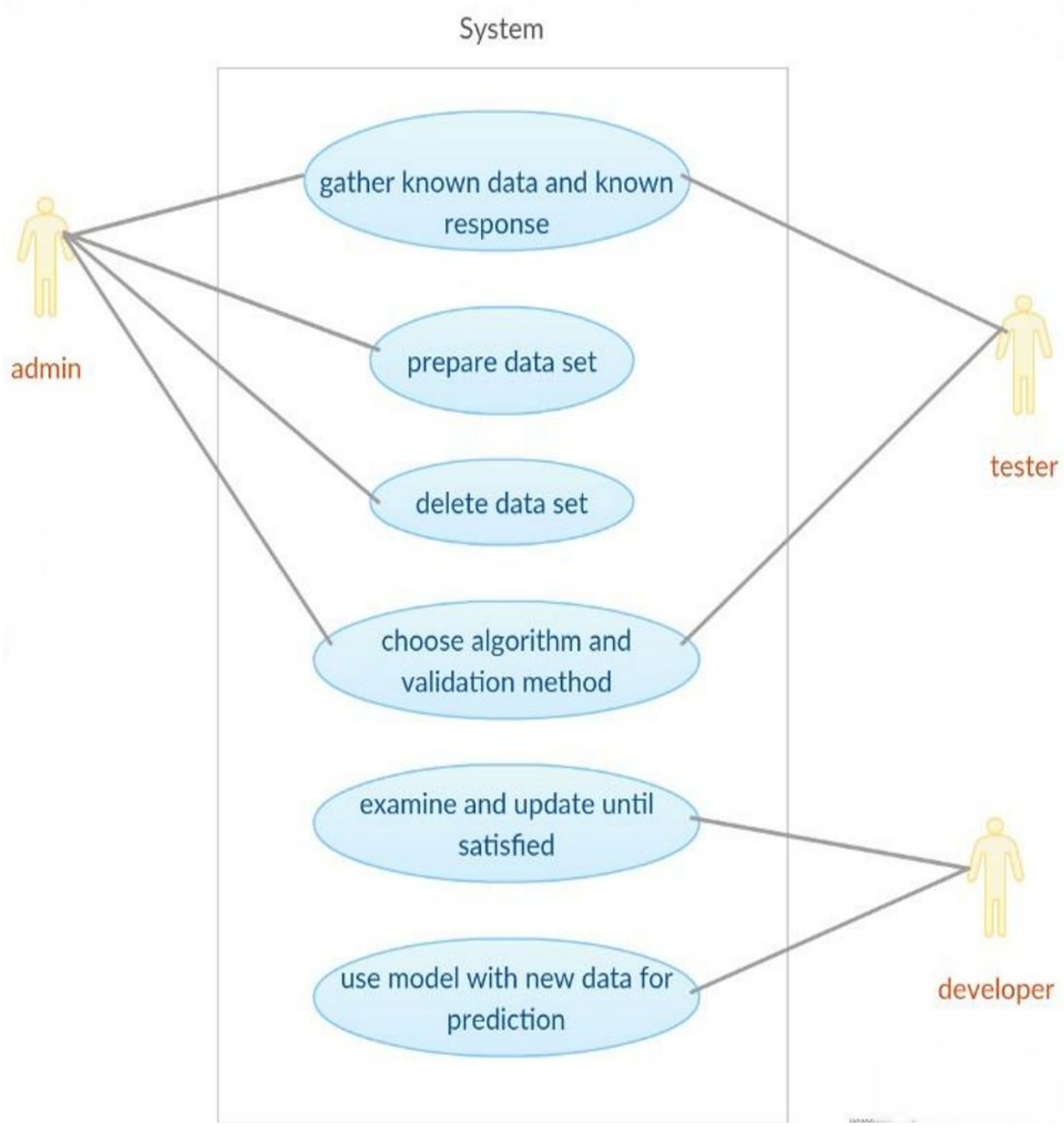


Figure 5.3.3 Use Case Diagram

Activity Diagram

An Activity Diagram models the workflow or business processes within a system. It illustrates the sequence of activities, decision points, parallel processes, and start/end points. These diagrams help in visualizing how users or systems interact with various actions, especially useful in logic-heavy scenarios like data analysis pipelines or user interaction flows.

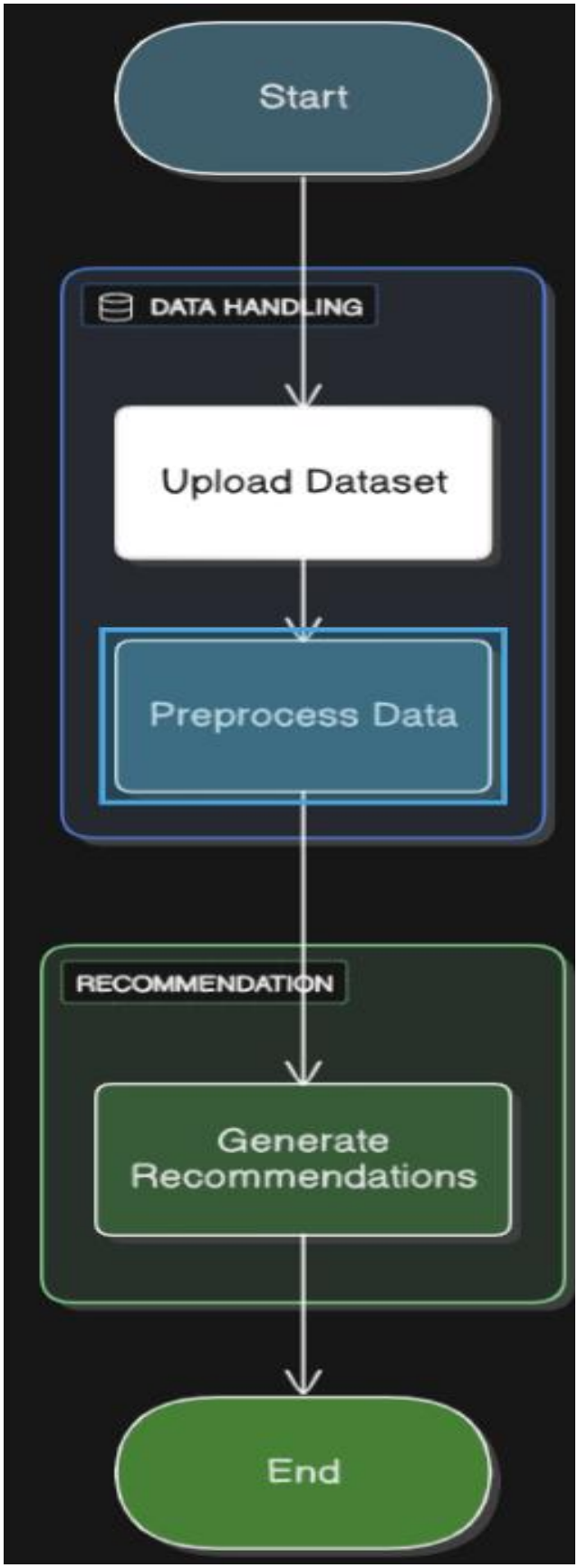


Figure 5.3.4 Activity Diagram

6. CODING AND ITS IMPLEMENTATION

6.1 Source code

```
import streamlit as st

import pandas as pd


# Title

st.set_page_config(page_title="Game Recommender", layout="centered")

st.title(" 🎮 AI-Powered Recommendation System")

st.subheader("Recommended Games:")

# Load data

@st.cache_data

def load_data():

    df = pd.read_csv("video_game_reviews.csv")

    return df[['Game Title', 'User Rating', 'Price', 'Genre']].dropna()


df = load_data()


# Sidebar Filters

st.subheader("Filters")


# Genre Dropdown

genre_list = ["-- Any Genre --"] + sorted(df['Genre'].dropna().unique())

selected_genre = st.selectbox("Genre:", genre_list)
```

```
# Price Range
min_price = float(df["Price"].min())
max_price = float(df["Price"].max())

col1, col2 = st.columns(2)
with col1:
    price_min = st.number_input("Min price", min_value=min_price, max_value=max_price, value=min_price, step=0.5, format="%.2f")
with col2:
    price_max = st.number_input("Max price", min_value=min_price, max_value=max_price, value=max_price, step=0.5, format="%.2f")

# Filter logic
def get_recommendations(genre, pmin, pmax):
    filtered = df[(df['Price'] >= pmin) & (df['Price'] <= pmax)]
    if genre != "-- Any Genre --":
        filtered = filtered[filtered['Genre'].str.lower() == genre.lower()]
    return filtered.sort_values(by='User Rating', ascending=False).head(5)

# Get Recommendations Button
if st.button("Get Recommendations"):
    st.subheader("Recommended Games:")
    results = get_recommendations(selected_genre, price_min, price_max)
    if results.empty:
```

```
        st.write("_No recommendations available with the selected filters_")
    else:
        st.dataframe(results.reset_index(drop=True))
```

6.2 Implementation

This section describes the practical implementation of the End-to-End Financial Analysis Platform using Python and Streamlit. The solution is modularly designed for loading data, cleaning and preprocessing, computing KPIs, and generating insightful visualizations. Below

are the key components:

6.2.1 Application Setup

- The app is built using the Streamlit framework for creating interactive web applications in Python.
- Initial configurations, such as setting the page title and layout, are done using st.set_page_config().
- Custom CSS styling is applied using st.markdown() to improve visual aesthetics (e.g., button colors, sidebar background, and data frame box styling).

6.2.2 File Upload and Data Ingestion

- The application allows the user to upload a CSV file containing sales data using st.file_uploader().
- Upon upload, the CSV is read using pandas.read_csv() and displayed using st.dataframe() for a quick preview.

6.2.3 Data Cleaning

• The app handles basic data cleaning operations:

o Trims column names to remove whitespace.

o Displays the count of missing and duplicate values.

o Drops missing (NaN) and duplicate records using dropna() and

drop_duplicates() to ensure analysis is based on reliable data.

o Success indicators (st.success()) inform the user that the cleaning is complete.

6.2.4 Filtering the Dataset

Once the user clicks the **"Get Recommendations"** button, the app executes the core recommendation logic:

- **Price Filtering**:
  - The app first filters the dataset to include only those games whose `Price` falls between the user-defined `price_min` and `price_max` values.
  - This step ensures the recommendations match the user's budget.
- **Genre Filtering**:
  - If the user has selected a specific genre (i.e., not "-- Any Genre --"), the app further filters the dataset to include only games that belong to that genre.
  - The filtering is case-insensitive to avoid mismatches due to inconsistent text casing in the dataset

7.SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

7.1 TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted. Invalid Input : identified classes of invalid input must be rejected. Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised. Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

7.2 TESTING STRATEGIES

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

8.OUTPUTSCREEN

Task 1: upload code to GitHub

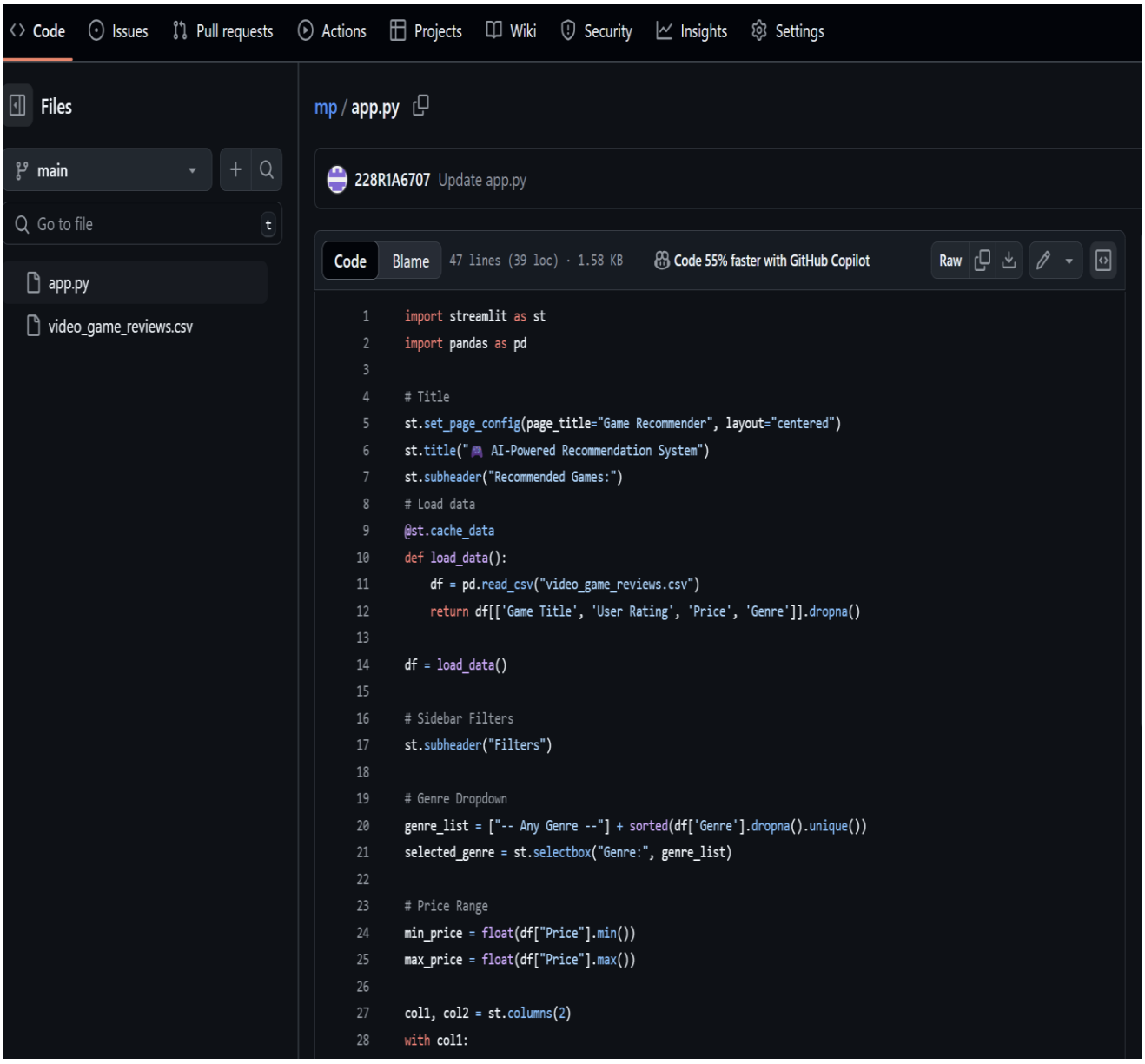
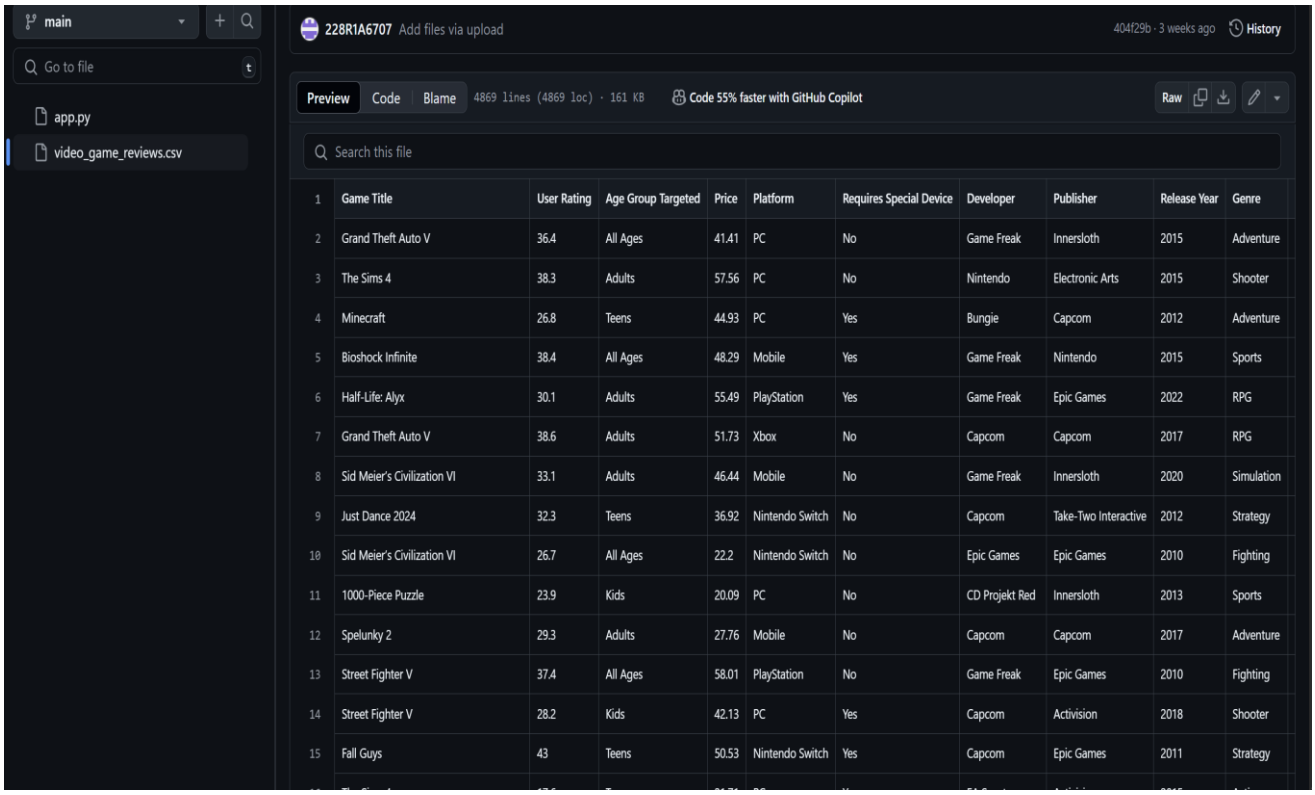


Figure 8.1: Implementation

Task 2: upload the Dataset



	Game Title	User Rating	Age Group Targeted	Price	Platform	Requires Special Device	Developer	Publisher	Release Year	Genre
2	Grand Theft Auto V	36.4	All Ages	41.41	PC	No	Game Freak	Innersloth	2015	Adventure
3	The Sims 4	38.3	Adults	57.56	PC	No	Nintendo	Electronic Arts	2015	Shooter
4	Minecraft	26.8	Teens	44.93	PC	Yes	Bungie	Capcom	2012	Adventure
5	Bioshock Infinite	38.4	All Ages	48.29	Mobile	Yes	Game Freak	Nintendo	2015	Sports
6	Half-Life: Alyx	30.1	Adults	55.49	PlayStation	Yes	Game Freak	Epic Games	2022	RPG
7	Grand Theft Auto V	38.6	Adults	51.73	Xbox	No	Capcom	Capcom	2017	RPG
8	Sid Meier's Civilization VI	33.1	Adults	46.44	Mobile	No	Game Freak	Innersloth	2020	Simulation
9	Just Dance 2024	32.3	Teens	36.92	Nintendo Switch	No	Capcom	Take-Two Interactive	2012	Strategy
10	Sid Meier's Civilization VI	26.7	All Ages	22.2	Nintendo Switch	No	Epic Games	Epic Games	2010	Fighting
11	1000-Piece Puzzle	23.9	Kids	20.09	PC	No	CD Projekt Red	Innersloth	2013	Sports
12	Spelunky 2	29.3	Adults	27.76	Mobile	No	Capcom	Capcom	2017	Adventure
13	Street Fighter V	37.4	All Ages	58.01	PlayStation	No	Game Freak	Epic Games	2010	Fighting
14	Street Fighter V	28.2	Kids	42.13	PC	Yes	Capcom	Activision	2018	Shooter
15	Fall Guys	43	Teens	50.53	Nintendo Switch	Yes	Capcom	Epic Games	2011	Strategy

Figure 8.2: Upload Dataset

Task 3: upload and run the file on streamlit

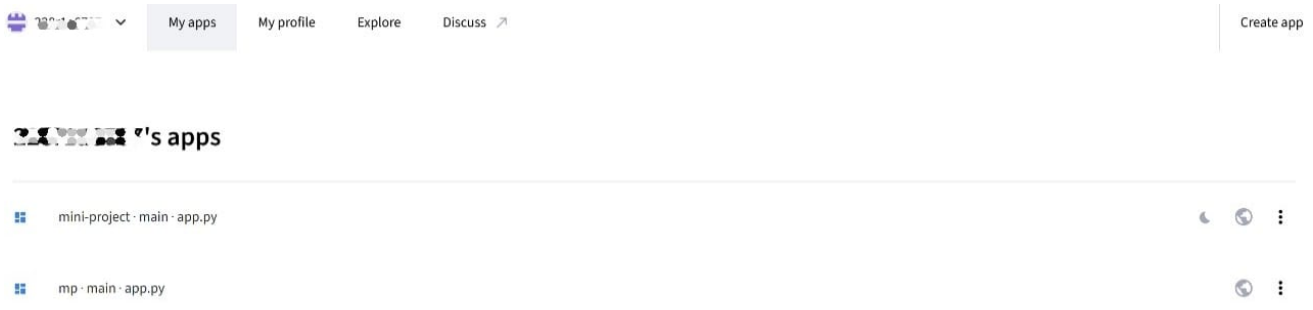


Figure 8.3 : csv file upload

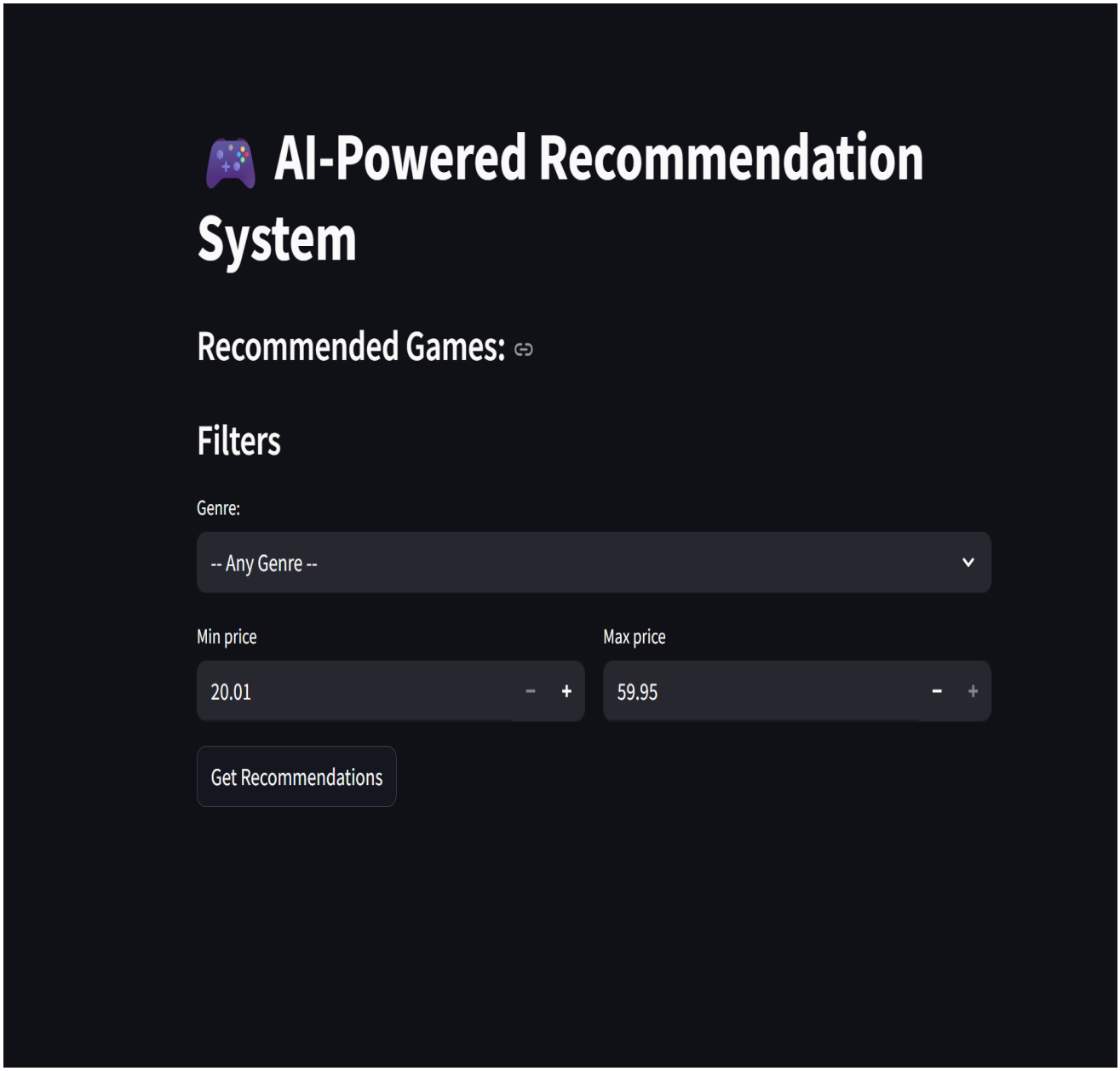


Figure 8.4 : Recommendation interface

Task 5:

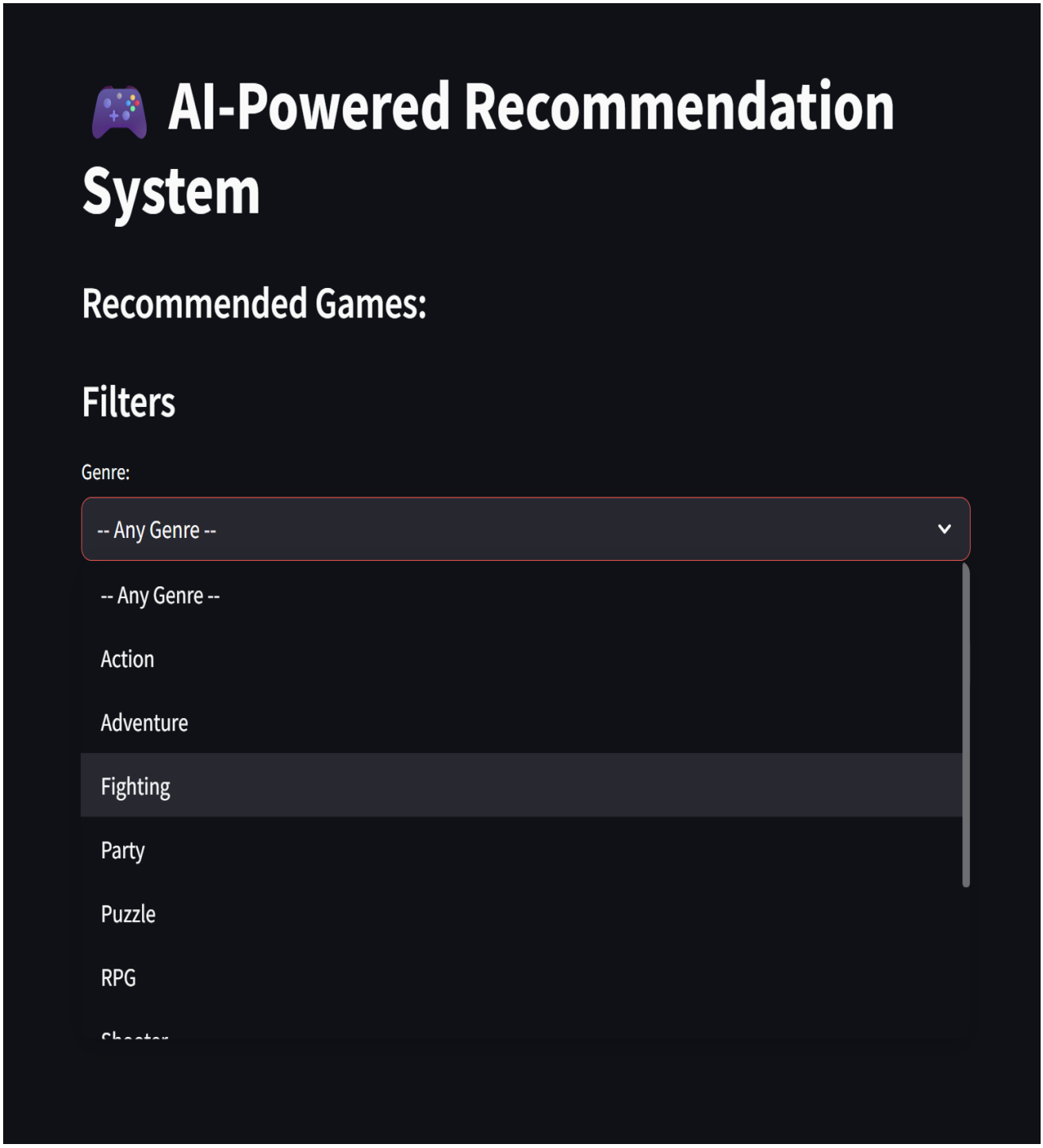


Figure 8.5 : User interaction

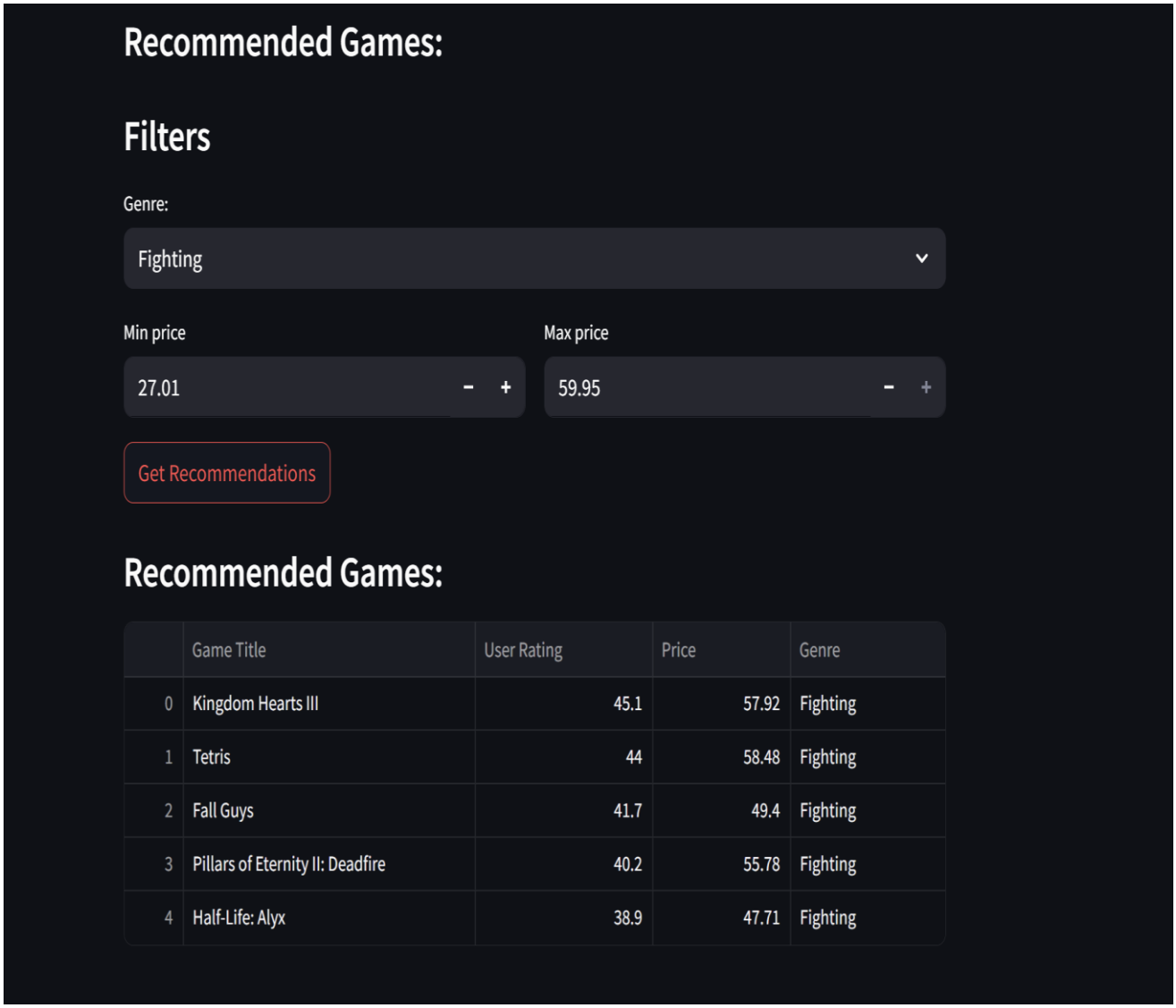


Figure 8.6: Recommendations generation

CONCLUSION

The AI-powered recommendation system utilizes a combination of hybrid modeling techniques to provide highly accurate and personalized suggestions to users. By integrating multiple recommendation methods, such as collaborative filtering, content-based filtering, and matrix factorization, the system is able to leverage the strengths of each approach to overcome their individual limitations.

Collaborative filtering, for instance, relies on user behavior data to recommend items based on the preferences of similar users. Meanwhile, content-based filtering utilizes metadata or item characteristics to match users with items similar to those they have shown interest in. By blending these methods, the system can provide more robust recommendations, even in cases where one method might struggle—such as in new-user or new-item scenarios, which is often referred to as the "cold start" problem.

Matrix factorization, another key technique used in the system, decomposes large datasets into latent factors that can be used to predict missing or unseen interactions. This enhances the system’s ability to uncover deeper patterns in user preferences and item relationships.

Through the implementation of these hybrid models, the recommendation system is capable of continuously learning from user interactions, adapting to evolving preferences over time. The result is a highly personalized experience, where users are regularly presented with items they are likely to find engaging or relevant. This not only boosts user satisfaction but also increases user engagement by ensuring that the recommendations are timely, diverse, and contextually appropriate.

In summary, the AI-powered recommendation system optimizes the user experience by leveraging advanced hybrid modeling techniques, making it a powerful tool for enhancing satisfaction, engagement, and long-term user retention.

FUTURE ENHANCEMENTS

1. Include Deep Learning Models

    ○ Overview: Deep learning models are a subset of machine learning that utilize neural networks with many layers to learn from vast amounts of data. These models have shown exceptional performance in tasks such as image recognition, natural language processing, and speech recognition.

    ○ Implementation:

        ■ Neural Networks: You can integrate pre-built models such as Convolutional Neural Networks (CNNs) for image tasks or Long Short-Term Memory (LSTM) networks for sequence tasks like time-series prediction or natural language processing.

        ■ Frameworks: Leveraging frameworks such as TensorFlow, PyTorch, or Keras can simplify the implementation and training of deep learning models.

        ■ Pre-trained Models: For efficient development, pre-trained models such as BERT for text or ResNet for images can be used as a starting point and fine-tuned based on the specific requirements of your project.

    ○ Benefits: Deep learning models can significantly improve the accuracy and robustness of your system, particularly in areas involving large amounts of unstructured data (e.g., images, audio, or text).

2. Real-Time Model Retraining

    ○ Overview: Real-time model retraining refers to the ability of your system to update its machine learning models continuously or periodically based on new incoming data.

    ○ Implementation:

        ■ Data Pipeline: Create a pipeline that continuously collects new data and preprocesses it in real-time. This can be integrated with Apache Kafka or other stream processing tools to ensure that fresh data is ingested as it becomes available.

■ Model Monitoring: The system should monitor the performance of the model over time. If there is a significant drop in model accuracy, retraining can be triggered automatically.

■ Model Retraining Strategy: The retraining can either be incremental, where the model is updated in small steps as new data comes in, or batch retraining, where the model is retrained periodically with a batch of accumulated new data.

■ AutoML Integration: Use AutoML frameworks like Google AutoML, H2O.ai, on Microsoft Azure AutoML to automate the retraining process, making it more efficient and scalable.

○ Benefits: Real-time model retraining ensures that your model adapts to new data patterns and remains accurate and relevant over time, especially in dynamic environments where the data distribution changes frequently.

3. Multi-Language Support

○ Overview: Multi-language support allows your system to work seamlessly with multiple languages, making it more inclusive and accessible to a global user base.

○ Implementation:

■ Natural Language Processing (NLP): Leverage NLP techniques like tokenization, named entity recognition, and part-of-speech tagging to process text in different languages. Libraries like spaCy, NLTK, and Hugging Face's transformers can help achieve this.

■ Translation Models: Use neural machine translation models, such as Google Translate API or OpenNMT, to automatically translate text from one language to another. For high accuracy, fine-tune pre-trained translation models with domain-specific data.

■ Language Detection: Implement language detection algorithms (e.g., langid.py or the langdetect library in Python) to identify the language of incoming text and route it accordingly for processing.

■ User Interface (UI): The UI should be designed to allow users to select their preferred language. It should also be able to adjust dynamically to different languages without breaking the system’s functionality.

○ Benefits: Multi-language support increases user engagement and expands your system's usability across different geographies and cultures, especially for applications like customer support, content recommendation, or e-commerce platforms.

REFERENCES

[1] M. Zareapoor and P. Shamsolmoali, "Application of Credit Card Fraud Detection: Based on Bagging Ensemble Classifier," International Conference on Intelligent Computing, Communication, & Convergence (ICCC-2016), pp. 59-65, 2016.

[2] A. K. Jain, "Data Mining: Concepts and Techniques," Morgan Kaufmann, 2016.

[3] J. M. Kleinberg, "The Convergence of Social Networks and Recommendation Systems," IEEE Transactions on Knowledge and Data Engineering, vol. 27, no. 7, pp. 1705-1718, July 2015.

[4] R. Salakhutdinov and A. Mnih, "Bayesian Collaborative Filtering," Proceedings of the 20th International Conference on Neural Information Processing Systems, Vancouver, Canada, 2007.

[5] Divya J, Aakash G (05-07 February 2025), "AI-Driven Personalized Dietary Recommendation", IEEE Computer, AI-Driven Personalized Dietary Recommendation System Leveraging Regional Cuisine | IEEE Conference Publication | IEEE Xplore.

[6] Gunjan Singh, Chintamani Sahasrabudhe (15-16 November 2024), "AI Powered Recommender Systems for E-commerce", IEEE Computer, AI Powered Recommender Systems for E-commerce | IEEE Conference Publication | IEEE Xplore.

[7] Kouayep Sonia Carole, Tagne Poupi Theodore Armand (2024)," **Enhanced Experiences: Benefits of AI-Powered Recommendation Systems**", IEEE Computer, Enhanced Experiences: Benefits of AI-Powered Recommendation Systems | IEEE Conference Publication | IEEE Xplore.

[8] Najmuddin Aamer & Tanusha Mittal (18-20 February 2025), "Comparing Collaborative Filtering Algorithms in AI-Powered Recommendation Systems for E-Commerce", IEEE Computer, Comparing Collaborative Filtering Algorithms in AI-Powered Recommendation Systems for E-Commerce | IEEE Conference Publication | IEEE Xplore.

[9] Durgesh Kumar Kushwaha (20-21 December 2024), "AI Powered Job Recommendation System", IEEE Computer, AI Powered Job Recommendation System | IEEE Conference Publication | IEEE Xplore.

[10] D. M. Ruiz, A. Watson, Y. Kumar (22-25 October 2024), " An AI-Powered Digital Foundation Recommender System", IEEE Computer, An AI-Powered Digital Foundation Recommender System | IEEE Conference Publication | **IEEE** Xplore.

[11] R. Golagana, V. Sravani, T. M. Reddy, and C. H. Kavitha, “Diet recommendation system using machine learning,” *Journal of Health and Nutrition*, vol. 15, no. 2, pp. 233–245, 2023. Google Scholar

[12] Christman J (2004) Relational autonomy, liberal individualism, and the social constitution of selves. Philos Studies 117(1–2):143–164. https://doi.org/10.1023/b:phil.0000014532.56866.5c

[13] Burr C, Cristianini N, Ladyman J (2018) An analysis of the interaction between intelligent software agents and human users. Minds Mach 28(4):735–774. https://doi.org/10.1007/s11023-018-9479-0

[14] Danaher J (2018) Toward an ethics of AI assistants: an initial framework. Philos Technol 31(4):629–653. https://doi.org/10.1007/s13347-018-0317-3

[15] de Vries K (2010) Identity, profiling algorithms and a world of ambient intelligence. Ethics Inf Technol 12(1):71–85. https://doi.org/10.1007/s10676-009-9215-9

[16] Dorrestijn S (2012) Technical mediation and subjectivation: tracing and extending foucault’s philosophy of technology. Philos Technol 25(2):221–241. https://doi.org/10.1007/s13347-011-0057-0