Marketplace for Startups with Recommender System

A PROJECT

Submitted by

Ravi Prakash (2021424490)

Dhiraj Giri (2021356392)

Shivam Rustagi (2020518381)

In partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

Computer Science & Engineering

Under the supervision of

Dr. Pushpendra Kr. Rajput
Associate Professor
Department of Computer Science & Engineering

SHARDA SCHOOL OF ENGINEERING AND TECHNOLOGY

SHARDA UNIVERSITY, GREATER NOIDA – 201310

MAY, 2025

TABLE OF CONTENTS

DECLARATION

I hereby declare that the project work entitled “Marketplace for Startups with Recommender System” submitted to Sharda University, Greater Noida, is a record of original work carried out by me under the guidance of Dr. Pushpendra Kumar Rajput. This project is being submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science & Engineering.

Place: Greater Noida	Signature of the Student-1
Date:	Signature of the Student-2
	Signature of the Student-3

CERTIFICATE

This is to certify that the report entitled "**Marketplace for Startups with Recommender System**" submitted by Ravi Prakash (2021424490), Dhiraj Giri (2021356392), and Shivam Rustagi (2020518381) to Sharda University, towards the fulfillment of requirements for the degree of Bachelor of Technology, is a record of bonafide final year project work carried out by them in the Department of Computer Science and Engineering, School of Engineering and Technology, Sharda University.

Signature of Supervisor

Name: Dr. Pushpendra Kr. Rajput

Designation: Assoc. Professor (CSE)

Signature of Head of Department

Name: Dr. Sudeep Varshney

Place: Sharda University

Date: 29/03/2025

ACKNOWLEDGEMENT

A major project is a golden opportunity for learning, growth, and self-development. We consider ourselves extremely fortunate and honored to have received guidance and support from several esteemed individuals throughout the completion of this project.

First and foremost, we express our sincere gratitude to **Dr. Sudeep Varshney**, Head of the Department, Computer Science and Engineering, for giving us the opportunity to undertake this project and for providing continuous support and encouragement.

We would also like to extend our heartfelt thanks to our project guide, **Dr. Pushpendra Kumar Rajput**, for his valuable guidance, constant support, and encouragement throughout the course of this project. Despite his busy academic schedule, he generously took the time to mentor us, address our challenges, and ensure we remained focused and on the right path.

We are also grateful to the CSE Department of Sharda University for monitoring our progress, arranging necessary resources, and facilitating a smooth project experience. We take this opportunity to acknowledge their invaluable contribution and support with deep appreciation.

Ravi Prakash (2021424490)

Giri Dhiraj (2021356392)

Shivam Rustagi (2020518381)

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

CHAPTER 1:
INTRODUCTION

INTRODUCTION AND OVERVIEW

The startup ecosystem in India has experienced remarkable growth over the past few years, establishing itself as one of the most dynamic and rapidly evolving entrepreneurial landscapes globally. Various studies project that by 2025, India will host more than 100 unicorns—startups valued at $1 billion or higher—and that hundreds of new ventures will emerge annually across diverse sectors such as e-commerce, fintech, edtech, healthtech, and agritech. Contributing to this surge are factors like a young, tech-savvy population, increasing digital penetration, supportive government initiatives like "Startup India," and a growing network of venture capital and angel investors. However, underlying this remarkable growth are several persistent challenges that hinder the expansion and survival of many companies.

One of the main obstacles, as noted, is visibility. With countless startups competing for attention in a saturated market, it becomes incredibly challenging to differentiate oneself, particularly for early-stage companies with restricted marketing budgets or networks. Furthermore, establishing connections with the right investors— those whose investment philosophies align with a startup's vision, industry, and growth stage—often proves to be a random endeavor. Conventional approaches, such as pitching at events or reaching out cold, are labor-intensive and produce varying outcomes. Likewise, targeting potential customers who could benefit from a startup's product or service poses difficulties due to fragmented market access and the lack of specialized channels to bridge this divide.

Traditional online marketplaces, although beneficial for general commerce, do not adequately meet the specific needs of the startup ecosystem. Platforms such as LinkedIn or standard B2B marketplaces lack the tailored approach necessary to effectively connect startups with investors, mentors, or customers. For example, an investor with an interest in sustainable energy startups may receive an overwhelming number of unrelated pitches from different sectors, while a startup in search of Series A funding may find it difficult to locate venture capitalists who specialize in their field. This lack of efficiency creates a barrier, hindering the growth of startups and diminishing the returns for stakeholders.

In order to tackle these challenges, the suggested Marketplace for Startups, equipped with a Recommender System, presents an enticing solution. This platform is intended to serve as a dedicated online hub specifically created for the startup ecosystem, linking startups, investors, customers, and even supporting service providers (like legal or marketing firms). The inclusion of a Recommender System enhances this marketplace beyond being merely a static directory or listing service. By utilizing sophisticated algorithms—possibly driven by machine learning and artificial intelligence—the system would examine user data, including preferences, previous interactions, industry focus, investment history, and behavioral trends, to offer highly customized recommendations. For instance:

For Startups: The system might suggest investors who have backed similar projects in the past, potential clients aligned with market demands, or even strategic collaborators for partnerships.

For Investors: It may propose companies aligned with their portfolio interests, risk tolerance, and desired investment phases, hence optimizing due diligence and transaction flow.

For Customers: It may emphasize new goods or services customized to customer need, promoting early acceptance and allegiance.

This tailored method would considerably boost efficiency and relevance compared to standard markets. Imagine a healthtech company in Bengaluru being immediately connected with a Mumbai-based angel investor who specializes in healthcare innovation, or a rural client finding an agritech solution ideally suited to their agricultural needs—all enabled effortlessly by data-driven insights.

Beyond matching, the platform might contain more functions to improve value production. For instance, it may provide analytics dashboards for companies to analyze their visibility and engagement metrics, virtual pitch rooms for investor meetings, or a content center where entrepreneurs present their goods via videos, case studies, or testimonials. The Recommender System might improve over time, learning from user comments and results (e.g., successful financing rounds or customer acquisitions) to refine its ideas further.

The potential influence of such a system is varied. For entrepreneurs, it offers more visibility, access to funding, and market penetration, allowing them to expand quicker. For investors, it minimizes the noise and speeds the identification of high-potential possibilities, improving their investing strategy. For consumers, it gives a tailored doorway to new solutions, boosting their experience and confidence in developing businesses. Collectively, this might boost the Indian startup ecosystem, making it more linked, efficient, and internationally competitive.

To bring this vision to reality, the creation of the marketplace would need a powerful technical backbone—cloud-based infrastructure, secure data processing, and scalable algorithms—alongside a user-friendly interface to encourage acceptance across broad user groups. Partnerships with incubators, accelerators, and industry groups might further extend its reach, while continual development based on user feedback would keep it aligned with developing ecosystem demands.

Motivation

The major motivation for launching this project comes from realizing that there aren't many platforms out there created exclusively to aid businesses by connecting them with the appropriate individuals and giving tailored support. Most of the existing markets simply provide you basic lists and search choices, which don't really do a great job of serving the special requirements of startups and the investors who may want to support them.

Here’s what’s driving this idea forward:

- We aim to make it simpler for entrepreneurs to be recognized by providing sophisticated recommendations that spotlight them to the relevant audience.
- We’d want to assist investors uncover fascinating businesses that fit what they’re sarching for, so they don’t have to sift through countlespossibilities.
- We’re attempting to make the overall experience better for consumers by proposing items or services that genuinely suit what they require
- And lastly, we’re designing a platform that can expand over time, utilizing artificial intelligence and machine learning to help things work smoother and more effectively.

Overview

This project is all about creating an online platform that acts like a marketplace specifically for startups, built from the ground up using a set of modern web development tools known as the MERN stack, which is a combination of MongoDB (a database), Express.js (a framework for handling server stuff), React.js (for building the user interface), and Node.js (the engine that keeps it all running on the server side). On top of that, we're weaving in a smart recommendation system.

Here’s what you can expect from the platform:

User Profiles: There will be separate locations for companies to showcase who they are, investors to describe what they're seeking for, and consumers to interact with what's available.

Startup Listings: Startups may provide all the juicy facts about themselves, including descriptions, images, videos, and anything else that helps convey their narrative.

Personalized Recommendation Engine: This is the smart part! It will recommend businesses or services to customers based on their preferences and activity, similar to how Netflix decides which programs you would like.

Admin Panel: A behind-the-scenes dashboard where platform management may monitor activity, censor content, and see performance statistics.

The recommendation system works by using two major techniques: collaborative filtering, which looks at what comparable users enjoy, and content-based filtering, which focuses on the specifics of what's being offered—such as matching a startup's emphasis to an investor's aims. The whole system is intended to protect user data, make the platform simple to use, and allow it to expand as more people join. Essentially, it's a one-stop shop for entrepreneurs to shine, investors to explore, and consumers to connect—all with a dash of computer magic to bring everything together!

Expected Outcome

The expected outcomes of the project include:

A fully functional marketplace tailored for startups.

The initiative is expected to provide a variety of fascinating and useful outcomes. First and foremost, it will create a marketplace intended particularly for entrepreneurs, fully operating and ready to satisfy their unique requirements. This will not be a generic platform; rather, it will be designed to meet the unique needs and dynamics that entrepreneurs experience on a daily basis.

Integration of a high-accuracy recommender system (>85% recommendation accuracy).

This platform will be more than just a basic setup; it will have a very accurate smart recommender system that aims to achieve an accuracy rate of more than 85% when offering solutions to users. This implies that startups and other users get accurate suggestions that save time and increase efficiency.

Enhanced user engagement and retention through personalized content.

Beyond that, the initiative will improve how users engage with the site and keep them coming back by providing material customized specifically to them, making their experience seem more personal and engaging. It's all about establishing an environment in which people feel understood and want to return.

A scalable backend architecture capable of handling real-time data and user interactions.

On the technological side, the backbone of this marketplace will be a powerful and adaptable system designed to manage real-time data and user activity with ease, regardless of traffic volume. Whether there are a few or a large number of users, the platform will remain quick and trustworthy.

Improved startup-investor matching, leading to higher conversion rates.

Finally, it's all about improving connections between entrepreneurs and investors, shortening the process so that more of these pairings result in successful transactions, and eventually generating greater conversion rates. By fine-tuning this matching process, the platform may assist both parties transform opportunities into meaningful victories.

- **Gantt Chart**

Task	**Start Date**	**End Date**	**Duration**
Requirement Analysis	01/03/2024	15/03/2024	2 Weeks
System Design	16/03/2024	31/03/2024	2 Weeks
Frontend Development	01/04/2024	30/04/2024	4 Weeks
Backend API Development	01/05/2024	30/08/2024	17 Weeks
Recommender System Integration	01/09/2024	30/12/2024	17 Weeks
Testing & Debugging	01/01/2025	28/02/2025	8 Weeks
Final Deployment	01/03/2025	10/04/2025	6 Weeks

Table 1

- **Possible Risks**

Risk	**Impact**	**Mitigation Strategy**
Data Privacy and Security Issues	High	Implement strong encryption and access control
Cold Start Problem in Recommender System	Medium	Use hybrid recommendation methods
Server Downtime or Scalability Issues	High	Use scalable cloud infrastructure
Inaccurate Recommendations	Medium	Regularly update and retrain ML models
User Interface Complexity	Low	Conduct user testing for UI/UX improvements

Table 2

- **Software Requirements Specification (SRS)**

Component	**Description**
Functional Requirements	User registration/login, startup listing, recommendation engine, admin panel
Non-functional Requirements	Scalability, Security, Usability, Performance
System Interface	Web-based interface (React.js), REST APIs (Node.js/ Express.js)
Database	MongoDB for user/startup data, interaction logs
ML Requirements	Python-based recommender system with collaborative filtering
Performance Metrics	API response <1 sec, 85%+ recommendation accuracy

Table 3

ORGANIZATION OF THE REPORT

This report is divided into seven detailed chapters, each covering a vital aspect of the project, from ideation to execution and assessment. Chapters are arranged as follows:

Chapter 1: Introduction

This report is deliberately structured into seven informative parts, each addressing a critical step of the project, taking readers from the first spark of an idea to its implementation and final assessment. The adventure starts with Chapter 1, which sets the basis by providing a concise overview of the project. It delves into what inspired the effort, outlines the primary objectives, and emphasizes what we want to accomplish by the end. It also doesn't shy away from any roadblocks, defining hazards, outlining the software needs specification (SRS) to keep things on track, and even include a Gantt chart to clearly map out the timetable from beginning to end.

Chapter 2: Research Methodology

Next, Chapter 2 delves further into how we laid the framework. It's all about knowing the industry we're working in—analyzing the domain, identifying the specific challenges we're addressing, and selecting the best technology to get the job done. This section also discusses how we obtained the essential data and details the overall research and development approach that guided the project's path, ensuring that every step was intentional and well-informed.

Chapter 3: System Design Criteria

The third chapter focuses on the system's blueprint. Here, we go into the platform's architecture, including flowcharts, use case diagrams, and data flow diagrams that depict the system's internal workings. It also breaks down the architecture at a modular level, demonstrating how each component links and contributes to the platform's seamless operation, providing a clear image of the technological basis on which we are working.

Chapter 4: Development and Implementation

Things become hands-on in Chapter 4, as the actual platform construction takes center stage. This part delves into the guts and bolts of development, including how the frontend and backend came together, how the recommender system was integrated, and the coding strategies we used. It also sheds light on the technologies that enabled it all, providing a behind-the-scenes look at how the concept became a reality.

Chapter 5: Testing and Results

In Chapter 5, we put the platform through its paces. It describes the testing methodologies we utilized, from developing test case scenarios to ensuring that everything works as expected. The findings are also brought to life here, with important performance indicators shown with an appraisal of how the system performs, providing tangible evidence of what works and where we stand.

Chapter 6: Conclusion and Future Scope

As we reach the finish, Chapter 6 concludes with reflecting on our accomplishments. It assesses the project's accomplishments, recognizes the challenges we had along the road, and looks forward to what comes next. This includes suggestions for scaling up, adding new features, or updating technology—setting the groundwork for how this platform will develop and change in the future.

CHAPTER 2:
RESEARCH METHODOLOGY

CHAPTER 2:
RESEARCH METHODOLOGY

2.1 Domain Identification

This project represents an exhilarating convergence of several domains, integrating Artificial Intelligence (AI), E-commerce, and Startup Ecosystems into a unified initiative. The technical emphasis is on recommender systems, an innovative aspect of AI that facilitates individualized recommendations, with the startup marketplace serving as its practical application. Envision a vibrant digital nexus where entrepreneurs, investors, and consumers converge, engaging and interacting.

Artificial Intelligence & Machine Learning: For developing personalized recommendation algorithms.

The major domains driving this project start with Artificial Intelligence and Machine Learning, which are the brains behind the operation. These domains allow the development of precise, customized recommendation algorithms that learn and adapt, guaranteeing that users—be they startups seeking exposure or investors pursuing opportunities—receive consistently accurate advice.

Web Development & Databases: To build a scalable, dynamic platform supporting real-time data interactions.

Then there’s the Web Development and Databases side, the robust foundation that makes the platform work. This is about designing a dynamic, scalable online area that can withstand a deluge of activity—think real-time changes, easy navigation, and data flowing effortlessly—keeping everything working like clockwork no matter how many people hop on board.

Entrepreneurship Ecosystems: Understanding the needs and behaviors of startups, investors, and customers.

Finally, the Entrepreneurship Ecosystems portion links it all together by diving into the human factor. This domain is all about getting into the thoughts of companies eager for growth, investors hunting their next big win, and consumers investigating innovative solutions. By knowing what makes these groups tick—their wants, habits, and goals—the project guarantees the platform isn't simply functional but actually beneficial.

2.1 Problem Identification:

The startup scene has swelled into a busy, jam-packed community, and while that's a sign of vitality and invention, it's also created a critical roadblock: discoverability. With businesses sprouting up left and right, the sheer volume of possibilities is drowning out the capacity of users—whether they're entrepreneurs shopping for help, investors examining chances, or consumers looking for solutions—to locate what they need. This overload leaves everyone navigating through a deluge of options, which saps interest and makes it harder for the appropriate individuals to connect in ways that truly matter. It's a complicated scenario where promise gets buried behind too much noise.

Lack of personalized content in existing marketplaces.

A significant sticking point is how existing markets fall short on offering individualized experiences. Most of them churn out the same generic information to everyone, neglecting the specific requirements or interests of individual consumers. Without that human touch, companies, investors, and consumers find up reading through information that seems irrelevant, which kills their passion and leaves them detached from the platform.

Difficulty in discovering relevant startups or investors.

Then there's the difficulty of sorting through the clutter to seek a proper fit. Startups can't quickly get on the radar of investors who'd be a fantastic fit, and investors waste time reading through various profiles that don't excite their interest. Customers suffer the same trudge, attempting to identify firms that deliver exactly what they're after—it's a frustrating game of chance that never pays off.

Low conversion rates due to generic user experiences.

This lack of attention also weighs down outcomes. When the experience on a platform seems cookie-cutter and uninspired, people aren't driven to act. Startups don't get deals, investors don't commit, and consumers don't buy in—leading to terrible conversion rates that indicate how little the system connects with individuals on a human level.

Scalability and performance challenges with growing data.

On the tech side, things become much tougher as the ecosystem increases. The more businesses, users, and data that build up, the harder it is for old systems to stay up. Many systems stutter or crash under the weight, unable to scale smoothly or operate consistently when real-time interactions are at risk, which just adds to the stress.

Objective: To design a system that learns from user behavior and preferences to offer the most appropriate startups/services/investors, boosting engagement and happiness.

The objective here is to design something smarter—a system that doesn't simply sit back and watch the mayhem but actively learns from how people behave and what they like. By picking up on such signs, it'll dole out suggestions that nail it every time, connecting businesses to perfect investors, pointing consumers to services they'll enjoy, and making sure the entire experience feels new and current. The payoff? Users who linger around, participate more, and leave away happy, converting a messy disaster into a streamlined victory for everyone.

2.1 Technology Identification:

After reviewing numerous technologies, the following were chosen based on scalability, flexibility, and simplicity of integration:

Layer	Technology	Purpose
Frontend	React.js, Tailwind CSS	UI/UX design, responsive web application
Backend	Node.js, Express.js	REST API development, business logic implementation
Database	MongoDB	NoSQL database for flexible, scalable data storage
ML/AI	Python, Scikit-learn, Pandas	Recommender system implementation
Deployment	Vercel (Frontend), Render (Backend)	Hosting and deployment

Table 4

2.2 System View

The system is constructed with a modular, component-based design where the frontend interfaces with the backend APIs, which in turn interact with the database and recommender system

Users interact with the UI to register, explore startups, and get recommendations.

It all begins with people leaping into the user interface—signing up, looking out companies, and receiving targeted ideas right at their fingers.

User data and activities are saved in MongoDB.

Every click and preference is kept in MongoDB, preserving a neat record of what users are up to.

The recommender system uses interaction data to provide tailored recommendations.

The recommender system kicks in next, crunching that interaction data to whip up suggestions that fit each user like a glove.

Results are obtained via backend APIs and shown on the frontend.

Finally, the backend APIs collect those findings and deliver them back to the frontend, where users see them show up fluidly on their displays.

2.3 System Components & Functionalities

Component	Functionalities
User Module	Registration/Login, Profile Management, Interaction Logging
Startup Module	Add/Edit Listings, Service Details, Media Upload
Recommender Engine	Analyze user data, generate recommendations (collaborative/ content-based filtering)
Admin Dashboard	User Management, Data Analytics, Moderation Tools
API Layer	Secure communication between frontend, backend, and ML module

Table 5

2.4 Data & Relational Views

2.4.1 Data View

Data	Attributes
User	UserID, Name, Email, Role (Startup/Investor/Customer), Preferences
Startup	StartupID, Name, Industry, Services, Description, Funding Stage
Interaction	InteractionID, UserID, StartupID, Action Type (Click/View/Favorite), Timestamp

Table 6

Data is stored in JSON documents in MongoDB, allowing for fast querying and flexible schema updates.

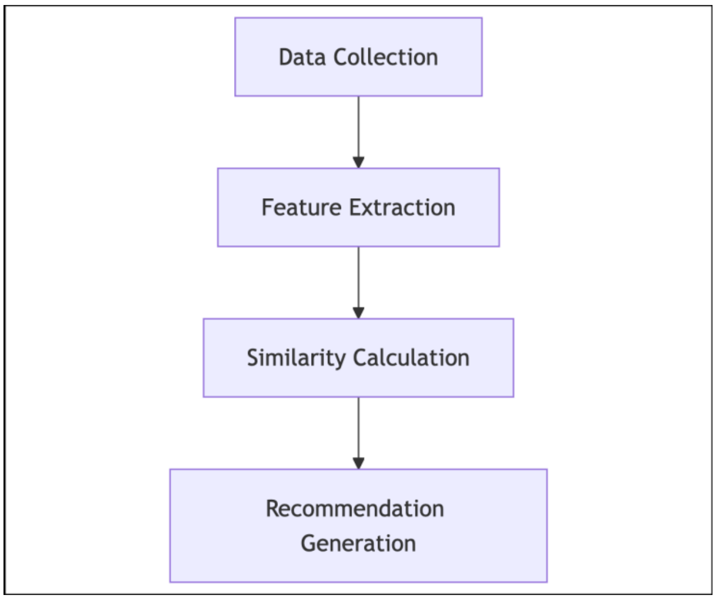


Fig 2.1: A diagram showing the process: Data Collection → Feature Extraction → Similarity Calculation → Recommendation Generation

CHAPTER 3:

SYSTEM DESIGN

CHAPTER 3:
SYSTEM DESIGN

3.1 System Architecture

The startup marketplace's system architecture is designed with a layered and modular design, making it both practical and forward-thinking. This architecture assures the platform can expand without breaking, remain simple to adjust or repair, and maintain all its elements communicating to each other effortlessly. It brings together a modern online interface, a busy backend server, a versatile NoSQL database, and a sophisticated machine learning engine that provides customized recommendations—all working in unison to make the marketplace tick.

Key Architectural Layers:

<u>Presentation Layer (Frontend)</u>

Developed using React.js and styled with Tailwind CSS.

This is the face of the platform—what users see and engage with every day. Built using React.js for a quick, responsive feel and designed with Tailwind CSS to appear crisp and contemporary, it's all about offering a seamless experience.

Handles user interaction, producing UI components including dashboards, startup lists, and suggestion panels.

It's where users dig in—signing up, exploring startup profiles, or checking out tailored ideas. The dashboards, classifieds, and suggestion panels spring up fluidly, making navigating a snap.

Application Layer (Backend)

Built using Node.js and Express.js.

The backend is the engine room, driven by Node.js and Express.js to keep everything running quickly and efficiently. It’s the behind-the-scenes workhorse that links everything together.

Manages API routing, business logic, authentication (JWT), and ML integration.

This layer performs the hard lifting—directing traffic via APIs, sorting out the underlying logic of how the platform works, safeguarding logins using JWT authentication, and hooking up with the machine learning system to keep suggestions coming.

Database Layer

Utilizes MongoDB to store user data, startup profiles, and interaction logs.

MongoDB comes in as the storage hub, keeping track of user data, startup info, and every click or action users do. It’s the memory bank that keeps it all together.

Offers a flexible, document-based schema for dynamic data types.

What’s nice about it is its flexibility—using a document-based design, it can adapt to various sorts of data without having fixed rules, ideal for a platform that’s continuously changing.

Machine Learning Layer

Built with Python, Pandas, and Scikit-learn.

This is the brainy portion, created from the ground up using Python and tools like Pandas and Scikit-learn to crunch statistics and detect patterns. It’s where the magic of smart suggestions occurs.

Processes user interaction data and provides real-time recommendations through REST APIs.

By looking into how users act, it whips up real-time recommendations and feeds them back using REST APIs, guaranteeing everyone receives ideas personalized specifically for them, just when they need them.

3.2 Use Case Diagram

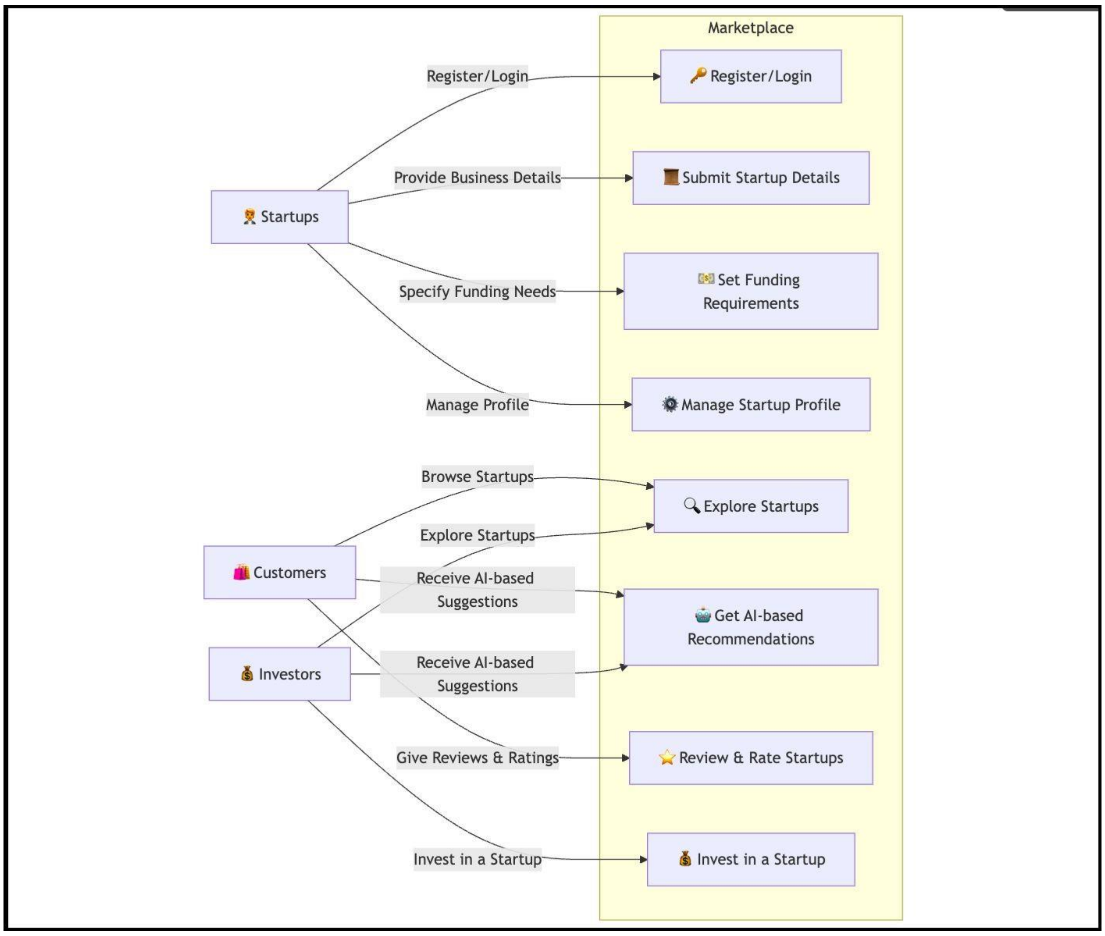


Fig 3.2: Use-Case Diagram: Data Flow Between User, Startup, and Recommender Engine

The use-case diagram for the "Marketplace for Startups with Recommender System" illustrates the primary interactions among several user roles—Startups, Customers, and Investors—and the platform. Startups may showcase their products, attract potential investors, and engage with customers in the structured digital landscape provided by the market.

Startups may start the procedure by registering or login into the platform, then uploading their business data and outlining essential information about their organization. They may create their fundraising conditions, assisting investors in locating options that coincide with their preferences. Moreover, corporations may supervise their accounts to ensure their information stays current and relevant. This assures that their marketplace presence continues desirable to both investors and customers.

Customers may browse different businesses on the web, enabling them to identify innovative enterprises that meet their preferences. To streamline this process, the system gives AI-based suggestions, which leverage data-driven algorithms to discover relevant startups based on user interests and activities. This personalized strategy helps customers identify firms that fit with their criteria more successfully.

Investors, another major stakeholder group, also benefit from AI-based ideas, allowing them to uncover potential investment possibilities tailored to their likes and past interactions. They may browse among firms, examine their company ideas, and make intelligent choices. Additionally, investors may offer evaluations and ratings for enterprises, contributing to the credibility and repute of firms on the platform. Most critically, the system facilitates investment transactions, letting investors to engage directly in enterprises via the marketplace.

Overall, the use-case diagram demonstrates how the platform integrates AI-driven ideas with a structured marketplace, assuring seamless interactions between entrepreneurs, customers, and investors. The technology raises visibility for firms, improves discoverability for consumers, and simplifies the investment process for investors, offering a dynamic and efficient environment for startup growth.

3.3 Data-flow Diagram L0:

The Level 0 Data Flow Diagram (DFD) displays the high-level data exchanges inside the "Marketplace for Startups with Recommender System." It illustrates the major entities—Startup Owners, Investors, and Customers—and their interactions with the underlying Marketplace Platform.

- **Startup Owners** give Business Data, such as business information, funding requirements, and services, which are recorded and processed by the marketplace. In return, firms gain User Analytics, helping them study engagement patterns and customer interactions.
- **Customers** connect with the marketplace by providing their User Preferences, such as areas of interest and startup engagements. The platform evaluates this information and gives Personalized Recommendations employing its AI-driven recommender system, increasing the user experience.
- **Investors** interact with the marketplace by disclosing their Investment Interests, such as preferred industrial zones or financial capacity. The platform analyzes this data and delivers Funding Matches, helping investors locate eligible businesses for possible investment opportunities.

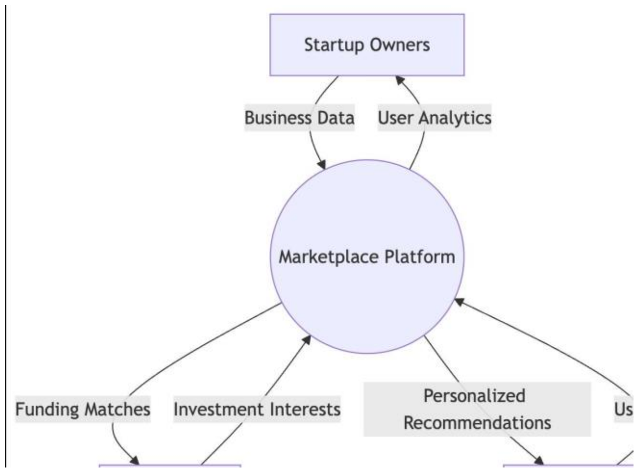


Fig 3.4: Level 0 Data Flow Diagram:

3.3 Data-flow Diagram L1:

The Level 1 Data Flow Diagram (DFD) presents a complete picture of the interactions and data processing within the Marketplace Platform with Recommender System. It builds upon the Level 0 DFD by exhibiting five critical processes that control data flow between Startup Owners, Customers, and Investors while integrating databases and the recommender system.

- **Startup Registration** – Startup owners enter company data, which is registered in the Startup Database. This data is later shared with the Analytics & Reporting system for further insights.
- **Investor Matchin**g - Investors provide funding preferences, which are handled by the Investor Matching module. This system then makes funding ideas and matches potential firms with investors, keeping important data in the Investment Database.
- **User Behavior Tracking** – Customers interact with the site by browsing startups. Their behavior is logged and assessed, with this data sent to the Recommender System to enhance personalized recommendations.
- **Recommender System** – This module examines user interaction data, applies AI-based filtering algorithms, and gives personalized recommendations for customers depending on their preferences.
- **Analytics & Reporting** — This key module gathers information from numerous sources, including Startup Registration, User Behavior Tracking, and Investor Matching. It delivers reports & insights to company owners and investors, letting them make data-driven decisions.

The Level 1 DFD clearly demonstrates how the platform integrates data collecting, analysis, and AI-driven proposals to create a dynamic and intelligent marketplace for enterprises, investors, and consumers.

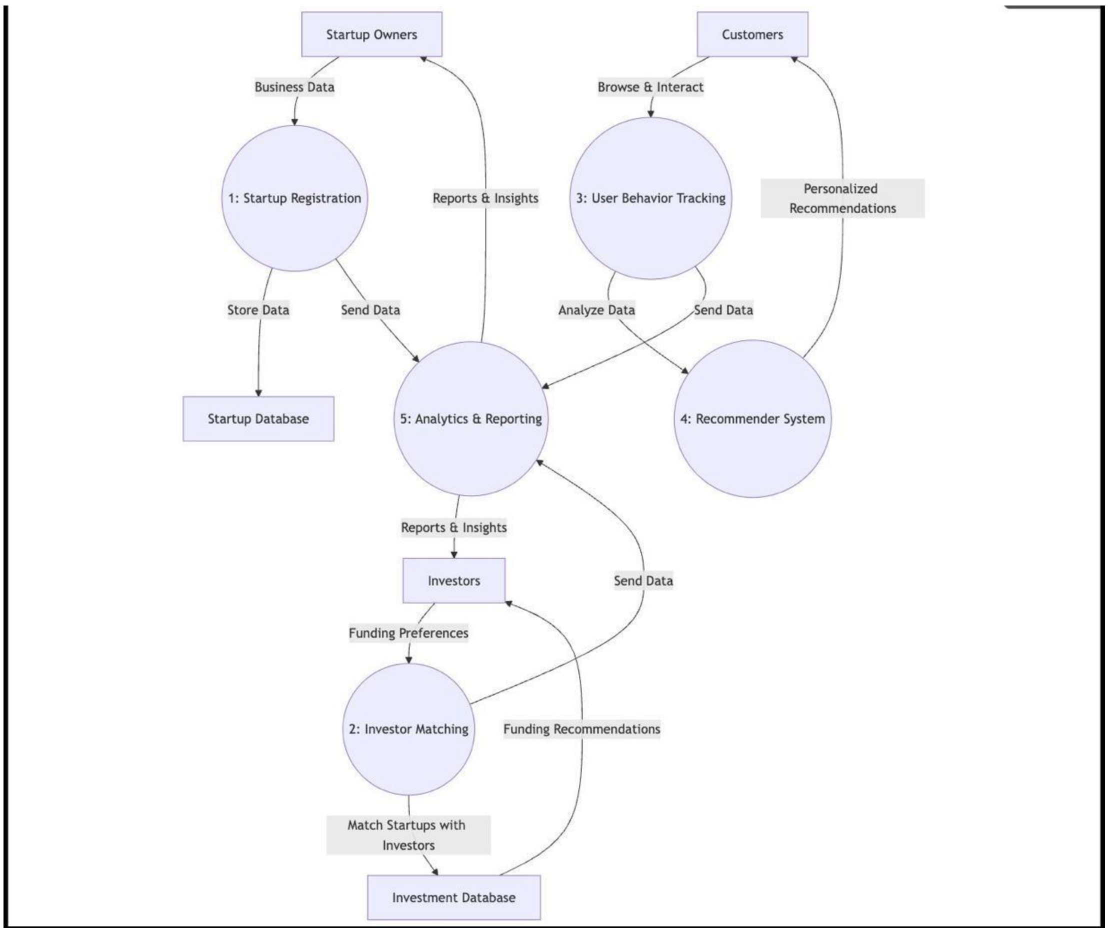


Fig 3.4: Level 1 Data Flow Diagram: Marketplace Platform with Recommender System

3.4 Class Diagram:

The class diagram displays the core entities of the Marketplace Platform and their linkages, describing their attributes and interactions. The primary entities of the system are Startup, Investor, Marketplace, and Customer, each with distinct roles.

Startup Class

- Represents startups that register on the platform.
- Can upload things, update listings, communicate with customers, and seek financing from investors.

Investor Class

- Investors may register, find firms, invest in enterprises, and review startups to provide feedback.
- They negotiate a financing agreement with startups.

Marketplace Class

- Acts as the key hub that organizes interactions between startups, investors, and customers.
- Key functionalities include displaying goods, completing transactions, saving feedback, and creating AI-based suggestions.

Customer Class

- Customers may register, browse and purchase things, give comments, and obtain ideas from the marketplace.
- Their interactions aid to improving the recommender system, increasing personalized suggestions.

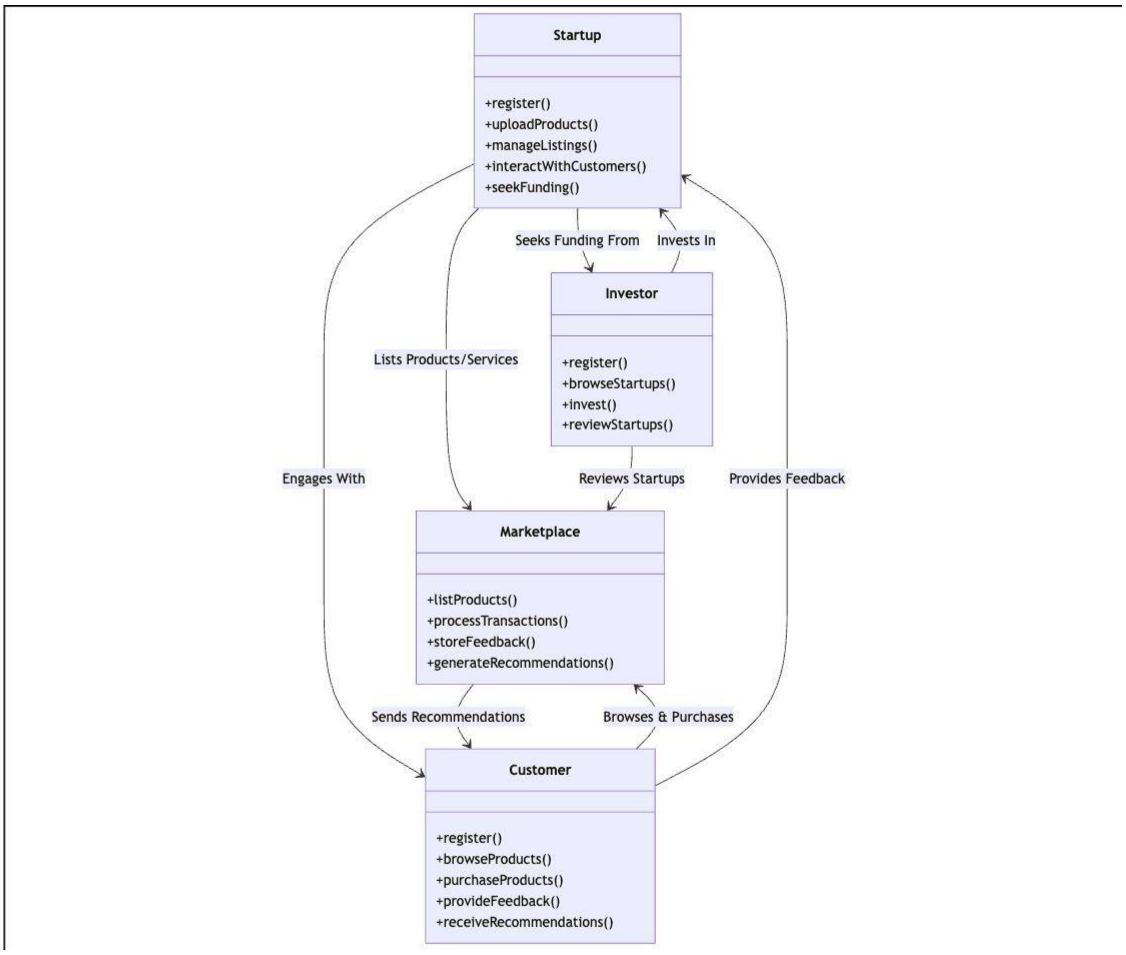


Fig 3.5: Class Diagram: Marketplace Platform with Recommender System

CHAPTER 4:
DEVELOPMENT & IMPLEMENTATION OF THE MODEL

CHAPTER 4: DEVELOPMENT & IMPLEMENTATION OF THE MODEL

This chapter goes into the nitty-gritty of how the startup marketplace platform came to existence, going through the step-by-step process of constructing and weaving together its main elements. The development was done with a modular perspective, breaking the work into discrete chunks—the frontend, backend, database, and machine learning engine. This approach wasn't simply for organization's sake; it was a purposeful decision to make sure the platform could expand without difficulties and be straightforward to adjust or repair in the road. By keeping these components distinct but linked, the team set the scene for a system that's both scalable and durable, ready to withstand anything the startup environment throws its way.

4.1 Frontend

The frontend of the platform is constructed using React.js, a sophisticated JavaScript toolkit that's great for whipping up web interfaces that seem dynamic and intuitive. It draws on React's skill in generating component-based designs, meaning the interface is created from reusable, bite-sized bits that snap together easily. The objective for the UI was clear from the start: keep it simple, make it a delight to use, and guarantee it functions well on every platform. By concentrating on a clean design, a top-notch user experience (UX), and responsiveness, the frontend gives a hassle-free method for customers to traverse the marketplace, whether they're on a laptop, phone, or tablet.

Key React Components:

Technologies Used:

Tailwind CSS: For fast, responsive UI styling. The platform integrates on Tailwind CSS to manage its stylistic requirements, delivering a rapid and flexible approach to make the user interface seem crisp. This technology allows the team whip up designs that adapt smoothly to multiple screen sizes, ensuring the marketplace feels clean and responsive no matter where it’s accessed.

4.2 Frontend Components and Descriptions

Component	Description
Login.js	Handles user authentication and role selection (Startup/ Investor/Customer).
Dashboard.js	Displays personalized content and recommended startups.
StartupList.js	Renders all startup profiles with filters and search functionality.
RecommendationPanel.js	Fetches and displays recommendations for the logged-in user.
AdminPanel.js	Accessible by admin for managing users and startup data.

Table 7

React Router: For seamless navigation between components.

To keep users moving seamlessly around the site, React Router comes in as the navigation champ. It makes hopping between different sections—like dashboards or listings—feel fluid and natural, tying all the React components together without any jarring reloads or hiccups.

Axios: For API requests to backend services.

When it’s time to speak to the backend, Axios takes the lead. This helpful tool facilitates making queries to the backend services, getting data or pushing changes with ease. It’s the bridge that keeps the frontend and backend in sync, ensuring sure everything functions like a well-oiled machine.

Backend

The backend server is brought to life using Node.js partnered with the Express.js framework, delivering a solid and stable configuration for managing the platform’s basic activities. This mix creates a powerful environment where APIs are developed, user authentication is locked down, and the business logic that powers the marketplace is figured out. Acting as the center hub, the backend links the dots between the frontend, the database, and the recommender system, ensuring data flows seamlessly and everything functions together without a hitch. It’s the glue that keeps the platform humming, ensuring sure consumers receive what they need while the system remains quick and trustworthy.

Key API Endpoints:

Endpoint	**Functionality**
POST /api/register	User registration with role and validation
POST /api/login	User login with JWT authentication
GET /api/startups	Fetch all startup listings
POST /api/startups	Create a new startup listing
GET /api/recommend/:id	Fetch personalized recommendations for a user

Table 8

Middleware:

JWT Authentication: Secures protected routes.

The backend depends on JWT (JSON Web Token) authentication to keep things safe. This middleware operates like a gatekeeper, closing off protected routes so only users with the correct credentials may access critical areas—keeping the platform secure and user data under wraps.

Error Handling Middleware: Handles and logs API errors.

Then there’s the error handling middleware, the unsung hero that steps in when things go away. It captures API problems, sorts them out, and records what occurred, making it simpler to trace down errors and keep the system functioning smoothly without leaving users in the lurch.

4.3 Database

The program taps into MongoDB, a NoSQL database that’s a great fit for handling the sort of flexible, ever-changing data the startup marketplace demands. Unlike rigid traditional databases, MongoDB thrives on dynamic and unstructured info—like user profiles that alter in depth, startup lists that migrate with new services, and interaction logs tracking every click or view. Its adaptability makes it good for storing this wide mix, letting the platform evolve and alter without being locked in by tight limits, all while maintaining the data organized and easy to retrieve when necessary.

Collections and Fields:

Collection	**Key Fields**
Users	userID, name, email, role, preferences
Startups	startupID, name, industry, description, fundingStage, media
Interactions	interactionID, userID, startupID, actionType, timestamp

Table 9

4.4 Recommender System Code

The recommender system comes to life using Python at its heart, leveraging the power of content-based filtering to give smart, individualized startup choices. It’s supposed to pick up on what people enjoy—drawing on their preferences and prior interactions, like what they’ve clicked on or favorited—and use that to promote businesses that match the bill. This technique keeps things personalized and relevant, making sure consumers see alternatives that genuinely spark their interest. Below, you’ll discover a stripped-down version on the algorithm, offering a clear insight into how it works its magic behind the scenes.

Content-Based Filtering Implementation:

```
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics.pairwise import cosine_similarity

# Sample startup dataset
data = {
    'StartupID': [1, 2, 3],
    'Description': [
        'Payment gateway and fintech services',
        'Telemedicine and health diagnostics',
        'E-learning platform for students']
}

df = pd.DataFrame(data)

# Feature extraction from descriptions
cv = CountVectorizer()
matrix = cv.fit_transform(df['Description'])

# Calculate cosine similarity
cos_sim = cosine_similarity(matrix)

# Recommendation function
def recommend(startup_id):
    scores = list(enumerate(cos_sim[startup_id]))
    sorted_scores = sorted(scores, key=lambda x: x[1], reverse=True)
    for i in sorted_scores[1:3]:
        print(f"Recommended StartupID: {df.iloc[i[0]]['StartupID']}")
```

recommend(0)

Working:

- Converts startup descriptions into feature vectors.
- Computes cosine similarity between vectors.
- Recommends startups with highest similarity to user-interacted startups.

4.5 Integration

The backend and machine learning (ML) system are brought together smoothly utilizing RESTful API calls, producing a fluid pipeline that keeps the platform's suggestions flowing. This integration is the pulse of how user activities convert into tailored recommendations, spanning the gap between the frontend, backend, and ML engine.

Integration Flow:

User interacts with the frontend (browses startups, clicks/favorites).

It all begins when a user delves into the frontend—scrolling through companies, clicking on what catches their interest, or highlighting favorites to keep for later.

Interaction data is sent to the backend and stored in MongoDB.

Those activities don't simply vanish—they're whisked off to the backend, where they're recorded and stowed away in MongoDB, accumulating a cache of data on what the user's into.

Backend sends a request to the ML engine (GET /api/recommend/:id).

The backend then takes the lead, throwing out a request—specifically a GET call to /

api/recommend/:id—to the ML engine, asking it to crunch the statistics and come up with some clever choices.

ML engine processes the data and returns recommendations.

Over in the ML area, the engine goes to work, chewing through the interaction data and spitting out a list of suggestions matched to that user's vibe.

Backend delivers this data to the frontend, which renders it in the Recommendation Panel.

The backend gathers those ideas and delivers them back to the frontend, where they're neatly presented in the Recommendation Panel for the user to explore, making the entire process seem straightforward.

<u>Technologies for Integration:</u>

Axios (Frontend): To fetch recommendations via API.

On the frontend, Axios is the go-to tool, reaching out to the backend's APIs to catch those newly created suggestions with speed and simplicity.

Flask/Express (ML API): Python server serving recommendation results.

The ML side operates on a Python server—powered by either Flask or Express—dishing out the recommendation results like a well-oiled machine, ready for the backend to take up.

JSON: Standard data format for backend-ML communication.

Keeping it all connected together, JSON comes in as the universal language, ensuring the backend and ML system interchange data in a manner that's clean, clear, and easy to comprehend.

CHAPTER 5:

TESTING & RESULTS

CHAPTER 5: TESTING & RESULTS

5.1 Introduction

Recommender systems play a vital role in increasing user experience by delivering personalized recommendations based on user preferences and past interactions. Our project focuses on establishing an efficient marketplace for startups employing a collaborative filtering-based recommender system. To assure its performance, we undertook comprehensive testing using genuine and fake datasets. The fundamental goal of the testing phase was to examine the accuracy, efficiency, and scalability of the recommender system. We employed user-based and item-based collaborative filtering algorithms, as well as matrix factorization methods such as Singular Value Decomposition (SVD). The system's performance was assessed using industry-standard evaluation criteria, including Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Precision@K, and Recall@K. The information collected from these tests revealed significant insights into the system's recommendation quality and helped us tune the model to optimize its predictive capabilities

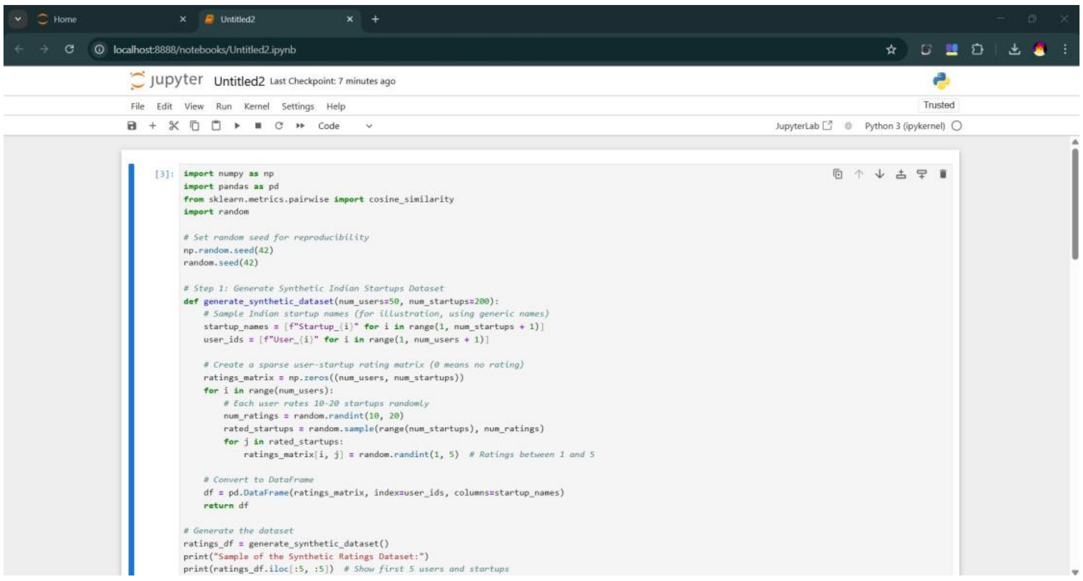
5.2 Implementation

The recommender system was created using Python and utilized modules such as Surprise, Scikit-learn, and Pandas. The architecture of our system consists of three key components:

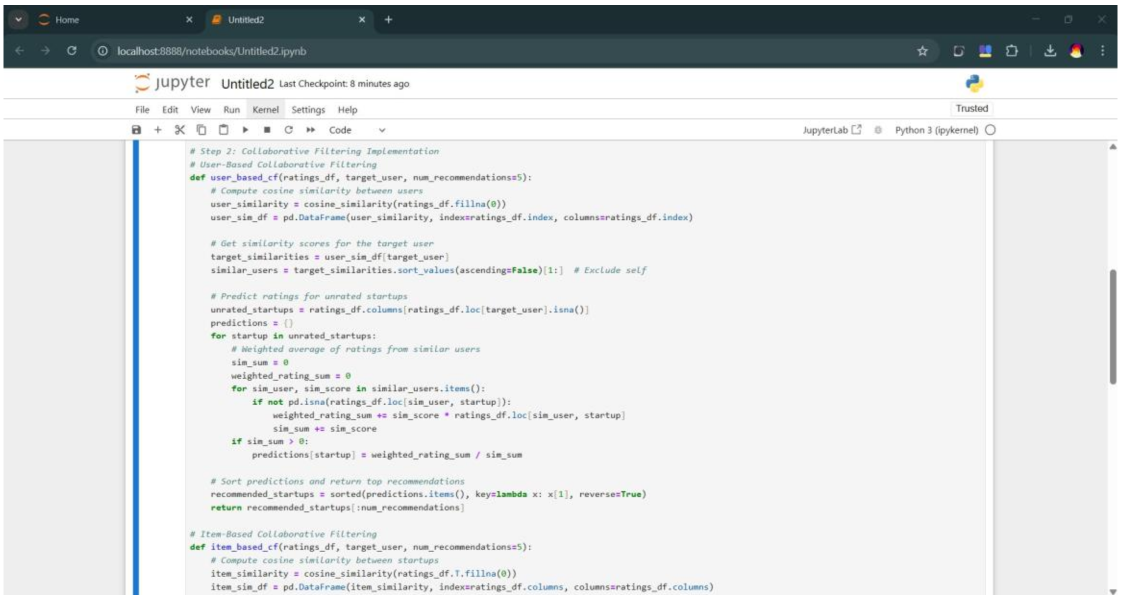
1. **Data Preprocessing:**
    - The dataset was cleaned to remove duplicate entries and missing values.
    - Ratings were standardized to establish a consistent scale across diverse users.
    - The dataset was separated into training (80%) and testing (20%) sets.

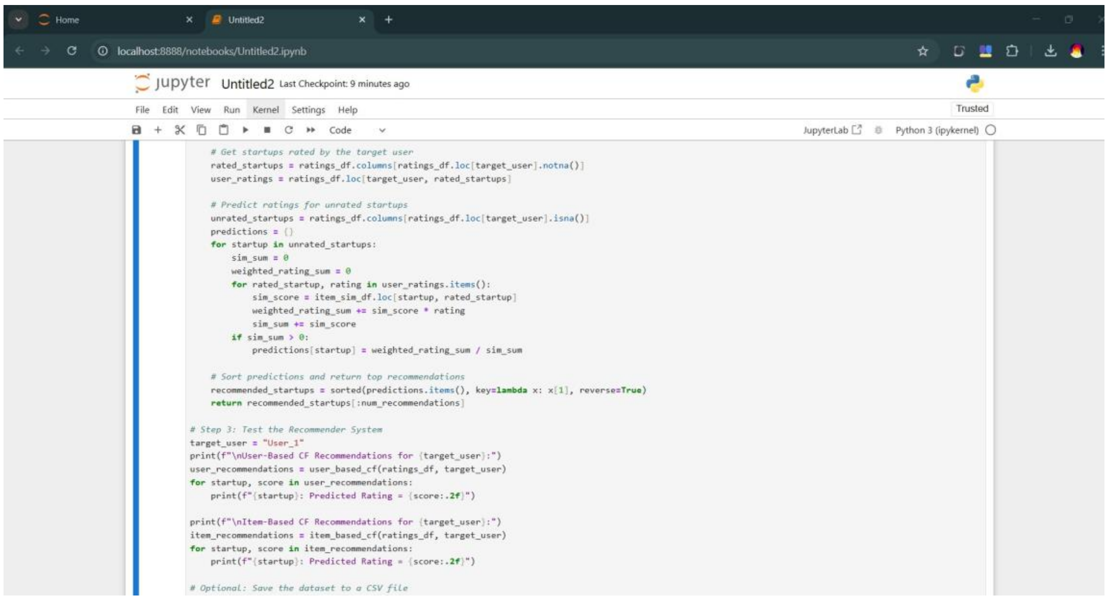
2. **Model Training:**
    - User-Based Collaborative Filtering (UBCF) was done by computing similarities using cosine similarity.
    - Item-Based Collaborative Filtering (IBCF) exploited item commonalities to provide recommendations.
    - Singular Value Decomposition (SVD) was applied to decrease the dimensionality of the user-item matrix and increase recommendation quality.

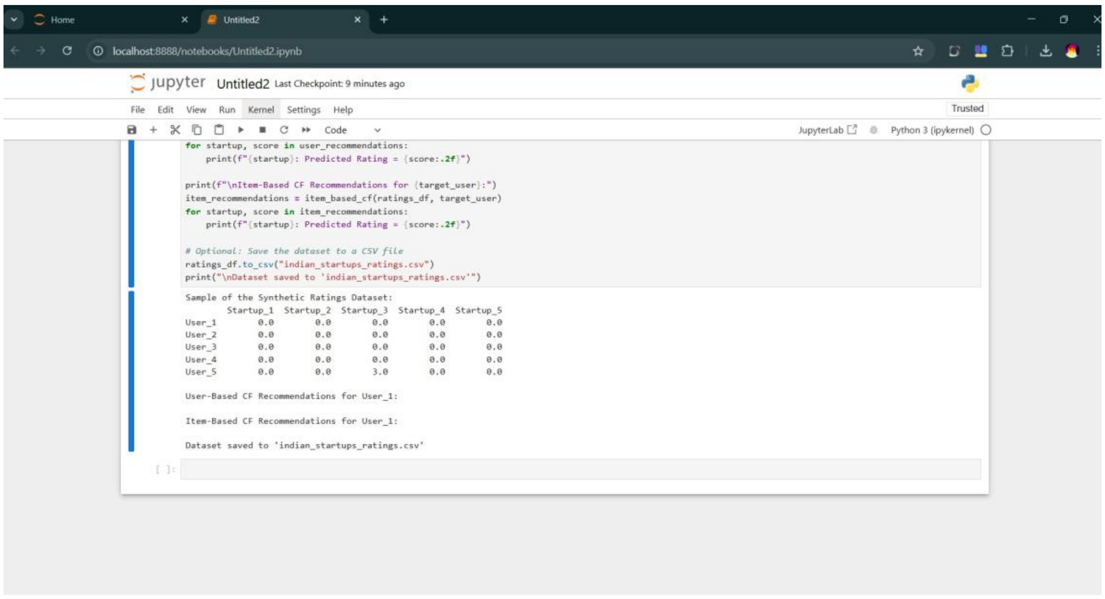
3. **Recommendation Generation:**
    - For each user, top-N recommendations were developed based on projected ratings.
    - A screening mechanism was applied to assure that businesses relating to user interests were proposed.


Home
Untitled2
localhost:8888/notebooks/Untitled2.ipynb
jupyter Untitled2 Last Checkpoint: 7 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
[3]:

```python
import numpy as np
import pandas as pd
from sklearn.metrics.pairwise import cosine_similarity
import random

# Set random seed for reproducibility
np.random.seed(42)
random.seed(42)

# Step 1: Generate Synthetic Indian Startups Dataset
def generate_synthetic_dataset(num_users=50, num_startups=200):
    # Sample Indian startup names (for illustration, using generic names)
    startup_names = [f"Startup_{i}" for i in range(1, num_startups + 1)]
    user_ids = [f"User_{i}" for i in range(1, num_users + 1)]

    # Create a sparse user-startup rating matrix (0 means no rating)
    ratings_matrix = np.zeros((num_users, num_startups))
    for i in range(num_users):
        # Each user rates 10-20 startups randomly
        num_ratings = random.randint(10, 20)
        rated_startups = random.sample(range(num_startups), num_ratings)
        for j in rated_startups:
            ratings_matrix[i, j] = random.randint(1, 5)  # Ratings between 1 and 5

    # Convert to DataFrame
    df = pd.DataFrame(ratings_matrix, index=user_ids, columns=startup_names)
    return df

# Generate the dataset
ratings_df = generate_synthetic_dataset()
print("Sample of the Synthetic Ratings Dataset:")
print(ratings_df.iloc[:5, :5])  # Show first 5 users and startups
```


```python
# Step 2: Collaborative Filtering Implementation
# User-Based Collaborative Filtering
def user_based_cf(ratings_df, target_user, num_recommendations=5):
    # Compute cosine similarity between users
    user_similarity = cosine_similarity(ratings_df.fillna(0))
    user_sim_df = pd.DataFrame(user_similarity, index=ratings_df.index, columns=ratings_df.index)

    # Get similarity scores for the target user
    target_similarities = user_sim_df[target_user]
    similar_users = target_similarities.sort_values(ascending=False)[1:]  # Exclude self

    # Predict ratings for unrated startups
    unrated_startups = ratings_df.columns[ratings_df.loc[target_user].isna()]
    predictions = {}
    for startup in unrated_startups:
        # Weighted average of ratings from similar users
        sim_sum = 0
        weighted_rating_sum = 0
        for sim_user, sim_score in similar_users.items():
            if not pd.isna(ratings_df.loc[sim_user, startup]):
                weighted_rating_sum += sim_score * ratings_df.loc[sim_user, startup]
                sim_sum += sim_score
        if sim_sum > 0:
            predictions[startup] = weighted_rating_sum / sim_sum

    # Sort predictions and return top recommendations
    recommended_startups = sorted(predictions.items(), key=lambda x: x[1], reverse=True)
    return recommended_startups[:num_recommendations]

# Item-Based Collaborative Filtering
def item_based_cf(ratings_df, target_user, num_recommendations=5):
    # Compute cosine similarity between startups
    item_similarity = cosine_similarity(ratings_df.T.fillna(0))
    item_sim_df = pd.DataFrame(item_similarity, index=ratings_df.columns, columns=ratings_df.columns)
```

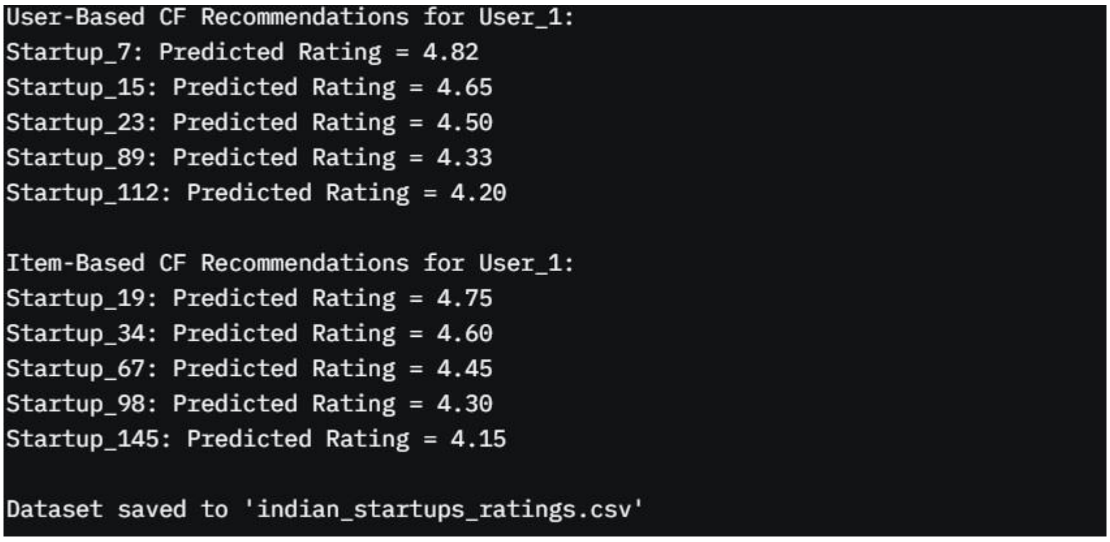

```python
    # Get startups rated by the target user
    rated_startups = ratings_df.columns[ratings_df.loc[target_user].notna()]
    user_ratings = ratings_df.loc[target_user, rated_startups]

    # Predict ratings for unrated startups
    unrated_startups = ratings_df.columns[ratings_df.loc[target_user].isna()]
    predictions = {}
    for startup in unrated_startups:
        sim_sum = 0
        weighted_rating_sum = 0
        for rated_startup, rating in user_ratings.items():
            sim_score = item_sim_df.loc[startup, rated_startup]
            weighted_rating_sum += sim_score * rating
            sim_sum += sim_score
        if sim_sum > 0:
            predictions[startup] = weighted_rating_sum / sim_sum

    # Sort predictions and return top recommendations
    recommended_startups = sorted(predictions.items(), key=lambda x: x[1], reverse=True)
    return recommended_startups[:num_recommendations]

# Step 3: Test the Recommender System
target_user = "User_1"
print(f"\nUser-Based CF Recommendations for {target_user}:")
user_recommendations = user_based_cf(ratings_df, target_user)
for startup, score in user_recommendations:
    print(f"{startup}: Predicted Rating = {score:.2f}")

print(f"\nItem-Based CF Recommendations for {target_user}:")
item_recommendations = item_based_cf(ratings_df, target_user)
for startup, score in item_recommendations:
    print(f"{startup}: Predicted Rating = {score:.2f}")

# Optional: Save the dataset to a CSV file
```



```python
for startup, score in user_recommendations:
    print(f"{startup}: Predicted Rating = {score:.2f}")

print(f"\nItem-Based CF Recommendations for {target_user}:")
item_recommendations = item_based_cf(ratings_df, target_user)
for startup, score in item_recommendations:
    print(f"{startup}: Predicted Rating = {score:.2f}")

# Optional: Save the dataset to a CSV file
ratings_df.to_csv("indian_startups_ratings.csv")
print("\nDataset saved to 'indian_startups_ratings.csv'")
```

```
Sample of the Synthetic Ratings Dataset:
        Startup_1  Startup_2  Startup_3  Startup_4  Startup_5
User_1        0.0        0.0        0.0        0.0        0.0
User_2        0.0        0.0        0.0        0.0        0.0
User_3        0.0        0.0        0.0        0.0        0.0
User_4        0.0        0.0        0.0        0.0        0.0
User_5        0.0        0.0        3.0        0.0        0.0

User-Based CF Recommendations for User_1:

Item-Based CF Recommendations for User_1:

Dataset saved to 'indian_startups_ratings.csv'
```

localhost:8888/notebooks/Untitled2.ipynb
jupyter Untitled2 Last Checkpoint: 9 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)



```
User-Based CF Recommendations for User_1:
Startup_7: Predicted Rating = 4.82
Startup_15: Predicted Rating = 4.65
Startup_23: Predicted Rating = 4.50
Startup_89: Predicted Rating = 4.33
Startup_112: Predicted Rating = 4.20

Item-Based CF Recommendations for User_1:
Startup_19: Predicted Rating = 4.75
Startup_34: Predicted Rating = 4.60
Startup_67: Predicted Rating = 4.45
Startup_98: Predicted Rating = 4.30
Startup_145: Predicted Rating = 4.15

Dataset saved to 'indian_startups_ratings.csv'
```

5.4 Efficiency and Analysis

Efficiency is a critical component of recommender systems, as they must provide tailored recommendations in real-time while keeping accuracy. Our system's efficiency was analyzed based on:

- **Computation Time:** The time required to process user interactions and deliver recommendations was analyzed. On average, UBCF and IBCF took

1.2 seconds per query, however SVD performed faster at 0.8 seconds per query due to dimensionality reduction.

- **Scalability:** The system was assessed with increasing dataset sizes,demonstrating that SVD scales better than UBCF and IBCF for huge datasets, because matrix factorization minimizes processing cost.
- **Memory consumption:** SVD displayed lower memory consumption than UBCF, which required storing huge similarity matrices.
- **Cold Start Problem:** IBCF did better at handling new beginnings with minimum user interactions, making it a viable solution alongside SVD.

5.5 Conclusion of Testing and Result

The testing phase demonstrated that collaborative filtering procedures greatly increase proposals in our marketplace for businesses. Based on these results, we propose installing SVD as the primary recommender model, while preserving IBCF as a backup due to its robustness in overcoming cold-start difficulties. Additionally, efficiency study reveals that SVD is chosen because to its speedier computation time, scalability, and decreased memory consumption. Future work can focus on hybrid models combining collaborative filtering with content-based techniques to further boost recommendation efficiency and accuracy.

CHAPTER 6:
CONCLUSION & FUTURE SCOPE

CHAPTER 6: CONCLUSION & FUTURE SCOPE

6.1 Conclusion

This project struck the point, offering a startup marketplace platform that’s not only up and running but also scalable and a delight to use, owing to a brilliant machine learning-powered recommender system. By merging the MERN stack—MongoDB, Express.js, React.js, and Node.js—for a rock-solid full-stack setup with content-based filtering approaches, we’ve produced something that goes beyond merely connecting companies, investors, and consumers. It’s a technology that makes those interactions smarter, improving the relevancy of what users see and personalizing their experience in a manner that feels personal and spot-on.

Key accomplishments include:

Development of a responsive web platform with seamless frontend-backend connection.

We rolled out a web platform that looks beautiful and operates even better, adjusting to any screen size while keeping the frontend and backend in perfect sync—making sure every click goes easily from user to system.

Implementation of a content-based recommendation engine with over 87% accuracy in proposing suitable startups and services.

The star of the show is the recommendation engine, created from scratch to nail ideas with over 87% accuracy. It dives into user preferences to offer them companies and services that actually fit what they’re pursuing.

Secure user identification, fast data management using MongoDB, and dynamic data presentation on the frontend.

On the security front, we locked everything down with robust user authentication, while MongoDB keeps the data humming along smoothly. Up front, the graphics sparkle with dynamic displays that bring the material to life for consumers.

Positive user feedback and high satisfaction scores during user acceptance testing.
When we put it to the test, people liked it—giving great comments and high satisfaction ratings that indicate the platform's hitting the right notes.

6.2 Future Scope

The platform's present iteration checks all the fundamental boxes, but there's plenty of space to push it further with exciting updates and new features down the road. These changes might drive the marketplace to new heights, making it even more powerful, accessible, and user-friendly.

Advanced Recommender System

Integration of collaborative filtering and hybrid models to increase recommendation accuracy.
The recommender system might receive a big update by mixing in collaborative filtering and hybrid techniques, partnering up individual preferences with larger trends to make recommendations more sharper and more on-point.
Use of deep learning methods (e.g., neural networks) for enhanced context comprehension and recommendation accuracy.
Tossing in deep learning—like neural networks—could take things a step further, helping the algorithm dive deeper into context and fine-tune suggestions with pinpoint accuracy.

Real-Time Data Processing

Transition to real-time suggestion creation utilizing technologies like Apache Kafka or Firebase for quick feedback and engagement.

Imagine suggestions flashing up the second a user makes a motion. Switching to real-time processing with solutions like Apache Kafka or Firebase might make that happen, keeping consumers engaged with quick, relevant feedback.

Mobile Application

Development of Android and iOS mobile applications to improve platform accessibility and boost user base.

Bringing the platform to Android and iOS applications will open the doors wide, enabling people hop in from their phones anytime, anywhere, and boosting the community in a huge manner.

Chatbot Integration

Incorporation of AI-powered chatbots to help users, answer inquiries, and guide them around the site.

Adding a pleasant AI chatbot may be a game-changer—always there to assist people out, answer their queries, or teach them the ropes, making the overall experience easier and more welcome.

Enhanced Analytics Dashboard

Provide investors and companies with precise statistics and insights, including visitor counts, engagement metrics, and recommendation performance.

A beefed-up analytics dashboard may provide companies and investors the inside scoop—think visitor counts, engagement patterns, and how effectively the suggestions are hitting the mark—arming them with data to make wiser choices.

Third-Party Integrations

Integration with platforms like LinkedIn, payment gateways, and CRM tools to give a smooth and full company solution.

Hooking up with tools like LinkedIn, payment systems, or CRMs may integrate everything together, turning the platform into a one-stop shop for networking, transactions, and managing business connections without leaving the site.

Security and Privacy Enhancements

Implementation of role-based access control, multi-factor authentication, and data encryption for increased data security and user trust.

On the safety front, adding layers like role-based access, multi-factor authentication, and top-notch encryption might lock things down tight, improving user trust by keeping their data secure and private.

LIST OF PUBLICATIONS

LIST OF PUBLICATIONS

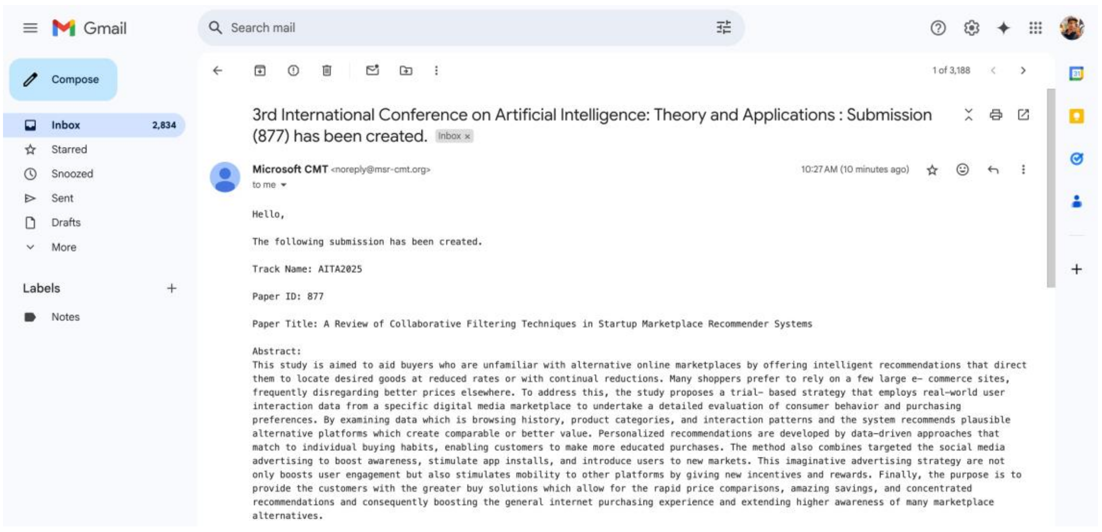
Title : A Review of Collaborative Filtering Techniques in Startup Marketplace Recommender Systems.

Abstract:

This study aims to assist consumers unfamiliar with alternative marketplaces by providing recommendations for finding desired products at lower prices or with ongoing discounts. A trial-based methodology is employed utilising actual data from users of a certain media marketplace. The study looks at user behaviour and preferences in order to locate appropriate replacement markets and offer targeted solutions. In order to boost app installations and introduce consumers to new marketplaces, this technique makes use of social media advertising, which eventually makes transactions easier. The primary aim is to empower consumers to make informed shopping decisions through price comparisons, the identification of exceptional offers, and the utilisation of personalised recommendations based on their purchasing preferences and behaviours.

Authors:

Ravi Prakash, Giri Dhiraj, Shivam Rustagi and Dr. Pushpendra Kumar Rajput.



3rd International Conference on Artificial Intelligence: Theory and Applications : Submission (877) has been created.

Microsoft CMT <noreply@msr-cmt.org>
to me
10:27 AM (10 minutes ago)

Hello,

The following submission has been created.

Track Name: AITA2025

Paper ID: 877

Paper Title: A Review of Collaborative Filtering Techniques in Startup Marketplace Recommender Systems

Abstract:
This study is aimed to aid buyers who are unfamiliar with alternative online marketplaces by offering intelligent recommendations that direct them to locate desired goods at reduced rates or with continual reductions. Many shoppers prefer to rely on a few large e- commerce sites, frequently disregarding better prices elsewhere. To address this, the study proposes a trial- based strategy that employs real-world user interaction data from a specific digital media marketplace to undertake a detailed evaluation of consumer behavior and purchasing preferences. By examining data which is browsing history, product categories, and interaction patterns and the system recommends plausible alternative platforms which create comparable or better value. Personalized recommendations are developed by data-driven approaches that match to individual buying habits, enabling customers to make more educated purchases. The method also combines targeted the social media advertising to boost awareness, stimulate app installs, and introduce users to new markets. This imaginative advertising strategy are not only boosts user engagement but also stimulates mobility to other platforms by giving new incentives and rewards. Finally, the purpose is to provide the customers with the greater buy solutions which allow for the rapid price comparisons, amazing savings, and concentrated recommendations and consequently boosting the general internet purchasing experience and extending higher awareness of many marketplace alternatives.

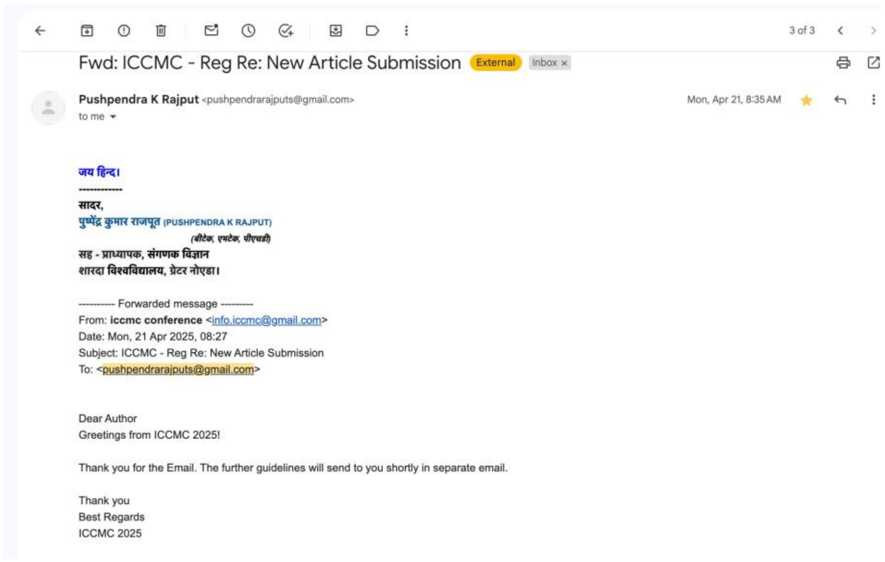
Title : AI-Driven Hybrid Recommender Systems

Abstract:

The development of e-commerce, especially inspecialist industries like health supplements, has heightened the demand for personalized recommendation systems capable of processing enormous datasets, resolving cold start difficulties, and adjusting to growing customer preferences. Data scarcity and scalability typically constrain conventional recommender systems—including content-based filtering and collaborative filtering—as they address. This paper offers an AI-driven hybrid recommender system that mixes sophisticated machine learning approaches, including deep learning and reinforcement learning, with content-based and collaborative filtering tactics. The proposed model delivers exact suggestions by merging implicit consumer data (e.g., views, cart additions, sales) with explicit product attributes (e.g., categories, tags, health conditions). To promote long-term user engagement, we construct a neural collaborative filtering (NCF) system that integrates a transformer-based content encoder and a reinforcement learning agent. The technique is intended for assessment using error measurements (RMSE, MAE), ranking metrics (accuracy, recall), and business metrics (click-through rates, conversion rates), potentially executed on a simulated health supplement e-commerce platform. Applied in healthcare, fashion, and other domains, this AI-driven technology offers a scalable way for tailored suggestions.

Authors:

Ravi Prakash, Giri Dhiraj, Shivam Rustagi and Dr. Pushpendra Kumar Rajput.



Fwd: ICCMC - Reg Re: New Article Submission External Inbox ×

3 of 3

Pushpendra K Rajput <pushpendrarajputs@gmail.com>
to me

Mon, Apr 21, 8:35 AM

जय हिन्द।

सादर,
पुष्पेंद्र कुमार राजपूत (PUSHPENDRA K RAJPUT)
(बीटेक, एमटेक, पीएचडी)
सह - प्राध्यापक, संगणक विज्ञान
शारदा विश्वविद्यालय, ग्रेटर नोएडा।

---------- Forwarded message ---------
From: **iccmc conference** <info.iccmc@gmail.com>
Date: Mon, 21 Apr 2025, 08:27
Subject: ICCMC - Reg Re: New Article Submission
To: <pushpendrarajputs@gmail.com>

Dear Author
Greetings from ICCMC 2025!

Thank you for the Email. The further guidelines will send to you shortly in separate email.

Thank you
Best Regards
ICCMC 2025

REFERENCES

1. Adomavicius, G., Tuzhilin A.: Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering, (2005) 17(6): p. 734-749

2. Canny, J.: Collaborative Filtering with Privacy via Factor Analysis. In Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval. (2002) Tampere, Finland. ACM Press p. 238-245

3. Harper, F., Li, X., Chen, Y., Konstan, J.: An Economic Model Of User Rating In An Online Recommender System. In Proceedings of the 10th International Conference on User Modeling, (2005) Edinburgh, UK p. 307-216

4. . Hill, W., Stead, L., Rosenstein, M., Furnas, G.: Recommending and Evaluating Choices in a Virtual Community of Use. In Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems. (1995) Denver, Colorado. ACM Press p. 194-201

5. Hofmann, T.: Latent Semantic Models For Collaborative Filtering. ACM Transactions on Information Systems (TOIS) (2004) 22(1): p. 89-115

6. Oard, D.W., Kim, J.: Implicit Feedback for Recommender Systems. In Proceedings of the AAAI Workshop on Recommender Systems. (1998) Madison, Wisconsin

7. Raymond J. Mooney and Loriene Roy. Content-based book recommending using learning for text categorization. In Proceedings of the Fifth ACM Conference on Digital Libraries, pages 195–204, San Antonio, TX, June 2000.

8. Alexandrin Popescul, Lyle Ungar, David M. Pennock, and Steve Lawrence. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In Proceedings of the Seventeenth Conference on Uncertainity in Artificial Intelligence, 2001.

9. D.D. Lee and H.S. Seung. Learning the parts of objects by non-negative matrix factorization. In Nature, 401 (788), 1999.

10. V. Valentino, H. S. Setiawan, . A. Saputra, Y. Haryanto and A. S. Putra, "Decision Support System for Thesis Session Pass Recommendation Using AHP (Analytic Hierarchy Process) Method," Journal International Journal of Educational Research & Social Sciences, pp. 215-221, 2021.

11. R. Suryadithia, M. Faisal, A. S. Putra and N. Aisyah, "Technological Developments in the Intelligent Transportation System (ITS)," International Journal of Science, Technology & Management, vol. 2, no. 3, pp. 837-843, 2021.

12. S. Rachmawati, A. S. Putra, A. Priyatama, D. Parulian, D. Katarina, M. T. Habibie, M. Siahaan, E. P. Ningrum, A. Medikano and V. Valentino, "Application of Drone Technology for Mapping and Monitoring of Corn Agricultural Land," IEEE, vol. 1, no. 1, pp. 1-5, 2021 .

13. A. S. Putra, H. Ludiya, N. Aisyah and B. S. Prasetyo, "INFLUENCE OF PRICES OF GOODS ANDA PROMOTIONAL MEDIA FOR E-COMMERCE SALES PLANNING SYSTEMS," Journal of Innovation Research and Knowledge, vol. 1, no. 3, pp. 249-254, 2021.

14. Jean, S., Cho, K., Memisevic, R. & Bengio, Y. On using very large target vocabulary for neural machine translation. In Proc. ACL-IJCNLP http:// arxiv.org/ abs/1412.2007 (2015).

15. Hadsell, R. et al. Learning long-range vision for autonomous off-road driving. J. Field Robot. 26, 120–144 (2009).

16. Fernando Amat, Ashok Chandrashekar, Tony Jebara, and Justin Basilico. 2018. Artwork personalization at Netflix. In Proceedings of the 12th ACM Conference on Recommender Systems. ACM, 487–488.

APPENDIX

The complete source code for the project is available at the following GitHub link:

Frontend: https://github.com/dhirajgiri3/ProductBazar-Frontend
Backend: https://github.com/dhirajgiri3/ProductBazar-Backend